Machine Learning-driven Intrusion Detection Techniques in Critical Infrastructures Monitored by Sensor Networks

by

Safa Otoum

A thesis submitted in partial fulfillment of the requirements for the

Ph.D. degree in

Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

In most of critical infrastructures, Wireless Sensor Networks (WSNs) are deployed due to their low-cost, flexibility and efficiency as well as their wide usage in several infrastructures. Regardless of these advantages, WSNs introduce various security vulnerabilities such as different types of attacks and intruders due to the open nature of sensor nodes and unreliable wireless links. Therefore, the implementation of an efficient Intrusion Detection System (IDS) that achieves an acceptable security level is a stimulating issue that gained vital importance.

In this thesis, we investigate the problem of security provisioning in WSNs based critical monitoring infrastructures. We propose a trust based hierarchical model for malicious nodes detection specially for Black-hole attacks. We also present various Machine Learning (ML)-driven IDSs schemes for wirelessly connected sensors that track critical infrastructures. In this thesis, we present an in-depth analysis of the use of machine learning, deep learning, adaptive machine learning, and reinforcement learning solutions to recognize intrusive behaviours in the monitored network.

We evaluate the proposed schemes by using KDD'99 as real attacks data-sets in our simulations. To this end, we present the performance metrics for four different IDSs schemes namely the Clustered Hierarchical Hybrid IDS (CHH-IDS), Adaptively Supervised and Clustered Hybrid IDS (ASCH-IDS), Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS) and Q-learning based IDS (QL-IDS) to detect malicious behaviours in a sensor network.

Through simulations, we analyzed all presented schemes in terms of Accuracy Rates (ARs), Detection Rates (DRs), False Negative Rates (FNRs), Precision-recall ratios, F_1 scores and, the area under curves (ROC curves) which are the key performance parameters for all IDSs. To this end, we show that QL-IDS performs with $\approx 100\%$ detection and accuracy rates.

Acknowledgements

I would like to express my gratitude to my supervisor Professor Hussein T. Mouftah, for his guidance, patience and caring. Without his support, encouragement, and especially his guidance, it would not have been possible to complete this work.

I am much obliged to my co-supervisor Dr. Burak Kantarci, for his unlimited support, valuable advice and sincere comments which helped me a lot to finish this study.

To my parents, who made all of this possible, for their endless encouragement and support, my success is yours. My high regards are addressed to my parents-in-law as well.

I would like to express my gratitude to my beloved husband Moayad for his support and all the sacrifices he has made.

I also acknowledge my sisters and brothers for the encouragements they provide.

Finally, I acknowledge my kids Rawad, Mohammad and Taj for being the joy of my life.

Dedication

To my parents who have always been behind me in every step of my life.

To my husband Moayad, without his encouragement and patience this work would not have materialized.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADSs	Anomaly Detection Subsystem.
AI	Artificial Intelligence.
AMG	Adaptive Model Generation.
ANN	Artificial Neural Network.
AR	Accuracy Rate.
ASCH-IDS	Adaptively Supervised and Clustered Hybrid Intrusion Detection System.
ASE	Attack Session Extraction.
B-H	Black-Hole.
BFS	Best First Search.
BN	Bayesian Network.
CH	Cluster Head.
CHH-IDS	Clustered Hierarchical Hybrid Intrusion Detection System.
DBN	Deep Belief Network.
DCO-IDS	Distributed and Co-Operative Intrusion Detection System.
DDoS	Distributed Denial of Service.
DL	Deep Learning.
DL-IDS	Deep Learning based Intrusion Detection System.
DoS	Denial of Service.
DR	Detection Rate.

DRBM	Discriminative Restricted Boltzmann Machine.
E-DBSCAN	Enhanced Density-Based Spatial Clustering of Applications with Noise.
ELM	Extreme Learning Machine.
FN	False Negative.
FNR	False Negative Rate.
FP	False Positive.
FPR	False Positive Rate.
GS	Genetic Search.
H-DSR	Hierarchical-Dynamic Source Routing.
HIDS	Hierarchical Intrusion Detection System.
ID	Intrusion Detection.
IDS	Intrusion Detection System.
KDD	Knowledge Discovery in Data mining Dataset.
KNN	K-Nearest-Neighbor.
MDSs	Misuse Detection Subsystem.
ML	Machine Learning.
ML-IDS	Machine Learning based Intrusion Detection System.
MRP	Markovian Reward Process.
NB	Naive Bayes.
NDAE	Non-symmetric Deep Auto-Encoder.
NIDS	Network Intrusion Detection System.
NS-3	Network Simulator version 3.
P2P	Peer-to-Peer.
PCA	Principal Component Analysis.
PDR	Packets Dropped Ratio.
PRCL	Pursuit Reinforcement Competitive Learning.
PSO	Particle Swarm Optimization.

QL-IDS	Q-learning based Intrusion Detection System.
R2L	Remote to Local attack.
RBC-IDS	Restricted Boltzmann Machine based Clustered Intrusion Detection System.
RBM	Restricted Boltzmann Machine.
RF	Random Forest.
RL	Reinforcement Learning.
RL-IDS	Reinforcement Learning based Intrusion Detection System.
RNN	Recurrent Neural Network.
RNN-IDS	Recurrent Neural Networks based Intrusion Detection System.
ROC	Receiver Operating Characteristics.
RS	Ranking Search.
RSS	Received Signal Strength.
RTMAS-AIDS	Real-Time Multi-agent System for an Adaptive Intrusion Detection System.
SIDS	Standalone Intrusion Detection System.
SL	Supervised Learning.
STL	Self-Taught Learning.
SVM	Support Vector Machine.
TN	True Negative.
TP	True Positive.
U2R	User to Root attack.
WSN	Wireless Sensor Network.

List of Symbols

A_τ	Action at time τ
a_x	Visible bias
b_y	Hidden bias
C	Number of sensor nodes in each cluster
D_n	Node n degree
d	Discount factor weight from future rewards to current rewards
$E(V, H\|\Theta)$	The energy function of the RBM
F_1	The weight associated with the *intimacy*
F_2	The weight associated with the *honesty*
F_3	The weight associated with the *energy*
F_4	The weight associated with the $unselfish$
$FP_1(t_i), FP_2(t_i)$	False Positive of ADSs and MDSs respectively at time t_i
g_d	Weight factor of ∂n
g_m	Weight factor of $\check{M}_n$
G_n	Overall weight of n
$g_{\hbar_n}$	Weight factor of $\hbar_n$
g_{sum}	Weight factor of $SRS_{strength}(n)$
H	Hidden element
$\hbar_n$	Cumulative duration of n being CH
$I(t_i)$	$M_1(t_i)/M_2(t_i)$

$Intrusive$	The binary output for intrusive detection
$\check{M}_n$	Mobility factor of n
$M_1(t_i)$	True positive to False positive ratio at time t_i for ADSs
$M_1(\triangle t)$	True positive to False positive ratio with time difference $\triangle t$ for ADSs
$M_2(t_i)$	True positive to False positive ratio at time t_i for MDSs
$M_2(\triangle t)$	True Positive to False Positive ratio with time difference $\triangle t$ for MDSs
$Maxpts$	Maximum neighbors to consider a point as core point
$Minpts$	Minimum neighbors to consider a point as core point
N	Number of clusters
n	Any node
$Normal$	The binary output for normal detection
O	Outputs
$P_{S\hat{S}}(A)$	The transitional probability from state S to $\hat{S}$ at A
$P(H)$	The probability allocated to any hidden element H
$P(H\|V)$	The probability of H independently with V
$P(V)$	The probability allocated to any visible element V, The network sets probability score to each case in hidden and visible elements[2][3]
$P(V\|H)$	The probability of V independently with H
$P(V,H)$	The probability of (V,H) formation
$\hat{Q}(S_\tau, A_\tau)$	Estimated Q value for state S and action A at time τ
R_τ	Reward at time τ
R^+	Positive reward
R^-	Negative reward
$R(S,\hat{S},A)$	Reward returned from transition from state S to $\hat{S}$ at A
$R_a(t_i)$	Proportion of incoming data directed to ADSs at t_i
$R_m(t_i)$	Proportion of incoming data directed to MDSs at t_i
$SRS_{strength}(n)$	Received signal strength sum of n

S_τ	State at time τ
T_a	The trust evaluation considering the *intimacy* factor
T_b	The trust evaluation considering the *honesty* factor
T_c	The trust evaluation considering the *energy* factor
T_d	The trust evaluation considering the *unselfishness* factor
T_{CH}	Aggregator (CH) trust value
T_{CH}^n	CH and node n trust evaluation
T_n	Node n trust value
T_{nm}	The trust evaluation between nodes n and m
${T_{nm}}^{intimacy}(t)$	The trust evaluation between nodes n and m considering the *intimacy* factor
${T_{nm}}^{honesty}(t)$	The trust evaluation between nodes n and m considering the *honesty* factor
${T_{nm}}^{energy}(t)$	The trust evaluation between nodes n and m considering the *energy* factor
${T_{nm}}^{unselfish}(t)$	The trust evaluation between nodes n and m considering the *unselfish* factor
$TP_1(t_i), TP_2(t_i)$	True Positive of ADSs and MDSs respectively at time t_i
T_{QoS}	The trust evaluation considering the QoS factors
T_{social}	The trust evaluation considering the *social* factors
V	Visible element
$V(S)$	Value estimation of R that at initial state S
$V_I(\hat{S})$	Value estimation of R at state $\hat{S}$ at the initial iteration I
$V_{I+1}(S)$	Value estimation of R at state S at the updated iteration $I+1$
W	The weights between visible and hidden layers
W_{xy}	Combined weights of visible V_x and hidden H_y units
X	Number of visible nodes

Y	Number of hidden nodes
α	Weight of the previous ROC characteristics in the evaluation of $M_1(t_i)$ and $M_2(t_i)$
β	Learning rate, where $0 < \beta < 1$
Γ	Capacity of a CH
Γ	Relative value of rewards, where $0 < \Gamma < 1$
$\triangle R$	Adjustment (incremental/decremental) value for the proportion of sensed data forwarded to a subsystem
$\triangle t$	Time difference between $(t_{i+1} - t_i)$
Θ	RBM parameters (W_{xy},a_x,b_y)
$\sum_{X,Y} e^{-E(V,H)}$	The normalization factor (all possible configurations including the visible and hidden elements)
∂n	Degree difference of n

Chapter 1

Introduction

1.1 Overview

Wireless Sensor Network (WSN) is a collection of independent sensor nodes that are scattered in an area to sense and monitor various circumstances then transmit the collected sensory data to a station [4]. WSN employs numerous types of sensors, such as: thermal, visual, radar, infrared and magnetic which help in increasing their ability to monitor a variety of conditions such as humidity, pressure, temperature, and noise levels [5][6].

WSNs architectures can be in distributed or hierarchical architectures. Hierarchical architecture, which is the one we adopt in our thesis work, forms a hierarchy based on nodes functionality which are sensor nodes, Cluster Heads (CH) and sink node [7].

WSNs are widely used in various critical infrastructures such as environmental, military and health-care fields. In most applications, WSNs deployed in open nature and unattended environments. Thus, the protection of the sensed data is strongly needed due to the existence of intruders who can eavesdrop the wireless links and interrupt the exchanged messages. Hence, security becomes one of the major issues while designing WSNs-based critical infrastructures.

Deployment of WSNs in critical infrastructures introduced additional security threats to both domains. The security vulnerabilities of such infrastructures can be in the cyber domain such as attacks on the communication links or in the physical domain such as attacks on sensors.

During the past few years and with the extensive growth of communication networks, many new hacking tools and intrusive approaches have appeared, as a result, security is becoming more stimulating. In addition to attacks and intrusions protection techniques such as user authentication and user authorization, intrusion detection is also essential for all-inclusive security of networks, dealing with suspicious activities within networks and automatically detect different intrusion attempts.

Intrusions are unauthorized tasks and malicious act of compromising networks conducted by intruders while IDSs are software applications or physical devices that detect malicious activities occurring in the studied networks.

Any data manipulation should be authorized by roles which defined in data integrity, availability, and confidentiality where the intruders aim to pass security layers in order to breach these roles[8][9].

Any intrusion aim to pass some stages such as [10][11]:

1. Probing stage where the intruder checks the targeted network to recognize its vulnerabilities.
2. Activity stage where the intruder gains some control of the targeted network for further activities.
3. Action stage where the intruder manipulates the networks' data and install some malicious software to manipulate other systems on the targeted network.
4. Masquerading stage where the intruder removes the attacks' traces.

On the other hand, IDS functions could be as Intruders' identification, stopping the ma-

licious behaviours and reporting the malicious behaviours to the networks' administrator [12]. IDS can be categorized into: First, the anomaly detection, Second, the misuse detection, and finally, the specification detection [12]. The anomaly and misuse-based detections' schemes are the ones used in our work. The anomaly-based detection depends on the behavioral methods in which it defines two types of behaviours, the normal and abnormal behaviours in which any deviation from the normal behaviours is considered as an anomaly action. Anomaly detection is suitable to detect unknown attacks. The advantage of this scheme is that it can accurately detect attacks with lower False Positives (FP) and False Negatives (FN) rates. On the other hand, the disadvantage of this scheme is that there is a need to update the normal profile regularly with any changes occur in the network.

The misuse/signature-based detection depends on attacks signatures such as the already known attacks' signatures are utilized to detect the future attacks. The advantage of misuse detection is that it can accurately detect the known attacks with low FP rate. On the other hand, the disadvantage of this scheme is that it is not useful for unknown attacks.

1.2 Motivations

WSNs are main technology in most of critical infrastructures due to their flexibility, self-deployment and low cost benefits. Their usage in monitoring and control processes are the most important fields. Monitoring critical infrastructures involve various types of sensors under different circumstances [13][14]. In most of the critical infrastructures, WSNs are deployed in an open nature environment [15][16][17][18]. Thus, sensor nodes can be captured and be prone to numerous attacks, such as black-Hole and sink-hole attacks, which can then manipulate the collected sensory data and launch various attacks which cause data loss. Sensors and the communication lines between them are vulnerable to numerous intruders that may interrupt and manipulate with the communication and transmitted data [19][20]. WSNs need to provide an acceptable level of security in order to achieve attacks free environments, reliable and secure data aggregation and transmission [21].

Intrusion Detection System (IDS) is an essential component for network security especially for the sensor networks to detect different types of intrusions. At the very high level, the validity of any IDS is determined by its ability to raise an alarm with the detection of any malicious node that exhibits intrusive behaviour [22].

1.3 Objectives

The objectives of this thesis are to carry on a critical investigation and review of security perspectives, schemes and mechanisms used in WSNs-based critical infrastructures, to investigate critical infrastructures' in order to find the proper solutions for different security threats/attacks and to propose IDS-based solutions that can provide maximum detection, maximum accuracy, minimum false alarms and maximum precision-recall. In addition, to propose an IDS that capable of detection different types of attacks include known and unknown attacks using machine-learning, deep-learning, and reinforcement-learning techniques. And finally, to compare the performance of different techniques to decide the best solution for intrusions' detection system in WSN-based critical infrastructures.

Several tools and software applications were used during the design and the evaluation of the proposed approaches, and the research also included many theoretical analyses and experiments.

1.4 Contributions

The major contributions of this thesis are as follows:

1. Proposed a hierarchical trust-based WSN monitoring model in order to detect the Black-Hole (B-H) attacks [21].

2. Proposed a Clustered Hierarchical Hybrid-IDS (CHH-IDS) to detect intrusive behavior of sensors for both anomaly and signature attacks and investigated mitigation of FNs through the two-tier intrusion detection approach [19].

3. Proposed an Adaptively Supervised and Clustered Hybrid IDS (ASCH-IDS) for wirelessly connected sensor clusters that monitor critical infrastructures. It transforms the previous work in [19] by continuously monitoring the behavior of the receiver operating characteristics, and adaptively directing the incoming packets at a sensor cluster towards either misuse detection or anomaly detection module [20].

4. Introduced Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS), a potential deep learning-based IDS methodology for monitoring critical infrastructures by WSNs [23].

5. Proposed an IDS approach by harnessing Q-learning as a reinforcement learning technique on a hybrid IDS framework.

6. Presented a comparative study of different IDSs for wirelessly connected sensors that track critical infrastructures. Specifically, we present an in-depth analysis of the use of machine learning, deep learning, and reinforcement learning solutions to recognize intrusive behaviour in the sensor network.

1.5 Outline

The thesis is organized as follows:

Chapter 2 presents the state-of-the-art knowledge in the field of Wireless Sensor Networks (WSNs)-based critical infrastructures security issues, and provides an overview of the Intrusion Detection Systems (IDSs) schemes, as well as, the approaches used in IDSs. Chapter 3 presents the use of machine learning and deep learning mechanisms in an IDS. Chapter 4 presents an adaptive solution for the presented machine learning scheme in chapter 3. Chapter 5 investigates the use of reinforcement learning in the proposed IDS. And finally, conclusions and future directions are presented in Chapter 6.

1.6 List Of Publications

- Journal Papers

    1. S. Otoum, B. Kantarci, and H. T. Mouftah, "Detection of Known and Unknown Intrusive Sensor Behavior in Critical Applications," in IEEE Sensors Letters, vol. 1, no. 5, pp. 1-4, Oct. 2017, Art no. 7500804.

    2. S. Otoum, B. Kantarci, and H. T. Mouftah, "AI-driven Intrusion Detection Techniques in Critical Infrastructures Monitored by Sensor Networks," submitted to IEEE Internet of Things Journal (IoT).

    3. S. Otoum, B. Kantarci, and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," in IEEE Networking Letters. doi: 10.1109/LNET.2019.2901792.

- Conference Papers

    1. S. Otoum, B. Kantarci, and H. T. Mouftah, "Hierarchical trust-based blackhole detection in WSN-based smart grid monitoring," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

    2. S. Otoum, B. Kantarci, and H. T. Mouftah, "Mitigating False Negative intruder decisions in WSN-based Smart Grid monitoring," 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, 2017, pp. 153-158.

    3. S. Otoum, B. Kantarci, and H. Mouftah, "Adaptively Supervised and Intrusion-Aware Data Aggregation for Wireless Sensor Clusters in Critical Infrastructures," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.

    4. S. Otoum, B. Kantarci, and H. Mouftah, "Empowering Reinforcement Learning

on Big Sensed Data for Intrusion Detection," Accepted in 2019 IEEE (ICC).

Chapter 2

State Of The Art

2.1 Introduction

Security of Wireless Sensor Networks' (WSNs) communications is an essential concern to deal with. WSN experience various security threats due to their essential characteristics and their vulnerability to numerous security threats due to their open communication environment compared to wired networks. In this chapter, an analysis of the state of the art on the security of WSNs and existing techniques to achieve reliability, accuracy and highest attacks detection rates, are presented. To have a wider perspective of the surveyed topic, the chapter is divided into different areas. The related work presented in this chapter presents a general overview of the work done in this area. This chapter consists of sections such as: In Section 2.2, we review the black-hole detection related works. In Section 2.3, we presented the state of arts that proposed the use of machine learning techniques for IDSs. The adoption of different deep learning techniques are presented in Section 2.5 and finally, in Section 2.6, we presented the reinforcement learning-based IDSs.

2.2 Black-Hole Attacks

Adoption of WSNs in critical infrastructures introduced additional security threats to both domains. The security vulnerabilities can be in the cyber domain such as attacks on the communication links or in the physical domain such as attacks on sensors. Black-Hole (B-H) attack is an example of cyber attacks. B-H is a routing layer attack and one of the emerging security threats in networks where the attackers apply loophole to spread malicious behaviors [6]. It occurs when a node blocks or drops the packets it receives instead of forwarding them towards the receiving node [24]. Thus, any data that traverses the B-H nodes is blocked which leads to performance degradation in network efficiency and excessive energy consumption.

Distributed evaluation of trust scores plays a vital role in distinguishing between normal and malicious devices [25]. Malicious devices can be attackers that threaten the security of the whole network. Replacing malicious devices help in achieving the security of the network.

In [26], the researchers proposed an exponential trust-based mechanism in order to detect the B-H attacks. In their proposed model, they introduce nodes with trust factors and a counter of the dropped packets, which helps them in detecting the malicious nodes. The trust factors of the nodes are calculated by the exponential formula $100X^{ni}$ where n is the counter value.

In [27], the authors propose an algorithm to overcome the B-H and grey-hole attacks. The grey-hole attack has the same behavior as the B-H; however, it does not drop whole packets as B-H but drops a sub-stream of an incoming packet stream instead. The algorithm is based on dividing the data into smaller data blocks to discover the malicious nodes. The neighboring nodes collaborate in the transmission of the data blocks from the source towards the destination. The acknowledgements by the destinations help in the detection

of malicious nodes.

The authors in [28] propose a hierarchical secure routing protocol against B-H attacks by using the symmetric key cryptography to find out the secure route. The authors divided the proposed network into a number of groups organized as a tree topology, each group leader acts as a root of the tree. The B-H attack was detected by using the randomized data acknowledgement scheme.

In [29] the authors propose an efficient and trust-based secure protocol to protect against single and cooperative B-H attack. In their proposed protocol, the trust metrics are used to find a node honesty during a secure path formation.

2.3 Machine learning-based Intrusion Detection Systems

During the past few years and with the extensive growth of communication networks, many new hacking tools and intrusive approaches have appeared, as a result, security is becoming more stimulating. In addition to attacks and intrusions protection techniques such as user authentication and user authorization, intrusion detection is also essential for all-inclusive security of networks, dealing with suspicious activities within networks and automatically detect different intrusion attempts. The effectiveness of any Intrusion Detection System (IDS) is determined by its possibility of identifying an anomalous condition upon an intrusion attempt [30]. Two different methods used to recognize attacks, namely the anomaly detection method and signature (misuse) detection method. In anomaly detection method, a normal system behavior profile is formed and any deviation from that defined profile is marked as an anomaly. Signature detection method detects the attacks based on their pre-known patterns [31]. Anomaly detection can detect attacks with a high False Positive (FP) rate due to its lower detection rates [32]. On the other hand, signature detection results in low False Positive (FP) rate due to its higher detection rates since the attack signatures are pre-defined.

Addressing security vulnerabilities is the most essential issue in WSN-based critical infrastructures. Although the intrusion detection literature is tremendously rich, there is no perfect intrusion detection approach, which can always properly differentiate between malicious and normal activities when known and unknown attacks co-exist. The consequences of False Negatives (FNs) are unauthorized activities in the networks whereas False Positives (FPs) may simply block legitimate access. Thus, when real attacks are experienced, True Positives (TPs) are hidden within FPs [33].

The authors in [34] provided an overview of different IDS algorithms, such as artificial neural networks, swarm intelligence, fuzzy sets and soft computing. A collaborative intelligent IDS and a fuzzy inference system were proposed to reduce FPs through fuzzy alert correlation in [35], [36], respectively, while the authors in [33] reduced both FPs and FNs with their environmental-aware IDS where they integrate the characteristics and the properties of the protected system in traffic analysis such as security policy and network topology. A system of Attack Session Extraction (ASE) was proposed in [37] to create a pool of traffic traces causing possible FPs and FNs to IDSs. The PCAPLib system is an extended version of the ASE [38], which anonymized users privacy in the FP and FN traffic traces out of security considerations. However, previous work only focuses on studying how to reduce FPs and FNs in IDSs or how to collect and extract the FP and FN traffic traces from real-world traffic.

In [39], the authors introduced an IDS using a Bayesian approach for wireless network aiming at minimum FPs and FNs. Their objectives were to recognize signatures of known attacks, match the observed behavior with those signatures and signal an intrusion alarm when the match occurs.

In [40] the authors proposed a hybrid intrusion detection framework based on random forests and k-means methods. They tested their framework by focusing on false positive rate with detection rate. Their system achieved a high detection rate with 1% attacks which started to reduce with rising the percentage of attacks reaching 95% overall detection rate. Their system checks for anomaly and signature detection individually; by checking for signature intrusions followed by anomaly intrusions checking.

In [41] the authors combined anomaly and misuse detection and proposed an integrated detection model in which they adopted the Adaboost algorithm with hierarchical structures for anomaly detection of nodes. On the other hand, they adopted Cultural-Algorithm and Artificial-Fish-Swarm-Algorithm to misuse detection of Sink node. Again, and as in

[40], they checked for anomaly and misuse intrusions individually and achieved an overall detection rate of 92.40%.

Some authors proposed IDSs based on network trust evaluation as the work done in [42], [43], [44], [45]. In [42], the researchers proposed a trust-based intrusion detection scheme by using a highly scalable hierarchical trust management protocol for clustered WSNs. They used an analytical model based on stochastic Petri-nets for performance evaluation of the proposed scheme, and a statistical method for calculating the probability of false alarms. The authors in [44] proposed a trust-based IDS for Mobile Ad hoc Network (MANET). They proposed a trust model to enable the trust among the nodes which verified before the intruder is detected then evaluated the effectiveness of the system by comparing with the Standalone Intrusion Detection System (SIDS), Distributed and Co-Operative Intrusion Detection System (DCO-IDS) and Hierarchical Intrusion Detection System (HIDS). Their model was a lack of trust which leads to a higher number of attacks and higher false detection percentage.

The work done in [45] discussed a trust-based mechanism combined with Ad Hoc Network based intrusion detection system (IDS) which ensure the security services required by users. As well as, the authors in [43] introduced an ID framework for MANET, based on the trust relationship. Their proposed IDS used the local and the global determination of attacks within the network where the nodes should watch the suspicious activities of neighboring nodes. An ID alert message is spread throughout the network to report the anomaly. Their proposed framework handled the disadvantages from the mobility of nodes and the probability of selfishness.

The authors in [46] presented a three phases based IDS such as they: 1) used a hybrid approach to feature selection, 2) used base classifiers for classification, and 3) deployed a majority voting strategy to form the final decision. The proposed feature selection process was based on Best First Search (BFS), Genetic Search (GS), and Ranking Search

(RS). The final set of features was derived, by combining the results from all three feature selection algorithms, where the features most commonly chosen by all three algorithms were propagated to the last set. The classification at the second stage was performed by three Nave Bayes (NB), Bayesian network (BN), and J48 (classification trees). They tested their proposed work using the NSL-KDD data-set.

The work presented in [47] adopted two dimensionality reduction methods, Principal Component Analysis (PCA) and Fuzzy PCA which allow keeping the most relevant information from the network traffic data. In order to classify the tested dataset into normal and attacks, they have applied the K Nearest Neighbour (KNN) algorithm. Their results show that the Fuzzy PCA method outperforms PCA in detecting U2R and DoS attacks.

Another work for enhancing the anomaly-based intrusion detection systems performance through dimensionality reduction has been presented in [48] where the authors adopted the PCA principle. PCA reduced the dataset dimensionality using the dependencies between the input features to represent it in a lower dimensional form. Their result shows that adopting PCA can reduce the dimensionality of the data being processed by anomaly-based IDSs and thereby minimize their computational overhead without adversely affecting their performances.

An IDS using hybrid binary particle swarm optimization (PSO) and k-nearest-neighbor (kNN) has been presented in [49] where the authors proposed a new hybrid algorithm k-NN-PSO which successfully enhanced the accuracy generated by KNN by up to 2%. The proposed technique consists of two steps: feature selection and classification, where the features are selected with the help of binary PSO, and classification is performed by kNN algorithm.

In [50], the authors evaluated the performance of different kernels of Support Vector Machine (SVM) against KDD and detection accuracy, detection time are studied. The researchers found that adopting Principal Component Analysis (PCA) had been reduced

the detection time as well as adopting the Gaussian Radial Basis Function kernel of SVM achieved the highest detection accuracy.

Another example of using PCA for data dimensionality reduction and a classifier is the work presented in [51] in which the authors proposed a novel model for intrusion detection which had been used as an online machine learning algorithm. Their model adopted the PCA to reduce data dimensionality and adopted softmax regression and kNN algorithms to develop a classifier.

In [52], the work presented a learning model for Fast Learning Network (FLN) based on Particle Swarm Optimization (PSO) namely the PSO-FLN. Their proposed model had been applied to intrusion detection issue. PSO-FLN had been compared with meta-heuristic algorithms and achieved the highest testing accuracy compared to other algorithms.

To the best of our knowledge, a holistic IDS for WSN-based critical infrastructures' monitoring that works for both known and unknown attacks remains non-addressed. Therefore, based on this motivation, we present our Clustered Hierarchical Hybrid-Intrusion detection system (CHH-IDS) approach by focusing on mitigation of misleading decisions against malicious or legitimate nodes. On the other side, there is no definite detection method of intruders in sensor networks combining RF-based signature and Enhanced-DBSCAN-based anomaly detection.

2.4 Adaptive machine learning-based Intrusion Detection System

Adaptive machine learning-based techniques for IDS have been tested in many studies to improve the accuracy of classification such as the work presented in [53] where the researchers presented a fully IDS validation procedure by presenting the automated and adaptive testing prototype.

As another example of an adaptive IDS, the work in [54] proposed an Adaptive Model Generation (AMG) as a model generator which would work adaptively in real time to perform IDSs. AMG enables evaluation of data in real time and automates the data collection, as well as the generation and deployment of detection models.

In [55], the authors proposed a methodology of anomaly-based IDS which used the game theoretical approach. They designed a lightweight anomaly detection technique by achieving a trade-off between detection rate, energy consumption, and false positive rates [55].

A Bayesian Network (BN) had been used to build an automatic IDS based on signature recognition [56]. In [56], the authors provide an adaptive IDS using BN by recognizing known attacks' signatures, matching the spotted behaviours with the recognized signatures, and detect intrusion when a match is found. Since intrusions' signatures change with time and the system must be retrained, the IDS must be adapt to such changes by auditing the data and detect both normal and abnormal connections.

The Genetic Algorithm (GA) had been used to design an adaptive IDS based on selecting features for profiling and parameters for anomaly-based intrusion detection methods [57]. In [57], the authors used two anomaly-based methods to be coupled with their proposed approach, such as: the first one is based on basic statistics and the second one is based on a projected clustering procedure. Their method achieved a detection rate of 92.85% and

false positive rate of 0.69%.

Other adaptive IDSs had been presented in [58, 59, 60, 61]. In [58], the researchers proposed a Real-Time Multi-agent System for an Adaptive Intrusion Detection System (RTMAS-AIDS) that is based on a multi-agent system to allow the IDS to adapt to real-time unknown attacks. Their method used the Support Vector Machine (SVM) and Extreme Learning Machine (ELM) methods to detect normal behavior and known attacks, where the adaptive SVM allows processes to run in parallel in order to detect and learn new attacks in real-time. Their results show that the RTMAS-AIDS can detect Probe, R2L, and U2R attacks better than the non-retrained multi-agent system using the multi-level hybrid SVM and ELM models as well as the multi-level hybrid SVM and ELM.

The work presented in [59] proposed a distributed self adaptive IDS based on programmable mobile agents which can act as a key line of defense against major security attacks, the IDS is organized as a combination of the rule-based and the behavior-based schemes.

In [60], the authors proposed an anomaly detection-based automatic IDS using data mining methods. Their system is based on applying mining algorithms to characterize the normal system activities with a profile so that abnormal activities can be detected by comparing the current activities with the profile [60]. The work proposed in [61] focused on DoS attacks prevention as well as they proposed an anomaly-based IDS that used the chi-square test and control chart to detect intrusions and identify the intruders.

To the best of our knowledge, an adaptive-IDS solution for WSN-based monitoring applications to deal with both known and unknown intruders remains an open issue. In our proposed scheme, ASCH-IDS, we present a dynamic adjustment methodology for the proportion of the sensed data directed to the anomaly and misuse subsystems.

2.5 Deep learning-based Intrusion Detection Systems

Deep learning (DL)-based methods had been used to deal with IDSs challenges such as the feature selections' difficulties. DL-based methods have been applied to IDS and achieved highly accurate results [62][63][2][64]. In [65] the authors tested the abilities of a Deep Belief Network (DBN) in the detection of intrusive patterns. The authors in [62] combined DBN and SVMs and introduced a hybrid IDS methodology where DBN served as a feature selector and SVM as the classifier. The hybrid approach resulted in ≈92% accuracy rate.

In[66], the authors proposed a new deep learning classification model while in [67] the authors used recurrent neural networks for intrusion detection tasks.

A deep learning approach based on a deep neural network for flow-based anomaly detection has been proposed in [68] where the results show that deep learning can be applied for anomaly detection in software defined networks.

Another deep learning-based technique for Self-Taught Learning (STL) on NSL-KDD dataset in the IDS classification has been developed in [69]. When comparing its performance with those observed in previous studies, the method is shown to be more effective.

Authors in [70] presented a partially supervised learning approach where the classifier was trained with normal traffic only so any knowledge about malicious behaviours could evolve dynamically. The authors applied Discriminative Restricted Boltzmann Machine (DRBM) to anomaly detection as an energy-based classifier.

High dimensionality is a grand challenge in big data applications; hence the authors in [71] applied an auto-encoder in the first stage of an IDS in order to reduce dimensionality and extract the features for a DBN to classify anomalous and normal behaviour patterns.

The essential design of any deep learning network requires using a Restricted Boltzmann Machine (RBM) as an unsupervised learning method [72]. Examples of this include the

work done in [63] and [73]. Researchers in [65] explored the capabilities of Deep Belief Networks (DBN) for detecting intruders through a series of experiments.

DBN and SVMs have been introduced for intrusion detection classification purposes on the KDDCup99 dataset. With DBN as a feature selector and SVM as a classifier, the results showed a 92.84% accuracy rate [62]. Partially supervised learning approaches are presented in [70], where authors used real world data to evaluate their approach. A hybrid approach based on DBN and auto-encoder is shown in [71]. The auto-encoder method is used to decrease data dimensionality and extract the main features. After feature reduction, the DBN is applied to detect anomalous behaviour. The work in [3] employed the Restricted Boltzmann Machine (RBM) to remove KDDcup99 noises and introduce a new data set. In [2], the researchers used RBM for network IDS to test its capability to learn the complex data. They also proposed a systematic way of dataset learning.

In [74], Recurrent Neural Network (RNN) is considered as reduced-size neural networks in which the authors proposed a three layer RNN architecture with 41 features as inputs and 4 intrusion categories as outputs for misuse-based IDS. However, the authors did not study the models' performance in the binary classification.

The work in [75] presented IoT-based threats analysis using an Artificial Neural Network (ANN). The MultiLayer Perceptron (MLP) as a type of ANN that trained using supervised learning procedures, has been adopted. The presented work has been assessed on its ability to detect Distributed Denial of Service (DDoS/DoS) attacks and results show 99.4% accuracy in detecting various DDoS/DoS attacks.

Recurrent Neural Networks-based IDS (RNN-IDS) had been used in [76] where the authors proposed an intrusion detection system based on deep learning. The RNN-IDS performance has been compared with those of J48, Artificial Neural Network (ANN), Random Forest (RF) and Support Vector Machine (SVM) and the results show that RNN-IDS is suitable for classification models with high accuracy and that its performance is outperformed that

of traditional machine learning classification methods.

Another example of adopting deep learning methods for IDS is the work presented in [77]. In [77], the authors presented a novel deep learning classification model using stacked Non-symmetric Deep Auto-Encoder (NDAE) for intrusion detection to address the Network IDSs' (NIDS) feasibility and sustainability. Their proposed model achieved improvements over existing approaches.

To the best of our knowledge, a comprehensive comparison/evaluation of an IDS for WSN-based critical monitoring infrastructures that works for both known and unknown attacks using RBM-based deep learning remains an open issue.

2.6 Reinforcement learning-based Intrusion Detection Systems

Reinforcement learning (RL) considered as an extension to machine learnin and involves agents that take actions to maximize the notion of rewards. Previously, the reinforcement learning process was applied into IDSs by several studies such as the works presented in [78, 79, 80, 81, 82].

In [78], the researchers proposed a distributed reinforcement learning approach in which each agent (i.e. sensor) analyzes state observations and communicates them to a central agent. Agents that are higher in the hierarchy are equipped with the knowledge for analyzing the collected data, and they broadcast an overall abnormal state to the network operator [78].

In [79], the authors presented reinforcement learning-based technique for host-based IDS through a series of system calls by presenting a Markovian reward process (MRP) to replicate the behaviour of system call series where the intrusion detection issue is transfigured to predict the MRP value functions. The work presented in [80] proposed an approach to detect intrusion with online clustering by using Pursuit Reinforcement Competitive Learning (PRCL).

The researchers in [81] presented an approach of an adaptive neural network to intrusion detection that detects new attacks autonomously with the use of reinforcement learning method. The work presented in [82] exploits the use of data flow information, Reinforcement Learning and tile coding to detect flooding-based Distributed Denial of Service (DDoS) attacks.

A comparative study of supervised learning and reinforcement learning in intrusion domain is presented in [83] where the authors compared the performance of Supervised Learning

(SL) and RL by using the feature ranking method, such as the number of discrete attributes are reduced and used for classification in the Supervised Learning. They tested some supervised methods such as CS-MC4, ID3, Naive Bayes, C4.5, Random tree and compared their classification result with the ones derived from using the reinforcement learning combined with Rough Set Theory (RST) classifier and show that RL, along with RS. Their results show that the RL achieved higher classification accuracy with respect to some SL classification algorithms.

2.7 Summary

This chapter reviewed the Intrusion Detection (ID) techniques used in Wireless Sensor Networks (WSNs). Much research has been conducted into motivating and accessing the benefits of Intrusion Detection Systems (IDSs), as well as the many services and applications that can be deployed. More specifically, we have discussed the existing methods for black-hole (B-H) attacks detection, Machine Learning-based IDS (ML-IDS), Deep Learning-based IDS (DL-IDS), Adaptive Machine Learning-based IDS (AML-IDS) and the Reinforcement Learning-based IDS (RL-IDS). However, it is noted that the most proposed IDSs in the literature are either inefficient or have low Detection Rates (DRs), low Accuracy Rates (ARs) and high False Positive Rates (FPRs).

Chapter 3

Machine and Deep Learning-based IDS for WSN

The integration of WSN in critical infrastructures has gained a huge interest according to their flexibility, self-deployment, and low cost benefits which in another hand introduced security threats, such as jamming. Security susceptibilities can occur in either cyber or physical domains, including intrusions to communication links and sensor nodes. Intrusion Detection (ID) was introduced as an essential solution for network security, to deal with intrusive activities in communication networks and detect various intrusion attempts automatically [84]. Efficiency and reliability of critical infrastructures such as medical infrastructure can be met by using reliable and secure data aggregation and transmission [21]. Sensors and the communication lines between them are vulnerable to numerous intruders that may interrupt and manipulate with the communication and transmitted data [19][20]. IDS is an essential component for network security especially for the sensor networks to detect different types of intrusions. At the very high level, the validity of any IDS is determined by its ability to raise an alarm with the detection of any malicious node

that exhibits intrusive behaviour [22]. IDSs are categorized into two sets: Anomaly-based IDS and Signature-based IDS. The former aim to detect abnormal traffic patterns that deviate from normal behaviour. To this end, it creates profiles of the features to seize the needed patterns. The variance between the extracted patterns and noticed activities may lead to an alarm. Thus, ideally, Anomaly-based IDSs are able to detect unknown attacks. However, it is worth noting that most IDS systems still suffer from False Positive (FP) decisions, i.e., a non-malicious activity's being marked as an intrusive event [19].

Apart from the anomaly-based IDS, signature IDSs uses the rule complement mechanism to detect malicious behaviour by comparing system activities with specific rules. An intrusive event is detected when the noticed activities match a known malicious pattern. The risk of solely adopting a signature-based IDS is that its possible low precision in the likelihood of an unknown attack [19][22].

In this chapter, we begin by reviewing previous works related to adopting machine and deep leaning for IDS. Next, we present a solution for Black-Hole (B-H) attacks' detection followed by a comparative study of intrusion detection in critical infrastructures monitored by sensor networks which compare the usage of machine learning-based solution (random forest and E-DBSCAN) and deep learning-based solution (restricted Boltzmann machine).

3.1 Related Work

With emerging of monitoring critical infrastructures-based Wireless Sensor Network (WSN), the traditional techniques become more complex to deal with critical applications. In this section, we show some researches that adopted machine learning techniques for IDS to deal with WSNs-based critical infrastructures. The work done in [85] used cluster machine learning technique by adopting k-Means method on Spark to determine if the network traffic is an attack or a normal one. The authors used the KDD Cup 1999 dataset for training and testing. The authors did not use feature selection technique to select the related features. The authors in [86] proposed a clustering method for IDS based on Mini Batch K-means combined with principal component analysis (PCA). The PCA method has been used to reduce the dimension of the dataset and the mini batch K-means++ method has been used for data clustering. The KDDCup1999 dataset has been used to test the proposed model.

In [87], the authors proposed an IDS system based on decision tree over Big Data in Fog Environment by introducing a pre-processing algorithm to classify the strings in the studied dataset and normalize the data to ensure the quality of the input data so as to improve the efficiency of detection. They used the decision tree method and compared the method with the Nave Bayesian method as well as the KNN method. Their experimental results showed that their proposed method is effective. Another work has been presented in [88] where the researchers evaluated the performance of SVM, Nave Bayes, Decision Tree and Random Forest classification algorithms of IDS using Apache Spark.

The work in [89] proposed a real-time intrusion detection system based on SVM and Apache Storm where the authors used lib-SVM and C-SVM classification techniques for intrusion detection.

Recently, Deep Learning (DL), has gained numerous interests in machine learning stud-

ies and pattern recognition [90] and shown their effectiveness in most ML tasks such as intrusion detection tasks. DL belongs to a class of machine learning techniques which adopts consecutive layers of information processing for pattern classification and feature learning in hierarchical manners [91]. In [92], the authors investigated the performances of their presented state-of-the-art attack algorithms against DL-based IDS by validating the vulnerabilities of neural networks employed by the IDSs.

In [77], the researchers proposed a new DL classification model for unsupervised feature learning namely non-symmetric deep auto-encoder (NDAE). In [69], the authors focused on flow-based anomaly detection by applying a DL procedure. Carlini and Wagner in [93] proposed new gradient-based attack algorithms namely (L_2, L_∞, L_0). Their proposed L_2 attack adopted a logits-based objective function whileL_∞, L_0 are based on the L_2 attack and tailored to different distance metrics.

3.2 Black-Hole (B-H) Attacks Detection

Black-Hole (B-H) attack is considered as an active attack in which the malicious node claims that it has the shortest path to the destination node even if it does not have any path to it, as a result, all packets pass through it will be discarded or dropped. Figure 3.1 represents an illustration of B-H attack. As shown in Figure 3.1 when the source node requests a path to the destination node, the B-H node claims that it has the shortest path to that destination node. B-H node drops the forwarded packets to prevent the communication between the source and the destination node.

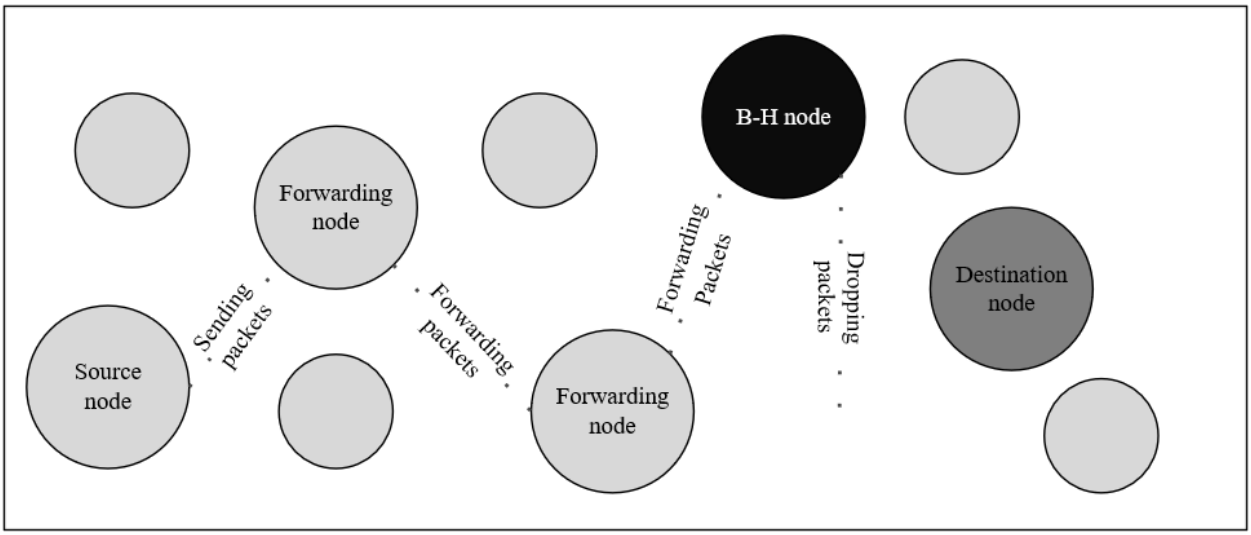

Figure 3.1: Illustration of Black-Hole (B-H) attack

Providing a secure environment for Wireless Sensor Networks (WSNs)-based critical infrastructures with minimum probability of attacks and intrusions is needed as well as is not an easy task due to the peculiarities and the difficulties of it. As an example of such attacks, the Black-Hole (B-H) attack considered as one of the vital attacks that face WSNs infrastructures which occurs when an intruder reprograms set of nodes to block or drop the packets they receive instead of forwarding them to the base station.

In the remaining sections, we introduce the adopted hierarchical trust evaluation procedure followed by our proposed solution to B-H attacks detection.

3.2.1 Hierarchical trust evaluation method

In our proposed model we adopt the peer-to-peer (P2P) hierarchical trust evaluation process in [94] in order to evaluate the trust between the sensor nodes that are installed on the critical infrastructures. We have also used the P2P hierarchical trust evaluation to distinguish the legitimate nodes from illegitimate nodes and to build a secure data aggregation scheme in the CHs, which can assess the trustworthiness of sensor nodes in their clusters.

The hierarchical trust evaluation method considers both the *Quality of Service (QoS)* and the *social* factors in order to consider the effect of both aspects on trustworthiness. The *QoS* factors are the *energy* (to measure the competence) and *unselfishness* (to measure the cooperativeness). While the *honesty* (to measure normality/abnormality) and the *intimacy* (to measure nodes' closeness based on the interaction experiences) fall under the social factors.

The *intimacy* refers to the degree of interaction experiences between nodes n and m where the more positive experiences between nodes n and m lead to more trust. The *honesty* component decides weather the node is malicious or normal.

The *energy* component measures weather the node is able to perform its function and the *unselfishness* component reflects if the node can execute the intended protocol [94].

The hierarchical trust evaluation method in [94] is described in eqs.(3.1) - (3.4) bellow where T_a stands for the trust evaluation between nodes n and m considering the *intimacy* factor, T_b is the trust evaluation considering the *honesty* factor, T_c refers to the trust evaluation considering the *energy* factor and, T_d stands for the trust evaluation considering the *unselfishness* factor.

T_{nm} represents the trust evaluation between nodes n and m respectively and F_i represents the weights associated with the components where $F_1+F_2+F_3+F_4=1$. T_{nm} is considered as a real number that lies in [0,1]. With the arrival of a new node, the trust value is initially evaluated.

$$T_a = F_1 {T_{nm}}^{intimacy}(t) \tag{3.1}$$

$$T_b = F_2 {T_{nm}}^{honesty}(t) \tag{3.2}$$

$$T_c = F_3 {T_{nm}}^{energy}(t) \tag{3.3}$$

$$T_d = F_4 {T_{nm}}^{unselfish}(t) \tag{3.4}$$

Where F_1, F_2, F_3 and F_4 stand for the factors associated with the trust evaluation between nodes n and m considering *intimacy*, *honesty*, *energy* and *unselfishness* factors, respectively. The *social* and *QoS* factors are formulated in eqs. 3.5 and 3.6. respectively.

$$T_{social} = T_a + T_b \tag{3.5}$$

$$T_{QoS} = T_c + T_d \tag{3.6}$$

The trust evaluation between nodes n and m is computed by eqs.3.7 and 3.8 below:

$$T_{nm} = T_a + T_b + T_c + T_d \tag{3.7}$$

$$T_{nm} = F_1 {T_{nm}}^{intimacy}(t) + F_2 {T_{nm}}^{honesty}(t) + F_3 {T_{nm}}^{energy}(t) + F_4 {T_{nm}}^{unselfish}(t) \tag{3.8}$$

In equation 3.8:

- ${T_{nm}}^{intimacy}(t)$ represents the intimacy, which refers to the level of interaction between nodes n and m. Intimacy trust is based on the maximum number of interactions between nodes n and m. The interaction refers to the communication when sending and receiving data from a node such as a node with the highest number of communications is the highly trustworthy node.
- ${T_{nm}}^{honesty}(t)$, stands for the node honesty, which is increased when a node performs its proposed functions successfully and is based on the successful/unsuccessful interactions between nodes. Successful interaction happens when a node successfully transmits the data to another while the unsuccessful interaction happens when a node fails to transmit the data to another node. Node n estimates ${T_{nm}}^{honesty}(t)$ by tracking the count of dishonest experiences of node m during [0, t] using the repetition, re-transmission, and delay rules as identified in [95]. Packet drops or detected attacks by a node leads to mark that node as dishonest.
- ${T_{nm}}^{energy}(t)$ represents the level of residual energy of the nodes. Node n estimates node $m's$ remaining energy by overhearing its packet transmission activities over [0,t].
- ${T_{nm}}^{unselfish}(t)$ denotes the unselfishness level of node m as estimated by node n based on the observations over [0, t] . Typical behavior of a selfish node is stopping sensing tasks and dropping its received packets to save energy.

3.2.2 Packet Drop Ratio-based Black-Hole Attacks Detection

One of our work objectives focused on detecting the B-H attack in order to achieve secure communication, data aggregatio, and private data environment by maintaining B-H proof environment. In order to achieve a successful aggregation operation, we have to ensure that the CHs are resilient against attacks. Each CH checks for a B-H attack before aggregating

the incoming data from the nodes inside its corresponding cluster by tracking its dropped packets in order to avoid any B-H occurrences. Thus, the ratio of dropped packets (Packet Drop Ratio (PDR)) should be less than a pre-defined threshold. In case of the PDR being larger than the threshold, a re-clustering operation should be considered.

The B-H detection overall system model is illustrated in 3.2. The system consists of N clusters where the cluster heads (CHs) are the forwarding nodes. The aggregated data is moved to the centralized sink, which is connected with the cloud in order to save the data.

The Hierarchical-Dynamic Source Routing (H-DSR) protocol has been employed in the proposed model in order to achieve the required efficiency because of its advantages of minimizing the resulted overhead.

Our strategy for B-H detection is presented in the flowchart in Figure 3.3. Initially, each node is assumed to maintain a 100% trust score. Second, weighted clustering operation is implemented in order to elect a CH for each cluster, which considered as the aggregator node. Due to the aggregator importance, the trust scores of the CHs and their corresponding sensor nodes are also assessed periodically. In order to keep track of CHs effectiveness, PDR-based trust score assessment is called: If the PDR of a CH exceeds a pre-defined threshold, we consider the corresponding CH as a B-H node and a re-clustering operation is triggered to elect a new CH for this cluster.

B-H detection algorithm is shown in Algorithm 1 bellow.

3.2.3 Simulation Settings and Results

We have tested the proposed B-H detection procedure by using the network simulator version 3 (NS-3 simulator) which is a discrete-event network simulator licensed under the GNU GPLv2 [96], by considering a network of twenty sensor nodes with four clusters spread out in a 100m x 100m area.

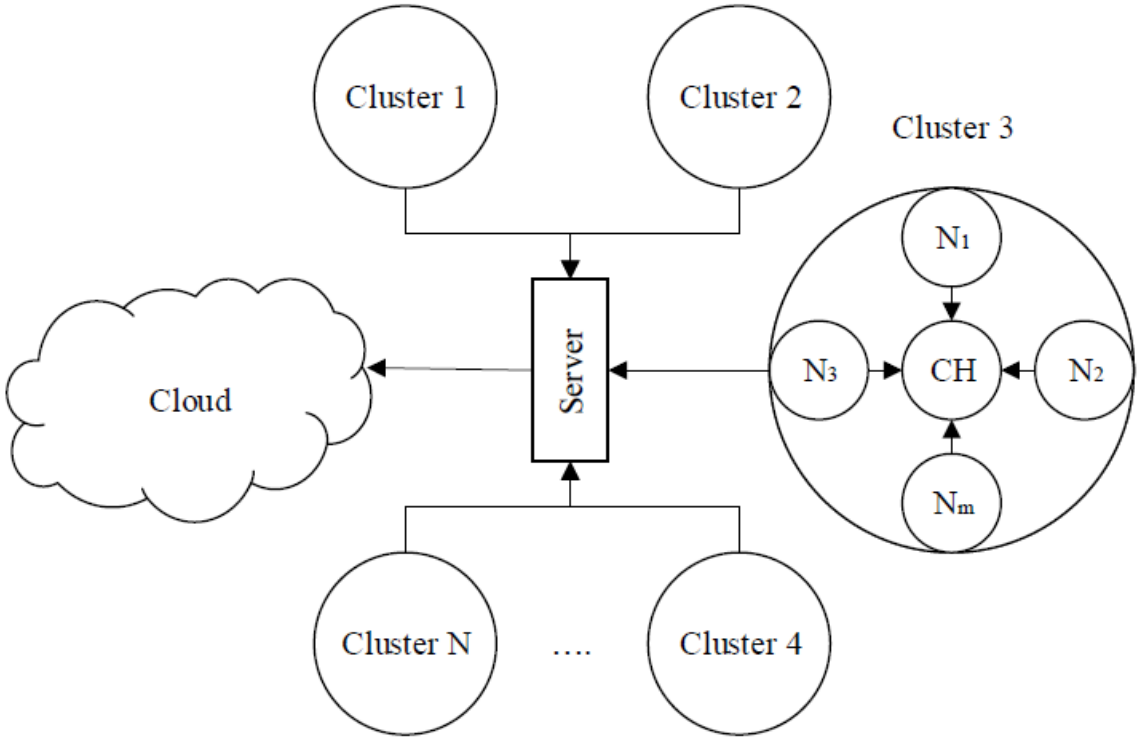

Figure 3.2: Black-Hole (B-H) detection overall system model

In the beginning, all sensors have a trust ratio of 100% (trust score of 1). The trust ratios of the nodes decrease continuously with the network functions. In order to test our proposed system model behaviour, we have run tests by setting the PDR threshold to different threshold values: 0.5, 0.4, 0.3 and 0.2. Figures (3.4 - 3.11) represent the pre-clustering and post-clustering phase values of the PDR with respect to varying trust score of each CH under four different trust thresholds. The post-clustering phases denote the situation after a number of re-clustering attempts where each CHs' PDR is enforced to be under the defined thresholds.

Different thresholds are represented in details as the following:

- PDR threshold of 0.2: By testing the threshold of 0.2, it needed five re-clustering operations. The re-clustering operations started from early periods when the trust scores were around 0.7 as shown in Figures 3.4 and 3.5. In the first four re-clustering

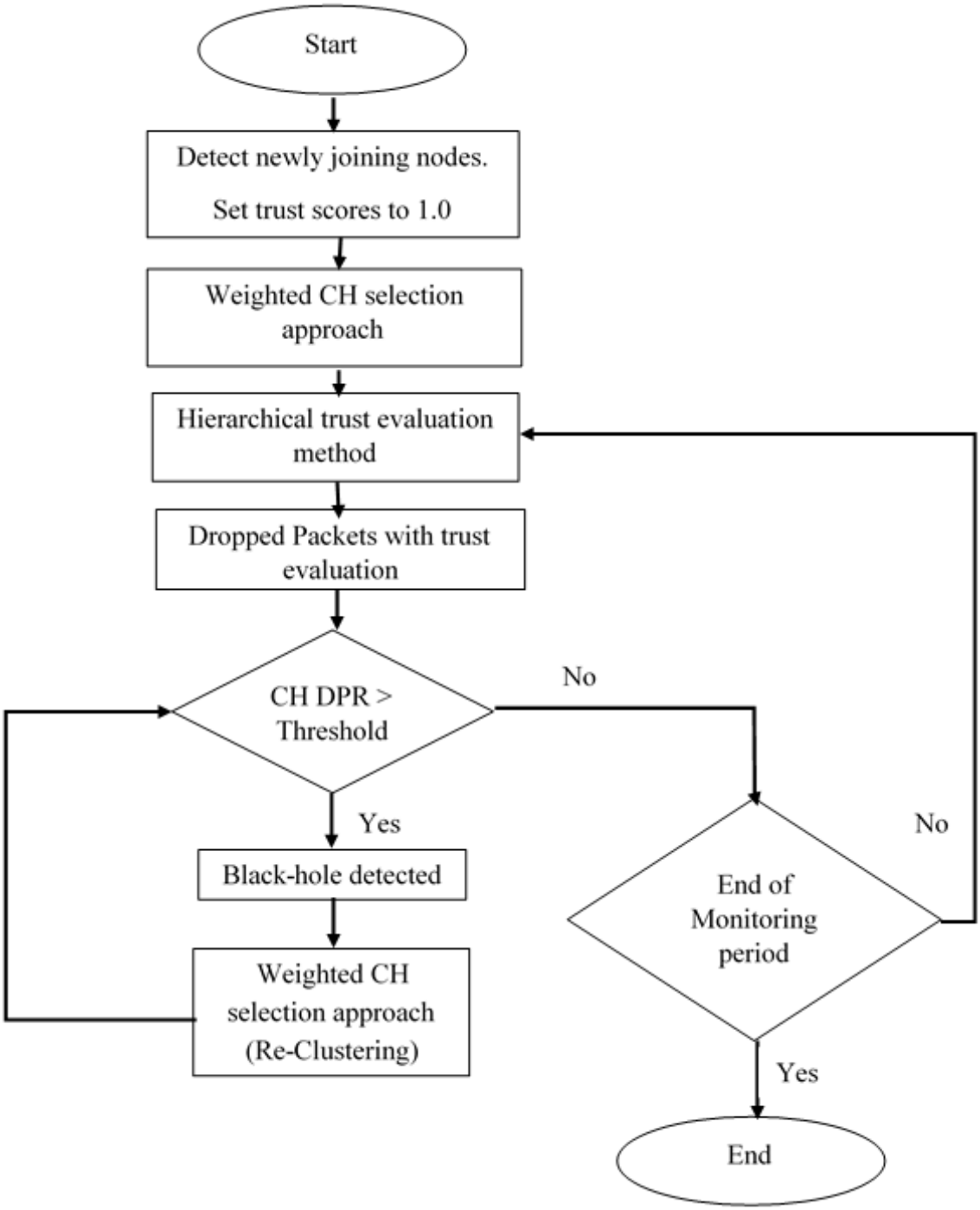

Figure 3.3: Black-Hole (B-H) detection flowchart

operations, all CHs needed re-clustering. After the forth re-clustering operation, the PDR of CH_2 has been minimized to a value less than the 0.2 while CH_0, CH_1 and

CH_3 needed a fifth re-clustering as represented in Figure 3.5.

- PDR threshold of 0.3: Choosing a threshold of 0.3 needed three re-clustering operations. As it is shown in Figures 3.6 and 3.7, the re-clustering operations started in the early periods before the end of simulation time. All CHs needed two re-clustering

Algorithm 1 Black-Hole attack detection

1: **procedure** B-H ATTACK DETECTION

Initialize: T_{score} , Th.

2: ▷ T_{score} is the trust evaluation between nodes n and m

3: ▷ Th is the pre-defined threshold

4: **Set:** $Th = 0.2, 0.3, 0.4, 0.5$

5: **Set:** $T_{score} = 1.0$

6: **for** each cluster **do**

7: Find CH using weighted cluster head selection technique

8: Perform hierarchical trust evaluation method

9: Check PDR

10: **if** (CH PDR >Th) **then**

11: **B-H detected**

12: **Re-clustering needed**

13: **else**

14: **if** *End of monitoring period* **then**

15: End

16: **else**

17: Go to *Hierarchical trust evaluation method*

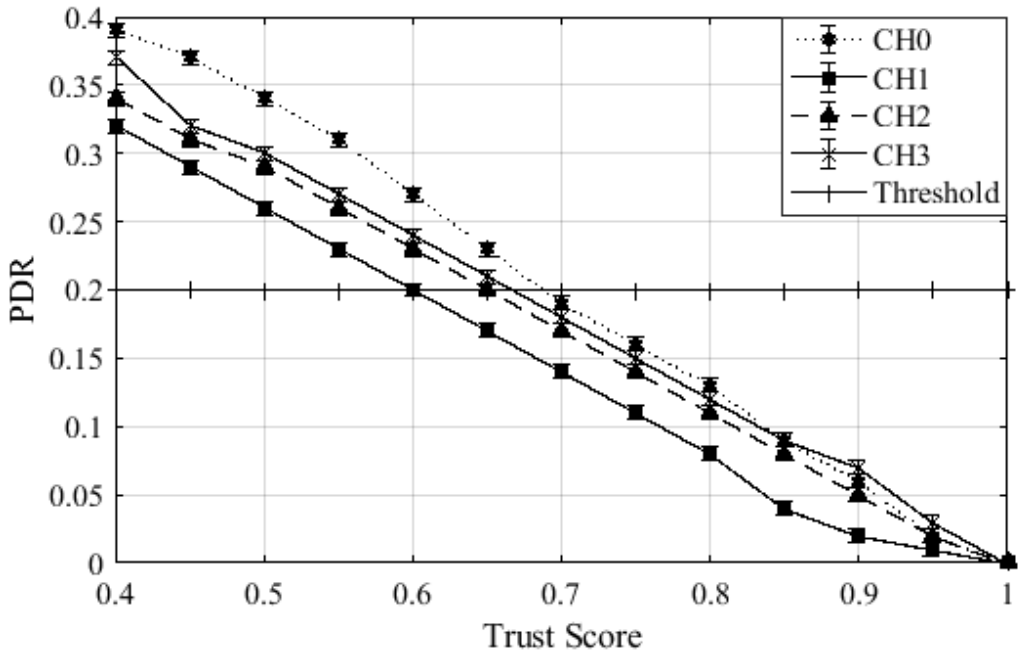

Figure 3.4: Pre-clustering with PDR threshold of 0.2

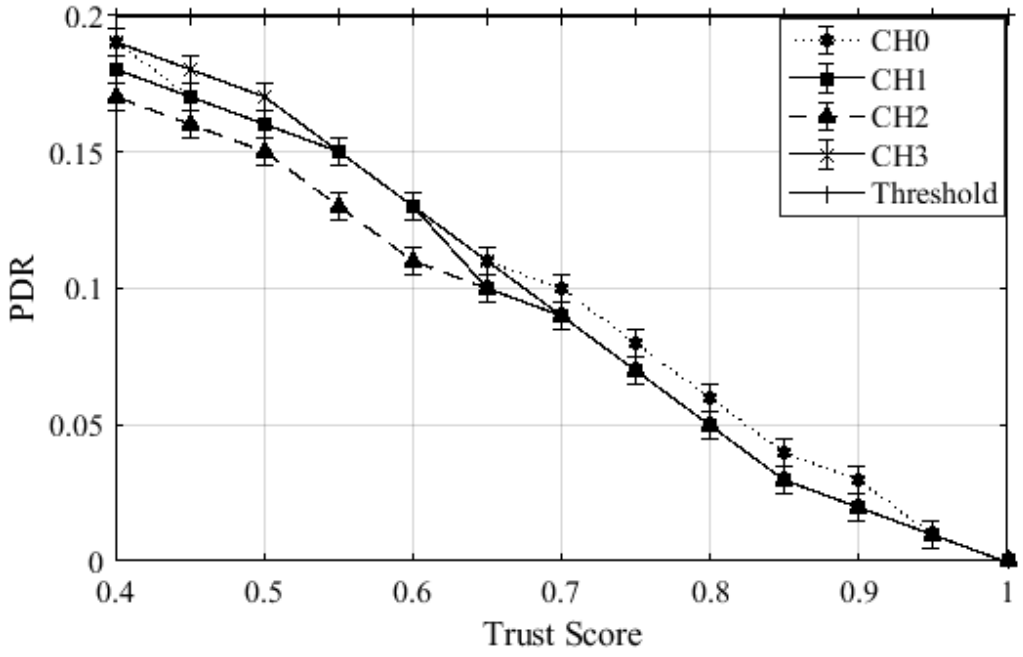

Figure 3.5: Post-clustering with PDR threshold of 0.2

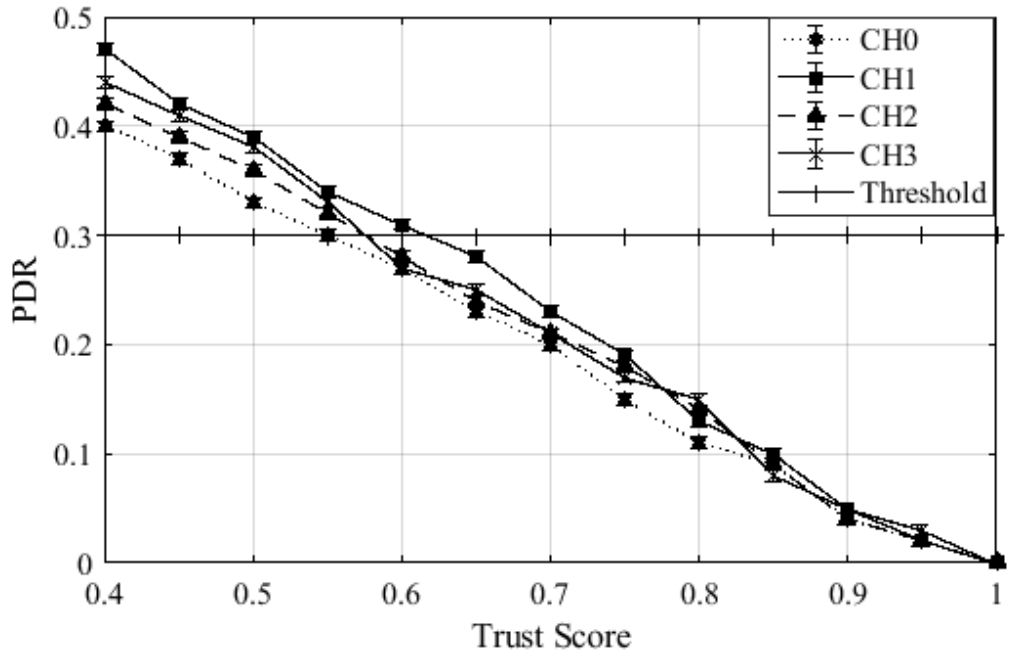

Figure 3.6: Pre-clustering with PDR threshold of 0.3

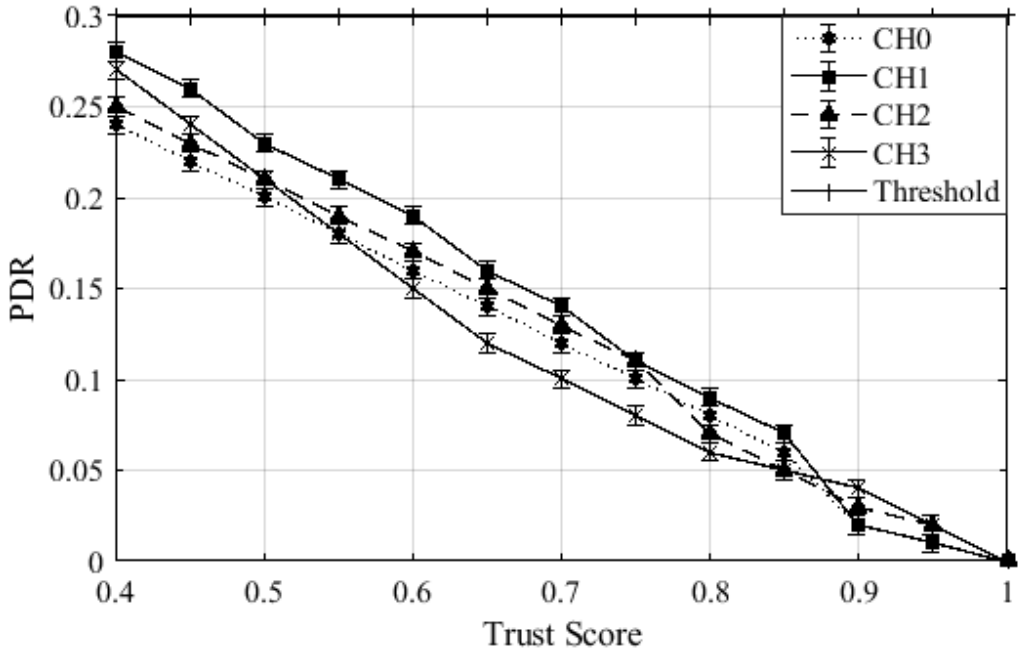

Figure 3.7: Post-clustering with PDR threshold of 0.3

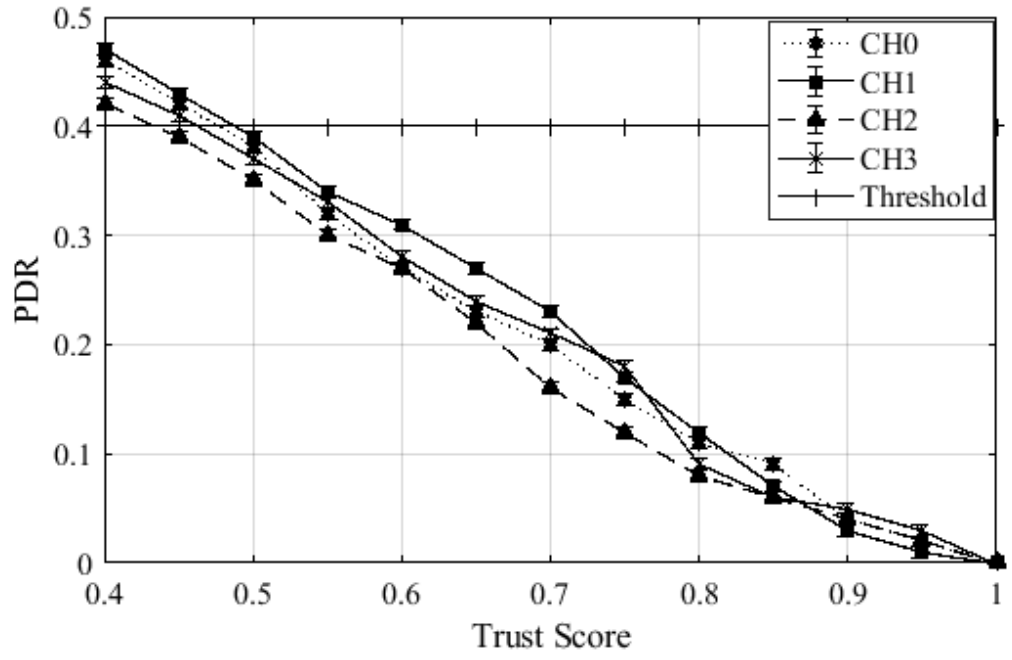

Figure 3.8: Pre-clustering with PDR threshold of 0.4

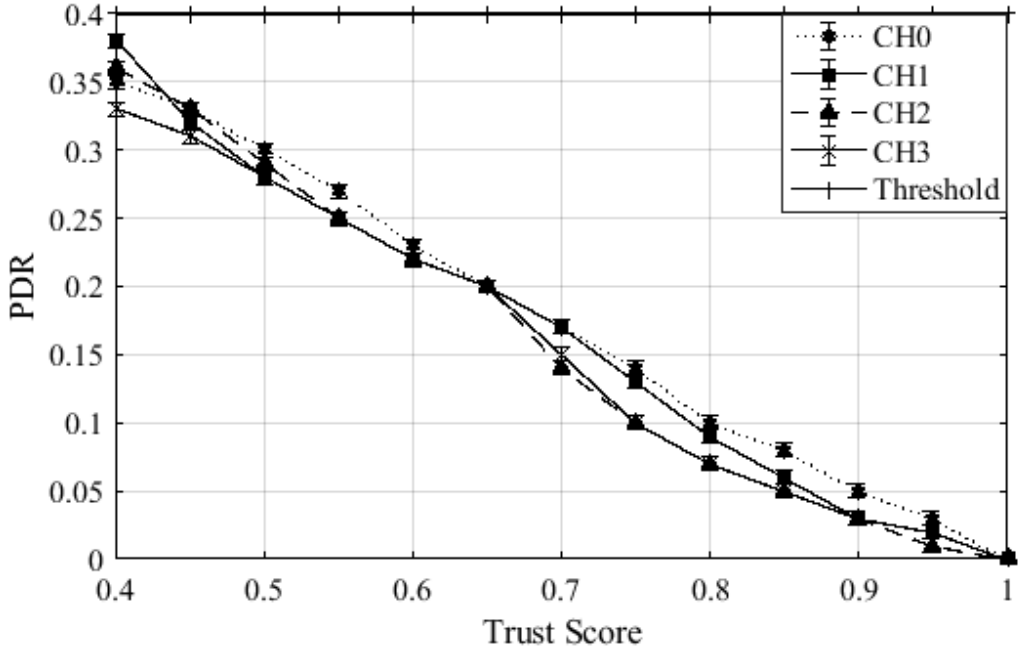

Figure 3.9: Post-clustering with PDR threshold of 0.4

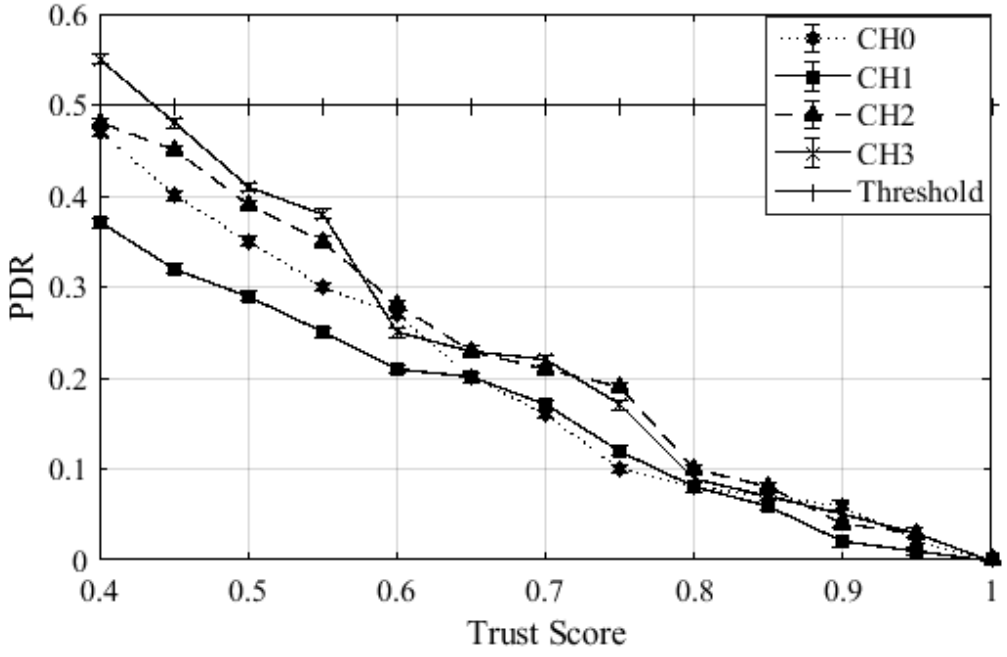

Figure 3.10: Pre-clustering with PDR threshold of 0.5

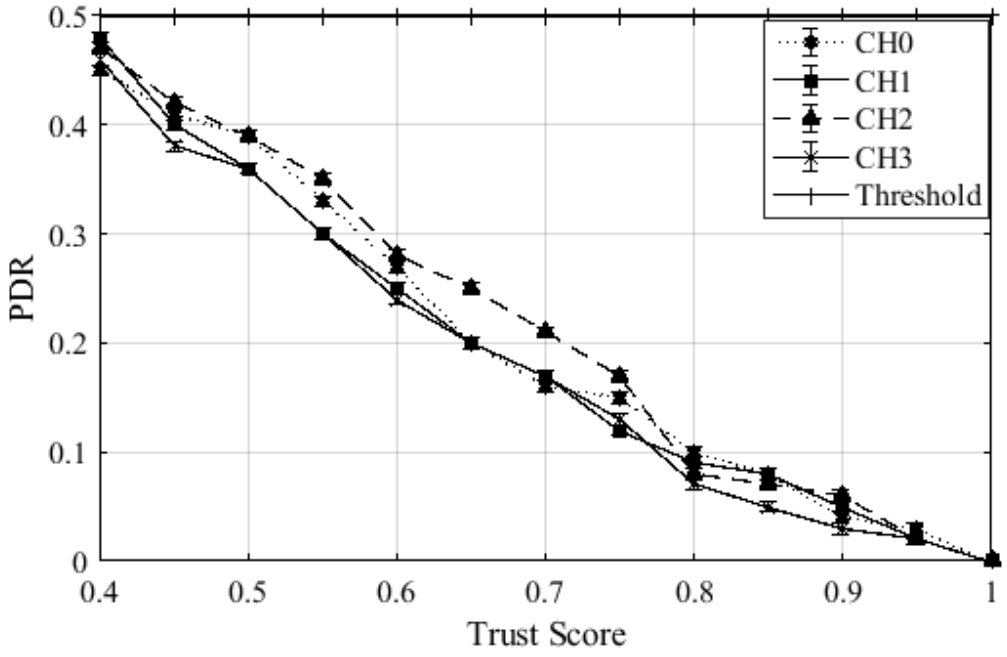

Figure 3.11: Post-clustering with PDR threshold of 0.5

operations in order to elect new CHs. After two re-clustering attempts, CH_1 and CH_3 needed a third re-clustering operation while the PDRs of CH_0 and CH_2 have reduced below 0.3 as shown in Figure 3.7. As a result, for the third re-clustering operation, all CHs have been approved as secured CHs.

- PDR threshold of 0.4: Three re-clustering operations have been taken place in order to achieve a B-H-proof environment. All CHs PDRs have been exceeded the 0.4 thresholds. As a result, all CHs have attempted the first re-clustering operation in order to elect four new CHs. After the first re-clustering, CH_0, CH_1 and CH_3 needed a second re-clustering operation due to their PDRs that exceeded the 0.4 threshold. Accordingly, for the second re-clustering, the PDRs of CH_0 and CH_3 have been reduced to less than 0.4 but CH_1 is still above the 0.4. The need for a third re-clustering attempt has arisen. Pre-clustering and post-clustering operations for PDR threshold of 0.4 are represented in Figures 3.8 and 3.9 respectively.
- PDR threshold of 0.5: As shown in Figure 3.10, CH_3 has dropped packet ratio (PDR) larger than the chosen threshold (Threshold=0.5). Accordingly, we have applied the re-clustering operation in order to elect another CH. The clusters formation after two re-clustering operations represents that all CHs stabilize under the 0.5. Choosing a threshold of 0.5 needed two re-clustering operations to achieve a B-H free environment as shown in Figure 3.11.

Tables 3.1, 3.2, 3.3, 3.4 and 3.5 represent the trust scores corresponding to the re-clustering operations.

Table 3.1: Trust scores corresponding to the first re-clustering operation where "NRN" refers to "No Re-clustering Needed"

PDR threshold	**Number of re-clustering**	CH_0	CH_1	CH_2	CH_3
0.20	5	0.69	0.60	0.65	0.66
0.30	3	0.55	0.61	0.57	0.57
0.40	3	0.47	0.49	0.43	0.46
0.50	2	NRN	NRN	NRN	0.44

Table 3.2: Trust scores corresponding to the second re-clustering operation where "NRN" refers to "No Re-clustering Needed"

Threshold	**Number of re-clustering**	CH_0	CH_1	CH_2	CH_3
0.20	5	0.56	0.55	0.60	0.61
0.30	3	0.51	0.56	0.49	0.52
0.40	3	0.44	0.45	NRN	0.41
0.50	2	NRN	0.41	NRN	NRN

Table 3.3: Trust scores corresponding to the third re-clustering operation where "NRN" refers to "No Re-clustering Needed"

Threshold	**Number of re-clustering**	CH_0	CH_1	CH_2	CH_3
0.20	5	0.52	0.55	0.50	0.47
0.30	3	NRN	0.47	NRN	0.40
0.40	3	NRN	0.42	NRN	NRN
0.50	2	NRN	NRN	NRN	NRN

Table 3.4: Trust scores corresponding to the forth re-clustering operation where "NRN" refers to "No Re-clustering Needed"

Threshold	**Number of re-clustering**	CH_0	CH_1	CH_2	CH_3
0.20	5	0.45	0.48	0.45	0.49
0.30	3	NRN	NRN	NRN	NRN
0.40	3	NRN	NRN	NRN	NRN
0.50	2	NRN	NRN	NRN	NRN

Table 3.5: Trust scores corresponding to the fifth re-clustering operation where "NRN" refers to "No Re-clustering Needed"

Threshold	**Number of re-clustering**	CH_0	CH_1	CH_2	CH_3
0.20	5	0.42	0.40	NRN	0.42
0.30	3	NRN	NRN	NRN	NRN
0.40	3	NRN	NRN	NRN	NRN
0.50	2	NRN	NRN	NRN	NRN

3.3 Machine Learning-based Intrusion Detection System (ML-IDS)

Machine learning techniques can be categorized into three main categories:

1. Supervised techniques: The system trained by a labelled dataset which helps in creating models for decision making. Supervised machine learning techniques are the techniques that predict the system behaviors based on pre-known behaviors in the training dataset.

2. Unsupervised techniques: In which the system creates the model of the unlabelled dataset without any pre-knowledge. Unsupervised machine learning techniques aim to detect the unlabeled data hidden structure. The clustering algorithm is an example of unsupervised machine learning techniques in which it detects malicious behaviors by categorizing data into clusters [97]. DBSCAN considered as clustering machine learning technique where the clustered data classified as normal and the data outside the clusters classified as abnormal data.

3. Semi-supervised techniques: Where the system trained by small labelled dataset and large unlabelled dataset in order to create the normal and abnormal data model.

This section presents our parallel intrusion detection model for WSN-based monitoring infrastructures. In the network under study, the aggregated sensory data undergoes two parallel intrusion detection subsystems (IDSs), namely the Anomaly Detection Subsystem (ADSs) for the unknown attacks and the misuse Detection Subsystem (MDSs) for the known ones, which refers to the Clustered Hierarchical Hybrid IDS (CHH-IDS). The aggregated sensory data (D) is distributed on the two detection subsystems, $IDSs1$ and $IDSs2$, following a time-slotted method in a round-robin fashion as shown in Figure 3.12. $IDSs1$ and $IDSs2$ refer to ADSs and MDSs respectively.

Aggregating the sensory data is the responsibility of the cluster head (CH) that collects the data from sensors and directs it to the IDS which is installed in a central node namely the server. The aggregated data then undergoes the intrusion detection method. The IDS consists of : 1) Weighted cluster head selection in Section 3.3.1, 2) Data aggregation in Section 3.3.2, 3) The detection methods in Sections 3.3.3, 3.3.4 and 3.4.

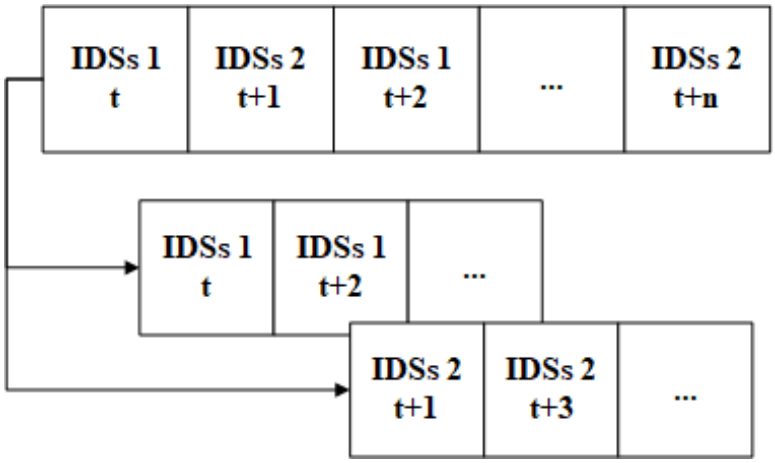

Figure 3.12: Aggregated sensed data distribution mechanism between $IDSs1$ and $IDSs2$.

The CHH-IDS proposed model consists of two subsystems, namely the ADSs and the MDSs. Each subsystem runs a different machine learning approach. These techniques are Random Forest (RF) and Enhanced Density-Based Spatial Clustering of Applications with Noise (E-DBSCAN). The MDSs adopts the Random Forest algorithm to classify network connections into intrusion and normal data based on a labeled training dataset while the ADSs adopts the E-DBSCAN algorithm, which is a density-based clustering technique that splits the network nodes into dense regions. The overall CHH-IDS is presented in Figure 3.13. Figure 3.14 represents the flowchart of the hybrid model for detecting intrusive sensors.

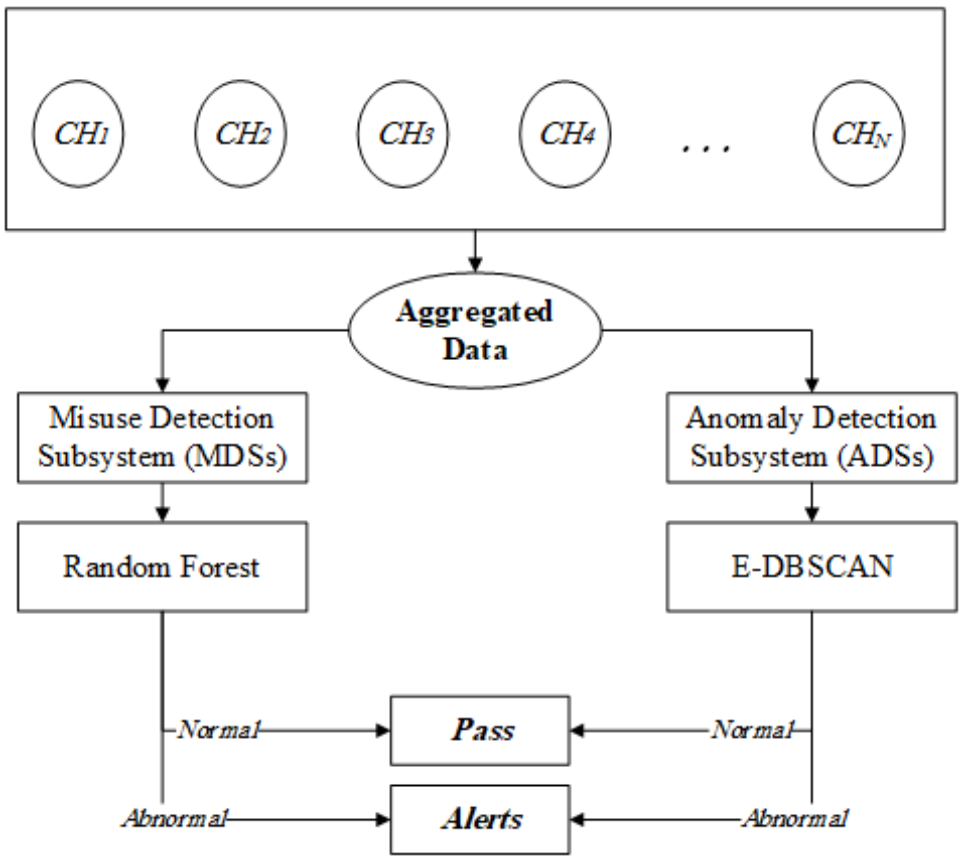

Figure 3.13: CHH-IDS system model where the collected sensed data is distributed between the ADSs and MDSs.

3.3.1 Cluster Head election technique

Cluster head (CH) election method is fulfilled by using the election technique that calculates the weight of each sensor node and compares it to the weights of the other nodes [98]. Each node has a weight G_n that is a function of its received signal strength, mobility, and degree. Following the weight computation, the node broadcasts its weight along with its unique identifier (ID). It then proceeds to compare it with the weights of the neighbour nodes such as the sensor node that achieves the lowest weight is elected as the CH[98][22][19]. The weight depends on the following four elements: Node mobility ($\check{M}_n$), degree difference (∂n) which refers to the node degree excluding CH capacity, cumulative

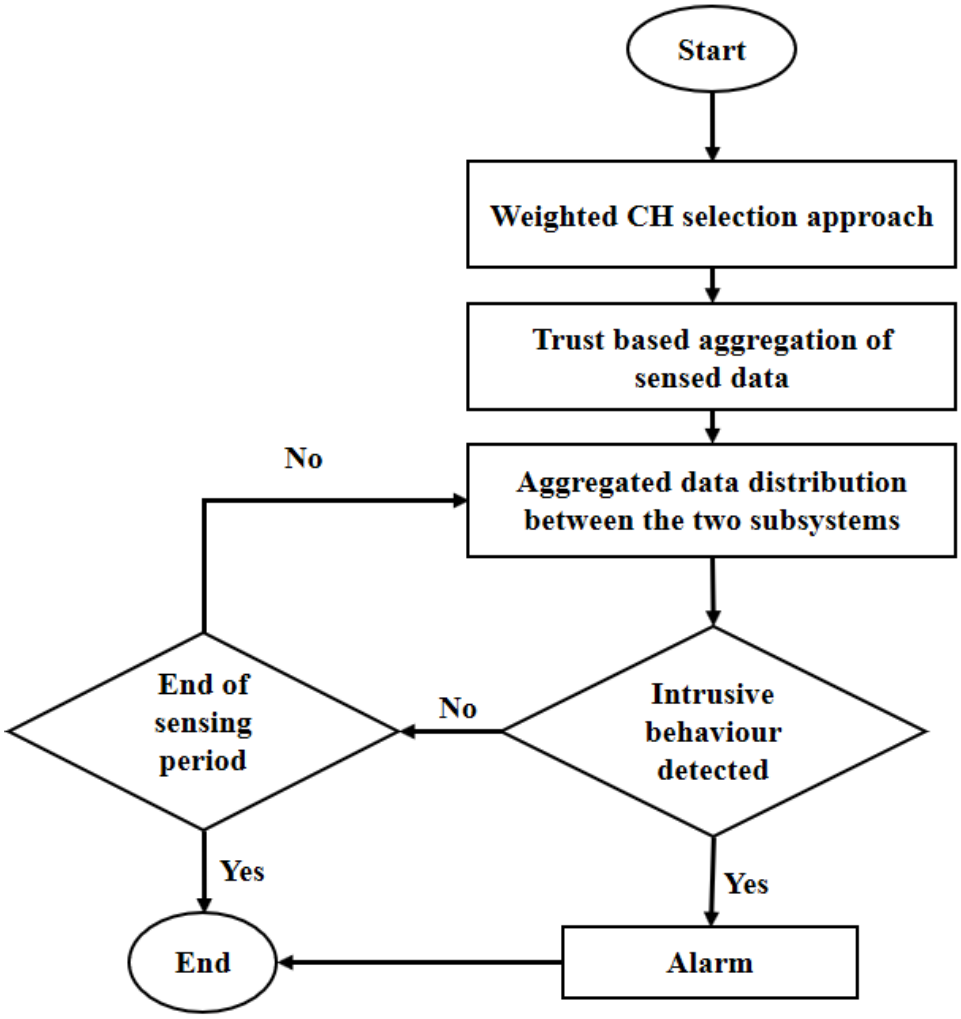

Figure 3.14: Flowchart for detecting intrusive sensors.

time ($\hbar_n$) and the sum of received signal strength ($SRS_{strength}$). The election procedure goes through the steps as shown in Algorithm 2.

3.3.2 Data aggregation procedure

Each CH gathers data from its corresponding sensors and sends them to the sink. The method presented in [99] has been adopted in our previous and ongoing research as the aggregation procedure. The method depends on assessing the trust score of the aggregator

by tracking the sensors' trust scores with the estimated trust between sensors and the aggregator [99]. Equation (3.9) computes the trust score of a CH [99] where T_{CH} denotes the CH trust value, T_n is node n trust value, and T_{CH}^n is the CH and node n trust evaluation.

$$T_{CH} = \frac{(\sum_{n=0}^{n-1}(T_n + 1) \cdot T_{CH}^n)}{\sum_{n=0}^{n-1}(T_n + 1)} \tag{3.9}$$

Algorithm 2 Weighted CH selection pseudocode

1: **for** each node n **do**

2: D_n:Degree of node n

3: Γ: Capacity of a CH *(i.e. number of nodes)*

4: ∂n: Degree difference for n

5: $\partial n = |D_n - \Gamma|$

6: $SRS_{strength}(n)$: Sum of n's received signal strength

7: $|1/SRS_{strength}(n)|$: normalized value of $SRS_{strength}$

8: $\check{M}_n$: Mobility factor for node n

9: $\hbar_n$: Cumulative time n

10: G_n: Combined weight for node n

11: $G_n = g_d \partial n + \frac{g_{sum}}{|1/SRS_{strength}(n)|} + g_m \check{M}_n + g_{\hbar_n} \hbar_n$

12: **Return** G_n

13: **Select** the node with a minimum G_n as CH

14: **Remove** CH from the nodes list

15: **Repeat** all steps for the remaining nodes

16: End

3.3.3 Anomaly Detection Subsystem (ADSs)

ADSs of the CHH-IDS runs the Enhanced-Density Based Spatial Clustering of Applications with Noise (E-DBSCAN) algorithm. DBSCAN is a density clustering algorithm in which it considers clusters as dense regions of objects in the data space that are separated by regions of low density objects [100]. It is one of the most recently used clustering algorithms. It has been updated to derive a new technique in calculating the threshold distance parameter which is a crucial parameter in the algorithm [101]. DBSCAN can discover clusters of random shapes. However, clusters that are close to each other belong to the same class [100]. DBSCAN is dependent on parameters provided by the users, and it is computationally expensive when applied on unbounded datasets. Such as with high dimensional datasets, the distance calculation with each instance will increase the computational cost, which makes it computationally expensive.

E-DBSCAN is used to perform a performance enhancement on the DBSCAN algorithm [101] [102].

DBSCAN consists of two factors, the ϵ and the $MinPts$ which considered as input parameters. It follows the rules descried below:

- $N_\epsilon(x) = y \in X | d(x, y)| \leq \epsilon$ is the $\epsilon - neighbourhood$ of point x.
- Neighbourhoods of core object has size $> MinPts$.
- A point j is density accessible from i as a core object.
- i and j considered as density-based connected when i and j are density accessible from a core object.

It is worthed mentioning that the reason behind using the E-DBSCAN is that it can find clusters with different densities while DBSCAN can find clusters with varying shapes and sizes. Furthermore, it selects suitable values of ϵ for each cluster. It depends on the local density of the starting point in each cluster and adopts the traditional DBSCAN for each

value of ϵ. In addition, E-DBSCAN depends on determining the highest density clusters first and then it adopts the value of ϵ in order to discover the next low-density clusters. The algorithm necessitates $Minpts$ and $Maxpts$ parameters. The $Maxpts$ parameter allows the ϵ value to be different from one cluster to another considering the density of the initial point in each one. The $Minpts$ and $Maxpts$ parameters estimate the lowest and the highest level of density allowed inside the cluster, respectively. Additionally, DBSCAN creates a cluster starting from any core point, while the E-DBSCAN creates the clustering from the highest density core point.

In E-DBSCAN, the Time Complexity ($T.C$) is derived from the number of region query requests [101]. DBSCAN algorithm performs one query for each point which gives a total run-time complexity $O(n)$ of $O(n) = (n.log(n))$.

In our analysis and simulation of E-DBSCAN, we found the following executions steps: The initialization step has been executed for one time (1). The comparison step has taken place for $(M + 1)$ times. The incremental step has been executed (M) times. The number of executions occurred, based on the previous analysis, is given in equation 3.10 or 3.11, respectively.

$$(OpNum) = 1 + (M + 1) + M \tag{3.10}$$

$$(OpNum) = 2 + 2M \tag{3.11}$$

The overall $T.C$ will be as in equation 3.12.

$$T.C(2 + (2M)) = O(m) \tag{3.12}$$

The integration of the E-DBSCAN into WSN-based critical infrastructures' monitoring system is described in Figure 3.15.

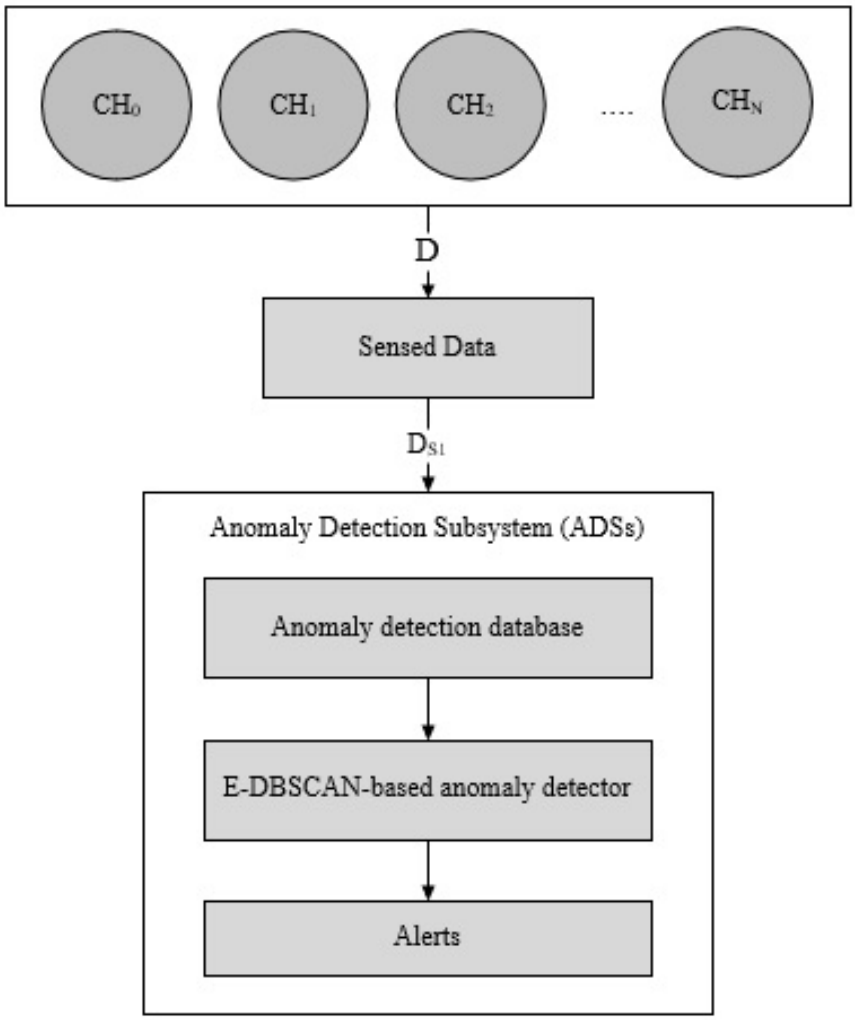

Figure 3.15: E-DBSCAN based anomaly detection procedure

3.3.4 Misuse Detection Subsystem (MDSs)

Our proposed signature detection method adopts the Random Forest (RF) algorithm as a supervised classification technique [103]. It works in two phases, the training phase and the classification phase. Training phase works offline in order to build the regular patterns and intrusion (i.e. anomalous) patterns by using the training dataset. A labeled training dataset is delivered after pre-processing operations into the intrusion pattern builder, as a result, it builds the detection module that needs the intrusive patterns. Classification phase

works online to detect intrusions based on the generated patterns from the training phase. In the classification phase, the network traffic is captured by the network aggregators and altered by pre-processing operations, and written into a network features database. In the end, the signature detector decomposes the features database into intrusive features database and normal features database by using the patterns generated in the training phase. An alarm is generated if an intrusion is detected. Diverse types of intrusions produce various network connections. Majority of the intrusions such as Denial of Service (DoS) produce more connections than non-major intrusions such as User to Root (U2R). This phenomenon leads to higher sensitivity in the detection of major intrusions whereas it does not work in favor of non-major ones. A balanced training dataset can be considered as a solution to this problem. The balanced training dataset is obtained by down-sampling the major intrusions and oversampling the non-major intrusions [103].

RF algorithm is a classification algorithm that consists of a collection of tree-structured classifiers, in which each tree sends a unit vote for the most popular class at each input [104]. Each tree is grown up as follows [103] [104]:

- If the size of the training set is Y, a sample population of size Y is taken randomly from the original dataset, and the sample population becomes the training set for growing the tree.
- If there are X input variables, x variables are selected randomly out of the X input variables. Then, the best split on these xs is used to split the node. The value of x is held constant during the forest growing.
- Each tree is grown to the largest extent possible.

RF Time Complexity ($T.C$) can be extracted from the decision tree complexity since RF considered as a special model of decision trees. The time complexity T.C for building a decision tree with r records and v variables is as shown in equation 3.13.

$$T.C(rrecords, vvariables) = O(r + vlog(v)) \tag{3.13}$$

In building our RF, we initiate the number of trees at the first step as Tr and the number of variables for each node as Var. The $T.C$ for building one tree will be calculated as in equation 3.14.

$$T.C(onetree) = O(Var * vlog(v)) \tag{3.14}$$

By introducing multiple trees, the $T.C$ will be as in equation 3.15.

$$T.C(multitrees) = O(Tr * Var * vlog(v) \tag{3.15}$$

If assuming $O(log(v))$ as the tree depth, the $T.C$ will be illustrated as in equation 3.16.

$$T.C(multitrees) = O(Tr * Var * v * depth) \tag{3.16}$$

RF-based signature detection is described in Figure 3.16.

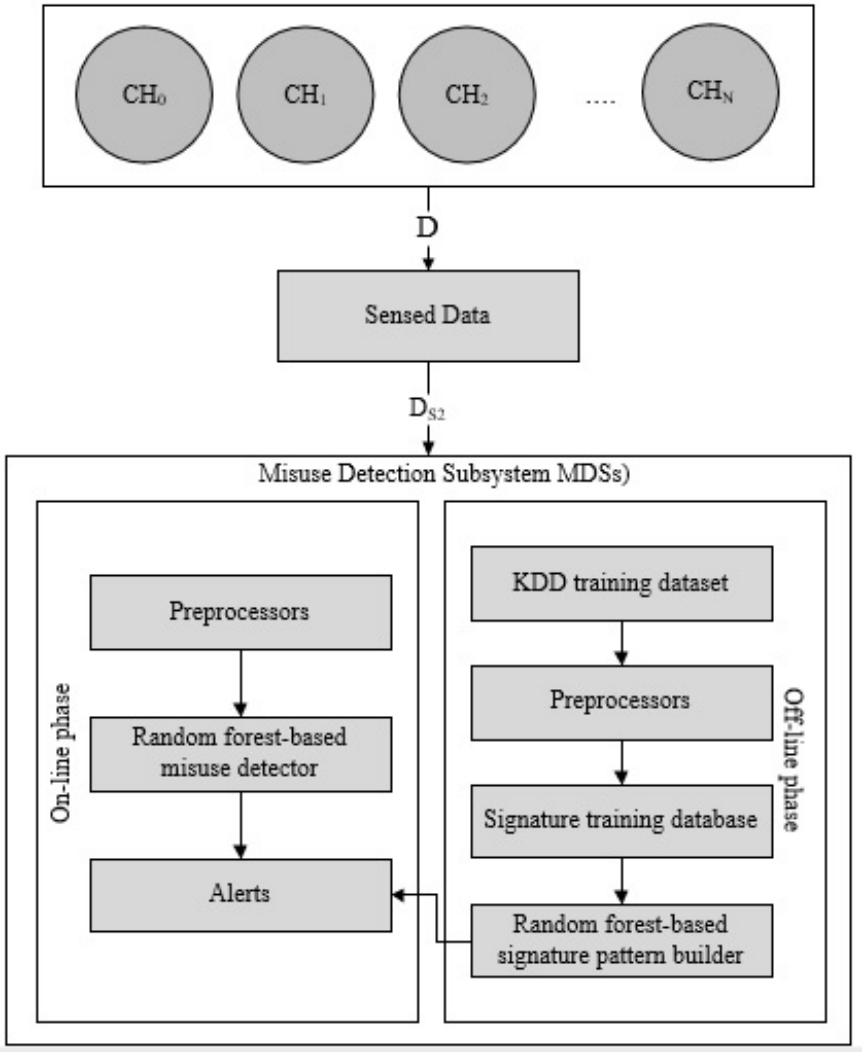

Figure 3.16: Random forest-based signature detection procedure

3.4 Deep Learning-based Intrusion Detection System (DL-IDS): Exploiting Restricted Boltzmann Machine

Deep Learning (DL) is a new area of Machine Learning (ML) research, which has been introduced with the objective of moving ML closer to one of its original goals: Artificial Intelligence (AI). According to [105], there are some reasons behind the recent DL prominent such as:

1. Processing abilities enhancement.
2. Computing hardware getting affordable with DL.
3. Recent breakthrough in ML research.

Usually, DL plays a vital role in image classification [106], in language, graphical modeling, pattern recognition, speech, audio, image, video, natural language and signal processing [91].

There are many DL techniques such as Deep Belief Network (DBN) [107], Boltzman Machine (BM), Restricted Boltzman Machine (RBM), Deep Boltzman Machine (DBM) [108], Deep Neural Network (DNN), Auto Encoder, Deep/stacked Auto Encoder [109], Stacked denoising Autoencoder [110], Distributed representation [91] and Convolutional Neural Network (CNN) [111].

3.4.1 Restricted Boltzmann Machine (RBM) Procedure

The RBM is a neural, energetic network with two layers: visible (V) and hidden (H). The learning procedure is managed by an unsupervised fashion [3]. The RBM permits

connections between neurons of the same layer, making it restricted. In RBM, W represents the weights between visible and hidden layers and W_{xy} represents the weight of both visible V_x and hidden H_y units. The energy function of the RBM is shown in Equation 3.17 below.

$$E(V, H|\Theta) = -\sum_{x=1}^{X} a_x V_x - \sum_{y=1}^{Y} b_y H_y - \sum_{x=1}^{X}\sum_{y=1}^{Y} V_x H_y W_{xy} \tag{3.17}$$

Θ refers to W_{xy},a_x,b_y (RBM parameters), a_x and b_y are the visible and hidden biases, and X and Y are the number of visible and hidden nodes.

The probability of (V, H) formation is calculated as in equation 3.18 [3].

$$P(V, H) = e^{-E(V,H)} / \sum_{X,Y} e^{-E(V_x,H_y)} \tag{3.18}$$

where $\sum_{X,Y} e^{-E(V_x,H_y)}$ refers to the normalization factor that represents all possible configurations, including the visible and hidden elements. With the energy function, the network allocates a probability score to each case in the hidden and visible elements. The probability allocated to a visible element V is presented in eq. 3.19[3].

$$P(V) = \sum_{Y} P(V_x, H_y) = \frac{\sum_Y e^{-E(V_x,H_y)}}{\sum_X \sum_Y e^{-E(V_x,H_y)}} \tag{3.19}$$

Likewise, the probability allocated to any hidden element H is presented in eq. 3.20 below [3].

$$P(H) = \sum_{X} P(V_x, H_y) = \frac{\sum_X e^{-E(V_x,H_y)}}{\sum_X \sum_Y e^{-E(V_x,H_y)}} \tag{3.20}$$

All visible and hidden elements in RBM are conditionally independent, as shown in eq. 3.21 and 3.22 below [3].

$$P(V|H) = \prod_{x=1}^{V} P(V_x|H) \tag{3.21}$$

$$P(H|V) = \prod_{y=1}^{H} P(H_y|V) \tag{3.22}$$

The representation of the binary output for intrusive and normal detection is presented in eq. 3.23 and eq. 3.24 below.

$$P(Intrusive) = P(O = 0|V) \tag{3.23}$$

$$P(Normal) = P(O = 1|V) \tag{3.24}$$

The RBM Intrusive and Normal outputs refer to O_1 and O_2 in Fig.3.17.

3.4.2 Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS)

The proposed Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS) consists of N clusters with C sensor nodes in each cluster. In each cluster, the cluster head (CH) is in charge of sending the sensor directed data to the IDS, which is installed in a central server. The aggregated data then undergoes deep learning-based Restricted Boltzmann Machine IDS, namely the RBC-IDS. Figure 3.17 represents RBC-IDS when the RBM method consists of input layer that contain x visible nodes, such as(V_1, V_2,...., V_x), hidden layers and the outputs.

The RBM permits connections between neurons of the same layer, which makes it restricted, and the procedure is presented in the pseudocode in Algorithm 3.

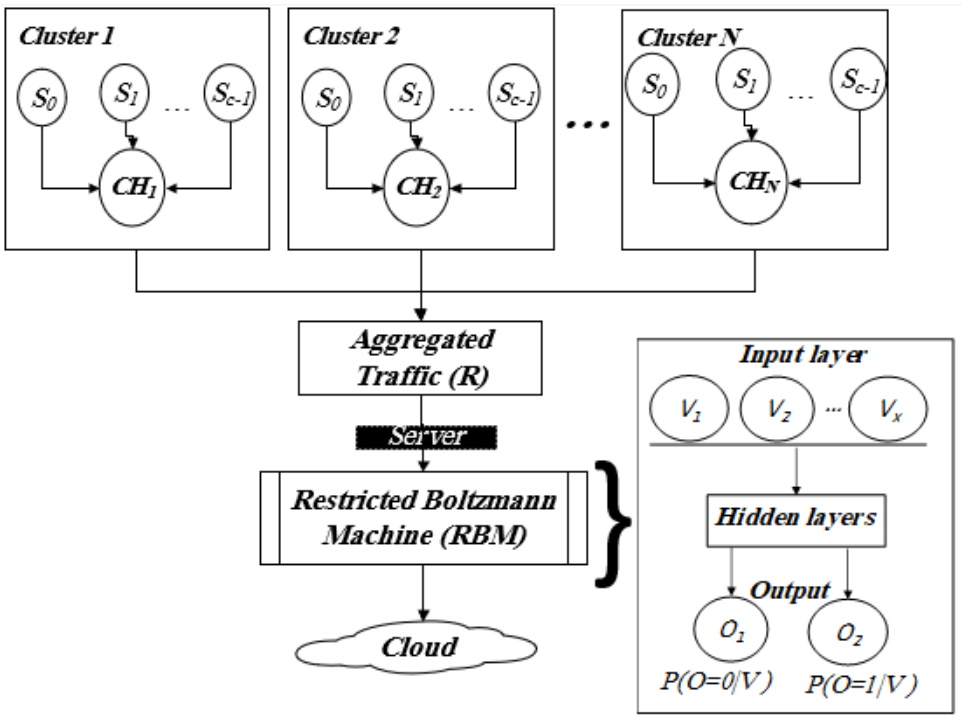

Figure 3.17: Deep learning module (Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS))

The network sets a probability score to each case in hidden and visible elements[2][3].

Fig. 3.18 represents the used RBM setting in RBC-IDS. The RBC-IDS consists of an input layer contains X visible nodes, three hidden layers, and an output layer with two outputs O_1 and O_2 for $Intrusive$ and $Normal$ outputs respectively. W_{11} represents the weight between the first visible layer and the first hidden layer while W_{12} refers to the weight of the first and the second hidden layers and W_{23} is the weight between the second and the third hidden layers.

In the RBC-IDS, each CH is responsible for aggregating the sensed data from the sensors in the same cluster and forwards them to the server by adopting the procedure in [99].

Algorithm 3 Restricted Boltzmann Machine (RBM) Procedure

procedure RBC-IDS

Initiate: W_{xy} , a_x, b_y, X , Y.

for $y = 1, 2, 3, ..., Y$ (All hidden layers) **do**

for $x = 1, 2, 3, ..., X$ (All visible layers) **do**

Compute $E(V, H|\Theta) = -\sum_{x=1}^{X} a_x V_x - \sum_{y=1}^{Y} b_y H_y - \sum_{x=1}^{X} \sum_{y=1}^{Y} V_x H_y W_{xy}$

Compute $P(V, H) = \frac{e^{-E(V,H)}}{\sum_{X,Y} e^{-E(V,H)}}$

Compute $P(V) = \sum_Y P(V, H) = \frac{\sum_Y e^{-E(V,H)}}{\sum_X \sum_Y e^{-E(V,H)}}$

Compute $P(H) = \sum_X P(V, H) = \frac{\sum_X e^{-E(V,H)}}{\sum_X \sum_Y e^{-E(V,H)}}$

Compute $P(V|H) = \prod_{x=1}^{V} P(V_x|H)$ **Compute** $P(H|V) = \prod_{y=1}^{H} P(H_y|V)$

Compute $P(Intrusive) = P(O = 0|V)$

Compute $P(Normal) = P(O = 1|V)$

End for

End for

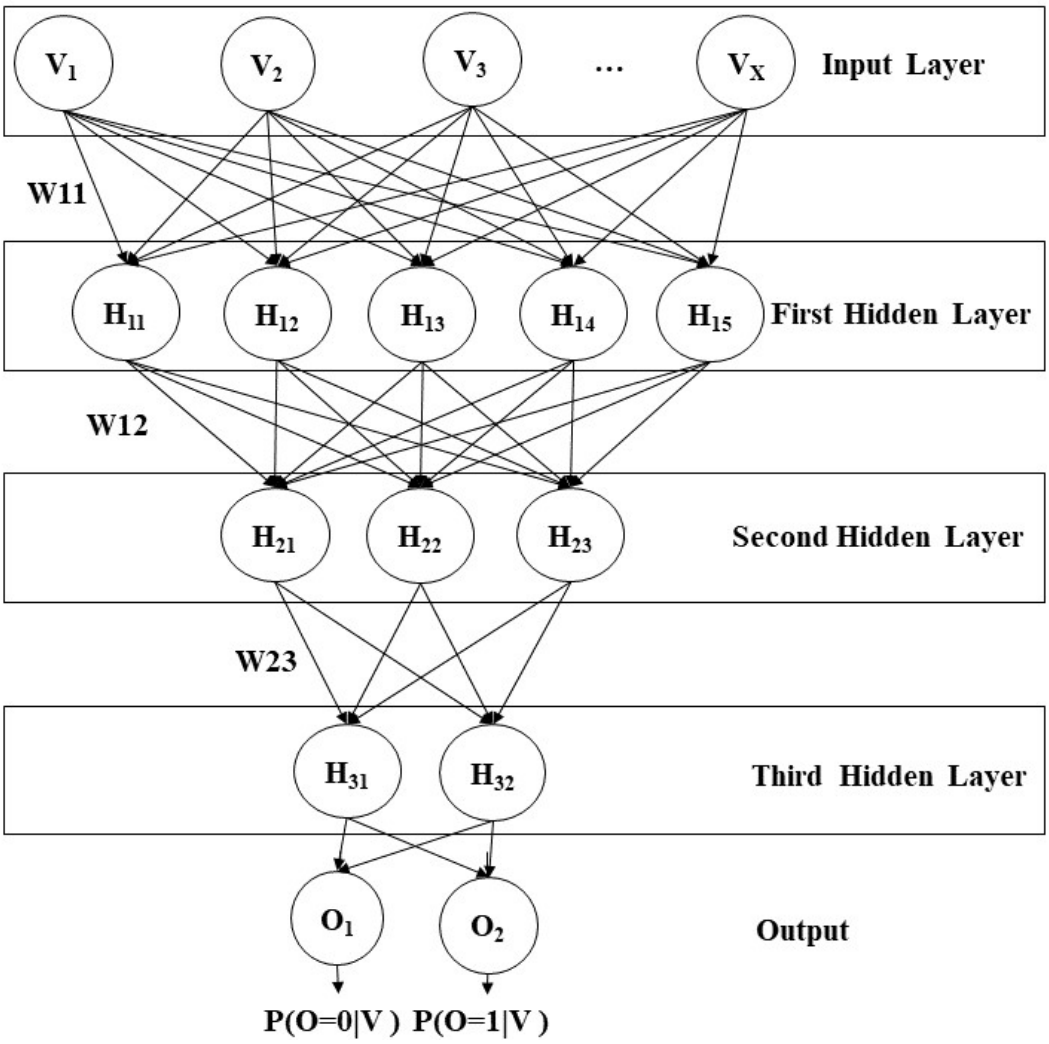

Figure 3.18: RBM used in RBC-IDS which contains one visible layer and 3 hidden layers and one output layer. W_{11} refers to the weight between the visible layer and the first hidden layer, W_{12} refers to the weight between the first and second hidden layers and W_{23} refers to the weight between the second and third hidden layers. O_1 and O_2 are the *Intrusive* and *Normal* outputs.

3.5 Comparison of ML-IDS (CHH-IDS) and DL-IDS (RBC-IDS) based solutions

3.5.1 Simulation Settings and Results

- **Experiments Environment:** CHH-IDS and RBC-IDS for WSN-aided critical infrastructures have been tested in order to achieve highest *Detection Rate* (DR), highest *Accuracy Rate (AR)*, minimum *False Negative Rate (FNR)*, highest F_1 *score*, highest *precision-recall* and biggest *area under curve (ROC)* in which these factors can decide the acceptability of any intrusion detection system. We have evaluated the performance of CHH-IDS and RBC-IDS on NS-3 simulator [96]. We have considered a network of 20 sensors adopt the Hierarchical-Dynamic Source Routing (H-DSR) protocol. The sensors make 4 clusters that spread out in a 100mx100m area. We have run each simulation scenario 10 times, and in the figures, we present the average of these runs with 95% confidence level. The simulation settings along with the assumptions are summarized in Table 3.6.

- **KDD CUP 99 Data set:** The KDD (Knowledge Discovery in Data mining) CUP 1999 Dataset is used to validate the efficiency of the CHH-IDS and RBC-IDS [112]. The KDD CUP 1999 intrusion detection dataset helps designers of IDS with evaluating different methodologies. Attacks in KDD CUP 99 are categorized under four types as follows:

  - **Denial of Service (DoS)**: Attacker tries to prevent users from using a service by making some computation or denning legitimate users access to systems.

  - **Remote to Local (R2L)**: Attacker does not have access to the attacked machines, therefore tries to gain access.

Table 3.6: Testing settings for methods in sections 3.3, 3.4

Simulation Inputs	**Input Value**
Visible nodes (x)	41
Hidden layers	3
Number of sensor nodes	20
Routing protocol	Hierarchical-DSR (H-DSR)
Number of clusters	4
Simulation time	600s
Packet size	250 bytes
Operational area	100m x 100m
Communication range	100m
Trust range	[0,1]
Dataset	KDD'99
Attack Types	DoS,R2L,U2R,Probe

- **User to Root (U2R)**: Attacker has local access to the attacked machine but attempts to have extra privileges.
- **Probe**: Attacker tries to gain information about the host by scanning a machine or a networking device in order to determine vulnerabilities that may be exploited later.

The KDD CUP 1999 intrusion detection dataset consists of three components, which are written in detail in Table 3 in the International Knowledge Discovery and Data Mining Tools Competition, 10% of KDD dataset is hired for the purpose of training. It covers 22 attack types and is a sub set version of the Whole KDD dataset.

Because of their nature, denial of service attacks accounts for the majority of the dataset. On the other hand, the Corrected KDD dataset provides a dataset with different distributions other than 10% KDD nor Whole KDD and covers 14 additional attacks. Since 10% KDD is employed as the training set in the original competition, the analysis of hybrid IDS was performed on the 10% KDD dataset. To carry the experiments effectively, KDD CUP 1999 dataset containing connection records with varying distribution of attack types and normal class has been used in our proposed hybrid IDS. It is worth mentioning that the proportion of data in the testing dataset is not the same as the training dataset ones, and the test data includes some specific type of attacks which are not in the training set. The KDD data set description is shown in Table 3.7 whereas the 22 attacks in training data set are classified in Table 3.9. KDD'99 features and their descriptions are listed in Table 3.8 [1].

Table 3.7: KDD data set description

KDD Data set	**DoS**	**R2L**	**U2R**	**Probe**
Whole KDD	3883370	1126	52	41102
Corrected KDD	229853	16347	70	4166
10% KDD	391458	1126	52	4107

Table 3.8: KDD'99 Features Set [1]

Index	**Feature Name**	**Description**
1	Duration	Connection length (s)
2	Protocol	Type of the protocol
3	service	Network service on the destination (ex. http)
4	Flag	Normal or error status
5	dst_bytes	Number of bytes from destination to source

6	src_bytes	Number of bytes from source to destination
7	wrong_fragment	Number of wrong fragments
8	land	1 if the connection is from/to the same host or 0 otherwise
9	urgent	Number of urgent packets
10	hot	Number of hot indicators
11	num_failed_logins	Number of failed login attempts
12	logged in	1 if successfully logged in or 0 otherwise
13	num_compromised	Number of compromised conditions
14	root_shell	1 if the root shell is obtained or 0 otherwise
15	su_attempted	1 if su root command attempted or 0 otherwise
16	num_root	Number of root accesses
17	num_file_creations	Number of file creation operations
18	num_shells	Number of shell prompts
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound commands in an FTP session
21	is_host_login	1 if the login belongs to the hot list or 0 otherwise
22	is_guest_login	1 if the login is a guest login or 0 otherwise
23	count	Number of connections to same host
24	srv_count	Number of connections to the same service
25	serror_rate	Number of connections have SYN errors
26	srv_serror_rate	Number of connections have SYN errors
27	rerror_rate	Number of connections have REJ errors
28	srv_rerror_rate	Number of connections have REJ errors
29	same_srv_rate	Number of connections to the same service
30	diff_srv_rate	Number of connections to different services
31	srv_diff_host_rate	Number of connections to different hosts

32	dst_host_count	Number of connections with same destination host IP address
33	dst_host_srv_count	Number of connections with same port number
34	dst_host_same_srv_rate	% of connections to the same service, among the connections aggregated in dst_host_count
35	dst_host_diff_srv_rate	% of connections to different services, among the connections aggregated in dst_host_count
36	dst_host_same_src_port_rate	% of connections to the same source port, among the connections aggregated in dst_host_srv_count
37	dst_host_srv_diff_host_rate	% of connections to different destination machines, among the connections aggregated in dst_host_srv_count
38	dst_host_serror_rate	% of connections have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_count
39	dst_host_srv_serror_rate	% of connections have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count
40	dst_host_rerror_rate	% of connections have activated the flag (4) REJ, among the connections aggregated in dst_host_count
41	dst_host_srv_rerror_rate	% of connections have activated the flag (4) REJ, among the connections aggregated in dst_host_srv_count

- CHH-IDS Results

  Our proposed IDS is evaluated based on the following criteria's:

  - **True Positive (TP)**: Are the anomalous cases that were correctly classified as abnormal.
  - **False Positive (FP)**: Are the normal cases that were incorrectly classified as anomalous.

Table 3.9: Attacks in KDD'99 dataset

Main Attack classes	**22 Attacks**	**Number of attacks**
DoS	Smurf,land,pod,Neptune,teardrop,back	6
U2R	Perl,rootkit,buffer_overflow,loadmodule	4
R2L	Imap,guess_passwd,multihop,phf, ftp_write,spy,warezmaster,warezclient	8
Probe	Nmap,portsweep,satan,ipsweep	4

– **True Negative (TN)**: Are the normal cases that were classified correctly.

– **False Negative (FN)**: Are the anomalous cases that were incorrectly classified as normal.

1. ***Accuracy Rate (AR)***

   The accuracy rate is the first performance metric we have used to evaluate the proposed hybrid intrusion detection system on critical infrastructure sensors. Accuracy rate refers to the percentage of the correctly classified instances, which are also denoted by True Positives (TP) and True Negatives (TN) [113] as shown in equations 3.25.

$$AR = (TP + TN)/(TP + TN + FP + FN) \tag{3.25}$$

   Figure 3.19 illustrates the accuracy rates for the anomaly detection subsystem, signature detection subsystem, and the overall hybrid intrusion detection system on the sensors. As shown, the hybrid model achieves the highest accuracy rate with an overall of 98.95%. Anomaly detection achieves better overall accuracy

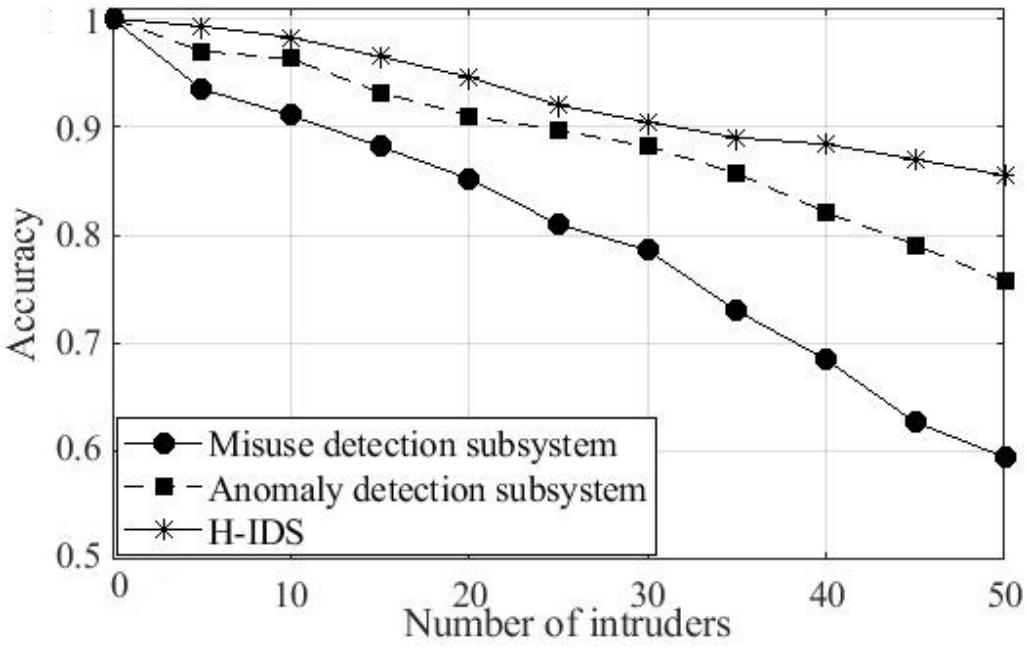

Figure 3.19: Accuracy rate comparison between anomaly, misuse and the hybrid (H-IDS)

rate since signature detection performs with the least accuracy rate. On the other hand, the signature detection subsystem helps in increasing the overall detection rate as shown in the next subsection.

2. ***Detection Rate (DR)***

The detection rate denotes the ratio of sensor behavior that is truly classified as intrusive. In other words, the detection rate denotes the true positive ratio. as formulated in Equation 4.12.

$$DR = TP/(TP + FP) \tag{3.26}$$

Figure 3.20 illustrates the detection rates for the anomaly detection subsystem, signature detection subsystem, and the overall hybrid intrusion detection system on the sensors. The proposed hybrid model leads to the highest detection rate

in detecting the sensors that are in intrusive behavior when compared to each of the individual anomaly detection and signature detection subsystems. Since anomaly detection results in the lowest detection rate, incorporation of the signature detection subsystem via E-DBSCAN helps in improving the detection rate.

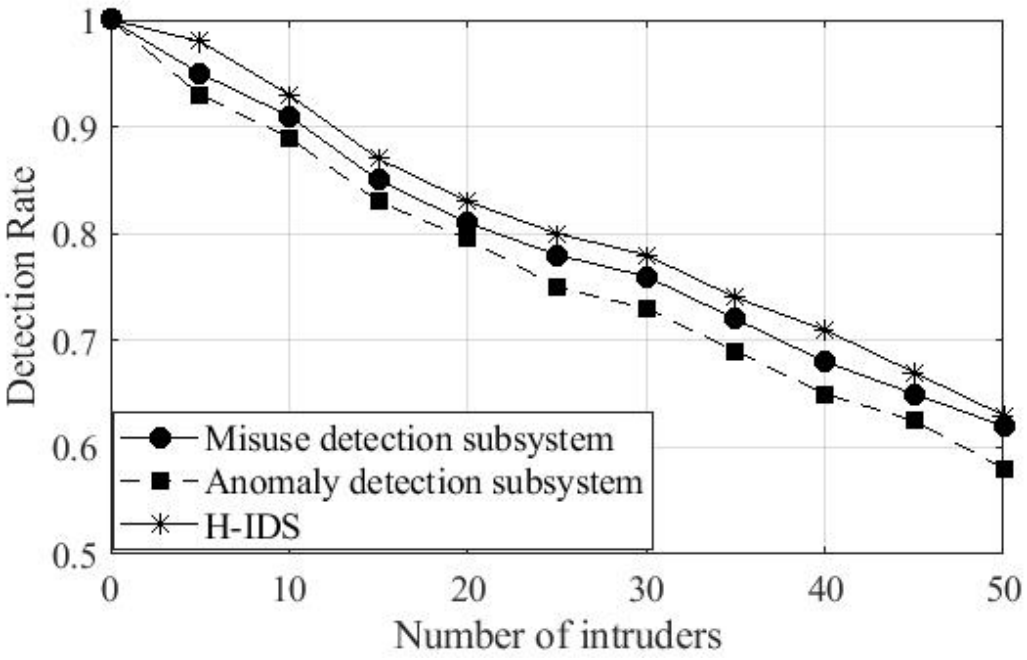

Figure 3.20: Detection rate comparison between anomaly, misuse and the hybrid (H-IDS)

3. ***False Negative Rate (FNR)***

False Negative (FN) refers to the percentage of intrusive sensor behavior, which has inaccurately been classified as non-intrusive, as formulated in equation 3.27.

$$FNR = FN/(TP + FN + FP + TN) \tag{3.27}$$

Where FN and TP are the False Negative and True Positive cases respectively. False negative is used to define a networks failure to detect intrusive sensor behavior under certain situations. In other words, malicious activity originated by

the sensors are not detected or alarmed although and an alert should have been raised. In our tests, we set 0.5 of data to be directed to the anomaly detection subsystem while the other 0.5 is directed to the signature detection module. The false negative based on this setting is shown in Fig. 3.21. By integrating anomaly detection with signature detection, the overall false negative rate has been reduced when compared to the case under the signature detection subsystem as anomaly detection via random forest algorithm is capable of detecting known attacks which in turn can reduce false negative intruder decisions on sensed data.

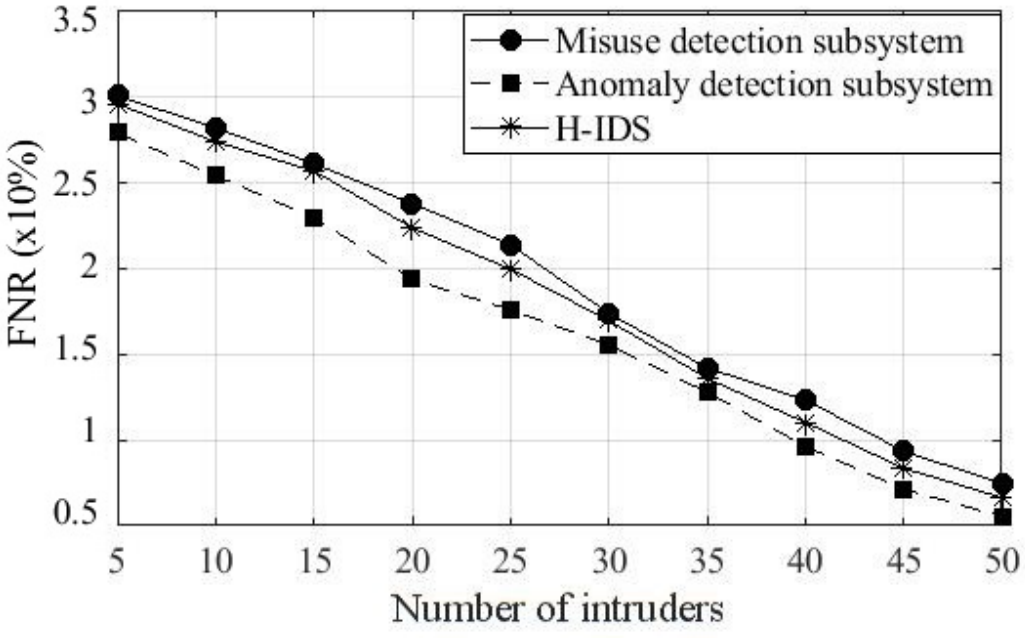

Figure 3.21: False Negative rate comparison between anomaly, misuse and the hybrid (H-IDS)

4. ***Impact of clustering sensors***

We also seek the benefit of using hierarchical network topology through clustering sensors. To implement a non-hierarchical topology, the system in Figure

3.13 is slightly modified by appointing every sensor as the cluster head of a one-node cluster.

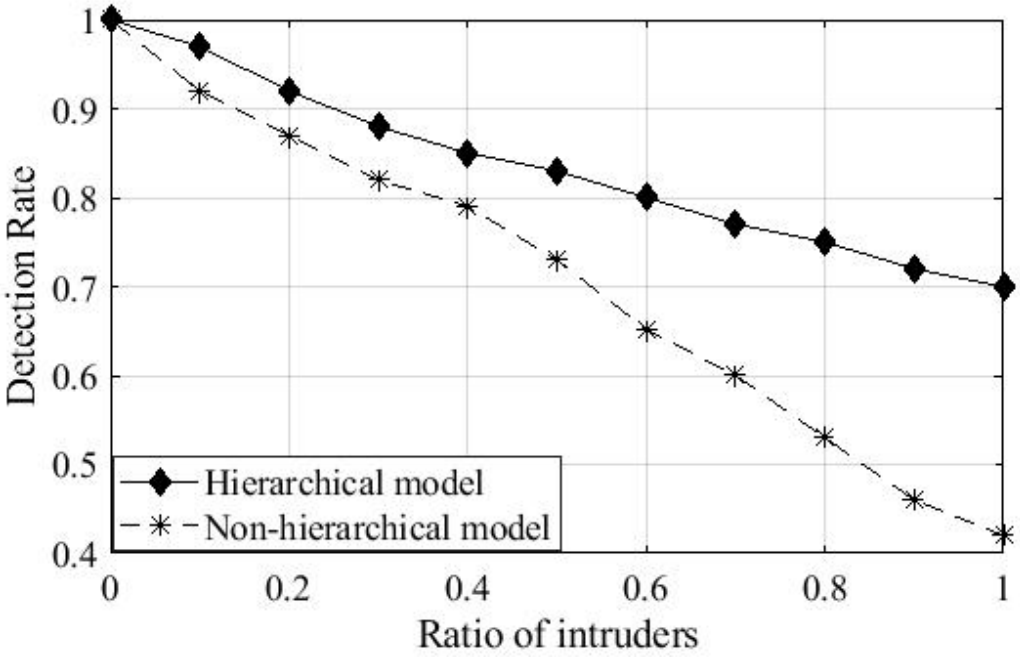

Figure 3.22: Detection rate under hierarchical topology and non-hierarchical topology

As seen in Figure 3.22, by clustering in a hierarchical topology, the detection rate under a non-hierarchical solution can be reduced by 6% (under 5% intruder ratio) and by $> 45\%$ (under 50% intruder ratio). The reason for this behavior is that the hierarchical topology enables trust evaluation for the cluster heads which results in trust score-based data aggregation. Thus, the sensed data that undergoes the H-IDS has already been fused with a certain trust score.

5. ***Receiver Operating Characteristic (ROC)***

Receiver Operating Characteristic (ROC) represents the relationship between the TP rate (Sensitivity) and the FP rate (1-Specificity) for different cut-off points. The larger the area under the curve, the better the sensitivity versus specificity trade-off performance is. Figure 3.23 confirms that the proposed

H-IDS is capable of providing accurate detection.

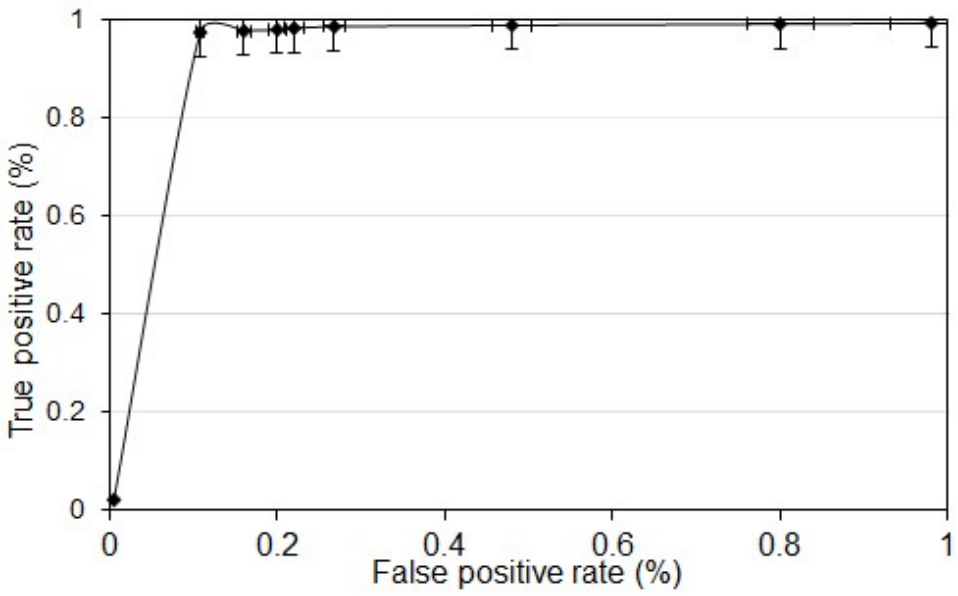

Figure 3.23: Receiver operating characteristic (ROC) curve for H-IDS

- ML-IDS vs. DL-IDS

  Figures 3.24 and 3.25 represent the comparison between using the DL-IDS with different number of hidden layers ($H = 1, H = 2$ and $H = 3$), ML-IDS (CHH-IDS), Misuse Detection Subsystem (MDSs) and Anomaly Detection Subsystem (ADSs), considering the accuracy and detection ratios.

  As it is shown in Figure 3.24, the proposed DL-IDS with (H=3) achieves the highest accuracy of $\approx 99.91\%$ followed by ML-IDS with an accuracy of $\approx 98.948\%$.

  Considering the detection ratio, DL-IDS with (H=3) achieves with the highest detection of $\approx 99.12\%$ compared to the ML-IDS which performs with $\approx 99.731\%$ detection ratio as shown in Figure 3.25.

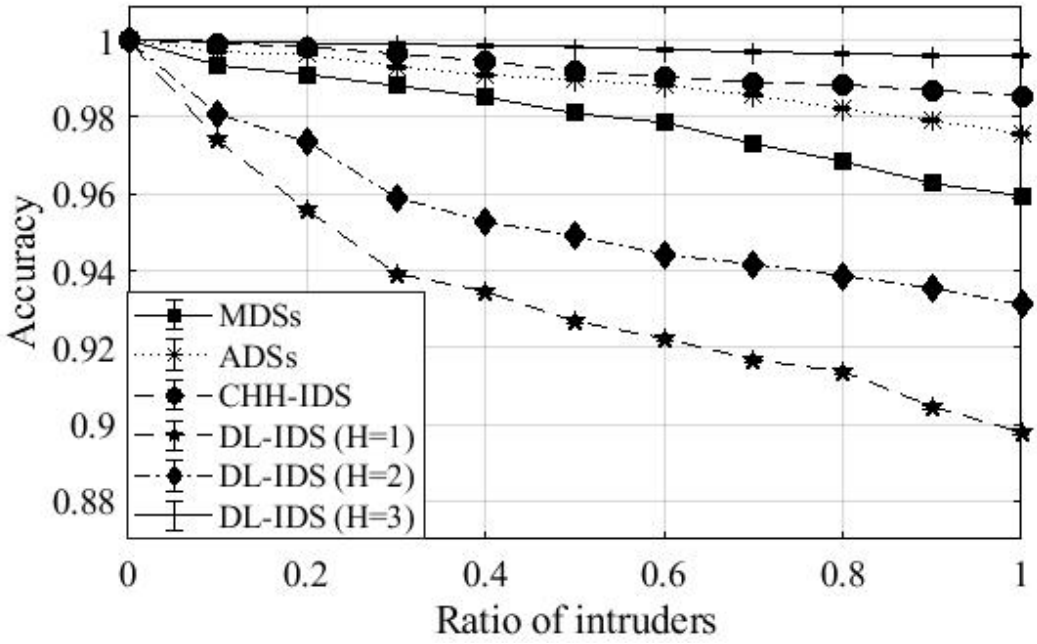

Figure 3.24: Accuracy comparison of DL-IDS with different numbers of hidden layers (H_1, H_2 and H_3) vs. CHH-IDS, MDSs and ADSs

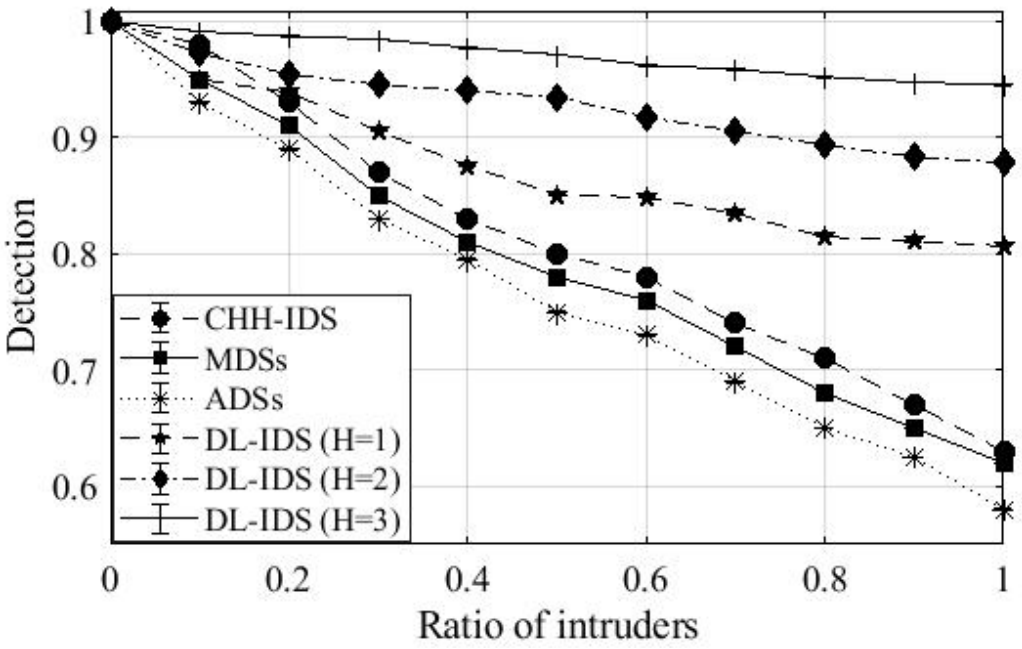

Figure 3.25: Detection comparison of DL-IDS with different numbers of hidden layers (H_1, H_2 and H_3) vs. CHH-IDS, MDSs and ADSs

3.6 Summary

One of the most important issues in monitoring-based critical infrastructures is lying down the security issues. In this chapter, we have proposed a Black Hole (B-H) detection system where the B-H CHs have been detected and replaced by new CHs in order to achieve a B-H free environment which helps in forwarding the transmitted information securely and without the fear of information lost. We tested the system model behaviours by choosing different thresholds (0.5, 0.4, 0.3 and 0.2) and found that the threshold of 0.5 has been approved to reach the system stability in early periods with the least number of re-clustering operations.

False negatives (FNs) may lead to severe consequences in such settings; however, our proposed solution can also reduce FNs that occur under the solely employed anomaly detection mechanism.

On the other hand, we have proposed a new hybrid method to detect sensors that are in intrusive behavior while monitoring critical infrastructure. The proposed methodology consolidates the advantages of anomaly-based and misuse-based intrusion detection and uses a trust-based hierarchical framework to aggregated sensor data. In the intrusion detection system, the anomaly detection subsystem employs the E-DBSCAN technique whereas the misuse detection subsystem employs the random forest mechanism. Through simulations, we have shown the effectiveness of the proposed approach by injecting real attack patterns into wirelessly networked sensors. ML-IDS performs with an accuracy of 98.95% with up to 99.73% detection rate.

Through simulations, we have verified that DL-IDS (RBC-IDS) outperforms ML-IDS (CHH-IDS) with $\approx$ 99.12% detection and $\approx$ 99.91% accuracy in the existence of intrusive behaviour in the tested WSN.

Chapter 4

An Adaptive Machine Learning (AML)-based IDS for WSNs

4.1 Introduction

With the wide usage and deployment of Wireless Sensor Networks (WSNs) and their integration with the Internet of Things concept, WSNs have been recognized as robust tools to meet the requirements for monitoring critical infrastructures such as smart grid, healthcare, and/or military applications. WSNs employ various types of sensors, i.e., thermal and magnetic which help in monitoring the different aspect of systems such as pressure and temperature [114][115].

In the long term and continuous monitoring of these critical infrastructures, detection of malicious traffic activity (i.e. intrusion) has become of paramount importance.

Majority of the existing intrusion detection solutions rely on various data mining methods. These models have been verified to be very effective [116][117][118]. Although there has

been remarkable progress in intrusion detection and prevention research, sensor networks monitoring critical infrastructures are still vulnerable to unknown attacks.

In this chapter, we aim to address the detection of known and unknown intrusive behavior at the sensory data aggregation stage of WSN-based critical infrastructure monitoring systems. To this end, we propose an adaptive intrusion detection system (Adaptive-IDS), namely Adaptively Supervised and Clustered Hybrid Intrusion Detection System (ASCH-IDS) to classify the aggregated data.

In ASCH-IDS, data gathered by sensors is directed into two machine learning-based subsystems namely misuse detection subsystem and anomaly detection subsystem. The former is effective in the detection of known attacks whereas the latter is effective in detecting unknown attacks. The Misuse Detection Subsystem (MDS) runs a random forest-based classifier to detect known attacks. The classifier basically compares the upcoming sensed traffic to attack patterns that are known from the training data to identify intrusive behavior.

The Anomaly Detection Subsystem (ADS), on the other hand, employs an Enhanced-DBSCAN classifier to detect unknown attacks by comparing sensed data to normal patterns in training data-set. The key question here is the following: How to decide the destination subsystem for an aggregated data stream? Can a probabilistic routing scheme be used for forwarding the data to one of these subsystems for analysis?

In our proposed solution (i.e., ASCH-IDS), we address these issues by adaptive supervision of our previously proposed Clustered Hierarchical Hybrid-Intrusion Detection System (CHH-IDS) [19]. The proposed scheme continuously keeps track of the Receiver Operating Characteristics (ROC) in each subsystem, and based on the improvement/degradation of the ROC behavior, it adaptively adjusts the proportion of aggregated data forwarded to one of the two subsystems. Our simulations on real attack data demonstrate up to 99% detection rate and up to 99.80% overall accuracy.

Here, in this chapter, we start by listing some related works in section 4.2. Then, we present our proposed adaptive intrusion detection system (IDS) in Section 4.3. In section 4.4, we present the performance comparison. Finally, we summarize the chapter in Section 4.5.

4.2 Related Work

All smart devices are vulnerable to different types of attacks. So, the need for developing an Intrusion Detection System (IDS) dedicated to monitor such devices, adapt to different changes in the environment, and be able to detect all malicious activities has arisen. Some works had been proposed for adaptively detection of various attacks, such as the works presented in [119], [120] and [121].

In [119], the authors proposed Pulse which is a novel IDS for the Internet of Things (IoT) that adapted Machine Learning (ML) techniques and successfully identified some attacks such as probing and simple forms of Denial of Service (DoS) attacks. A Naive Bayes classier had been used in their proposed model.

The authors in [120] proposed a Knowledge-Based Intrusion Detection Strategy (KBIDS) in order to detect different attacks over different networks. They started by adopting an unsupervised mean shift clustering algorithm to decide the clusters and then they decided the anomaly clusters if they had a certain amount of deviation from the normal clusters. The decision function for deciding the detected behaviours was built by using the weighted Support Vector Machine (SVM). The decision function was updated periodically by adding the new features in order to adapt to the network variability and to achieve the time efficiency [120].

The authors in [121] compared several approaches to intrusion detection in sensor networks by investigating the detection and accuracy rates and comparing them with the networks' energy efficiency.

4.3 Adaptively Supervised and Clustered Hybrid IDS (ASCH-IDS)

The proposed IDS framework for WSNs builds on our previous proposal, Clustered Hierarchical Hybrid-IDS (CHH-IDS) [19], which is illustrated in a minimalist way in Fig. 4.1.

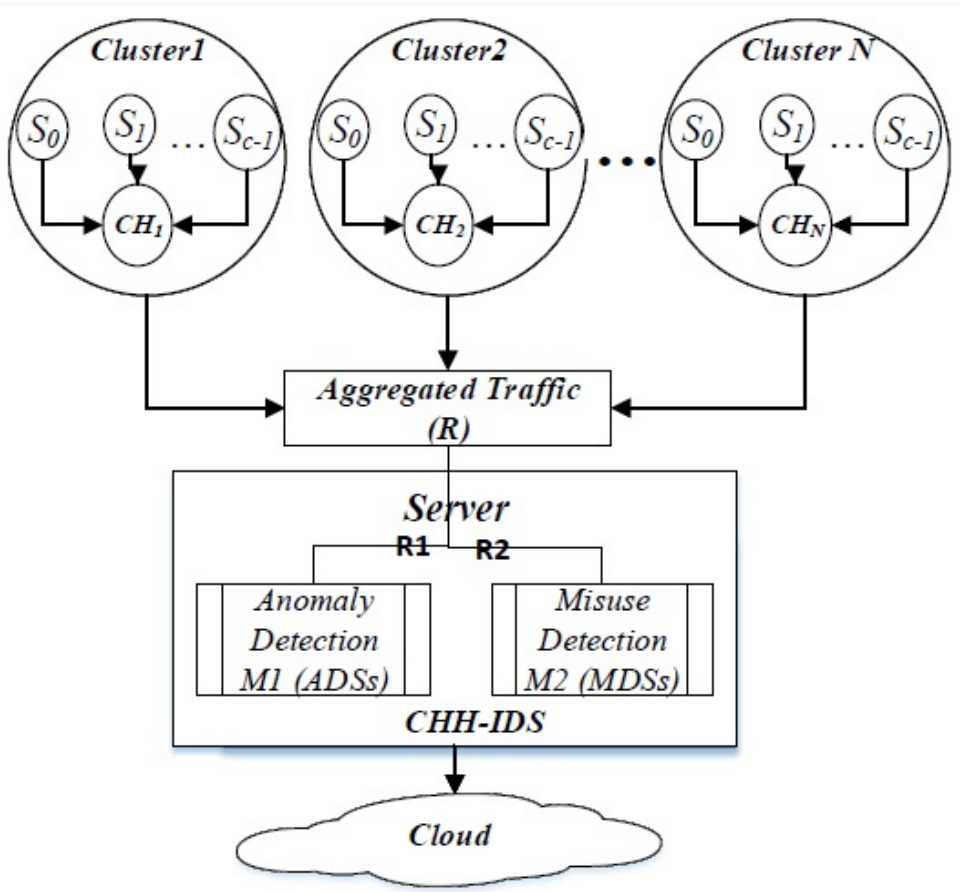

Figure 4.1: A minimalist illustration of the system model. The cluster head aggregates sensed data from sensor cluster nodes. The aggregated data is distributed between the anomaly detection and misuse detection subsystems.

Like its predecessor, ASCH-IDS operates on a clustered WSN that consists of N clusters

each of which is made up of C sensor nodes. In each cluster, a cluster head (CH) is responsible for aggregating the data forwarded by the sensor nodes. Upon aggregation of the sensed data, the CH forwards the data to a central server where the IDS is actually deployed. ASCH-IDS, as well, adopts the weighted cluster head selection algorithm represented in Section 3.3.1, in which the CH is selected based on the comparison of each sensor weight with the other nodes inside its cluster.

In ASCH-IDS, as CHH-IDS [19], each CH aggregates the sensory data from the other sensors in its corresponding cluster and sends the aggregated data to the centralized sink. The data aggregation method in [122] had been used in CHH-IDS. It measures the aggregator's trust score based on the trust score of each sensor along with the trust evaluation between the aggregator and the sensors [122]. We used the function in (4.1) [122] in CHH-IDS in order to calculate the trust score of CHs which are represented as the aggregators. In the equation, T_{agg} is trust value of the aggregator, T_n is the trust value of node n, and T_{agg}^n is the trust evaluation between the aggregator and node n.

$$T_{agg} = \frac{(\sum_{n=0}^{n-1}(T_n + 1) \cdot T_{agg}^n)}{\sum_{n=0}^{n-1}(T_n + 1)} \tag{4.1}$$

In CHH-IDS, the aggregated traffic undergoes two parallel intrusion detection subsystems, namely the ADSs for the unknown attacks and the MDSs for the known ones, which refers to a hybrid system. ADSs of the CHH-IDS runs the Enhanced-Density Based Spatial Clustering of Applications with Noise (E-DBSCAN) algorithm. DBSCAN is a density clustering algorithm in which it studies clusters as dense areas of objects in the data space that are divided by areas of low density objects [123]. MDSs in CHH-IDS uses the random forest algorithm as a controlled classification method which works in two phases; the training and the classification phases [124]. It is basically a classification algorithm consists of tree-structured classifiers collection where each tree sends a unit vote for the most common class at each input [104].

ASCH-IDS aims to keep track of the changes in the Receiver Operating Characteristics (ROC) of the misuse and anomaly detection subsystems, and adaptively adjusts the proportion of the sensed data forwarded to one of these two subsystems. The True Positive (TP) to False Positive (FP) ratios for ADSs and MDSs at time t_i are denoted by $M_1(t_i)$ and $M_2(t_i)$ as shown in Eqs. (4.2)-(4.3) below.

Algorithm 4 presents a detailed overview of the ASCH-IDS.

$$M_1(t_i) = \frac{TP_1(t_i)}{FP_1(t_i)} \tag{4.2}$$

$$M_2(t_i) = \frac{TP_2(t_i)}{FP_2(t_i)} \tag{4.3}$$

As ASCH-IDs is proposed for real time operation, the TP/FP ratio is kept track as a running average value with time steps ($\triangle t$) as shown in Eqs. (4.4)-(4.5). It is worth noting that $\triangle t$=$t_{i+1} - t_i$.

$$M_1(\triangle t) = \frac{TP_1(\triangle t)}{FP_1(\triangle t)} \tag{4.4}$$

$$M_2(\triangle t) = \frac{TP_2(\triangle t)}{FP_2(\triangle t)} \tag{4.5}$$

When the ROC behavior of the two subsystems during the time step $\triangle t$ is obtained, the overall ROC behavior per subsystem can be calculated as a weighted sum of the current overall ROC behavior and the behavior during the time step as shown in Eqs. (4.6) and (4.7) shown below. In the equations, the α parameter denotes the weight of the overall TP/FP value that has been calculated so far and the TP/FP value during the time step ($\triangle t$) where $t_{i+1} = t_i + \triangle t$ as described before.

$$M_1(t_{i+1}) = \alpha M_1(t_i) + (1 - \alpha) M_1(\triangle t) \tag{4.6}$$

$$M_2(t_{i+1}) = \alpha M_2(t_i) + (1 - \alpha) M_2(\triangle t) \tag{4.7}$$

Besides the ROC behavior in each subsystem, ASCH-IDS also keeps track of the relative running average ROC behavior of the two subsystems at any time t_i. To this end, an

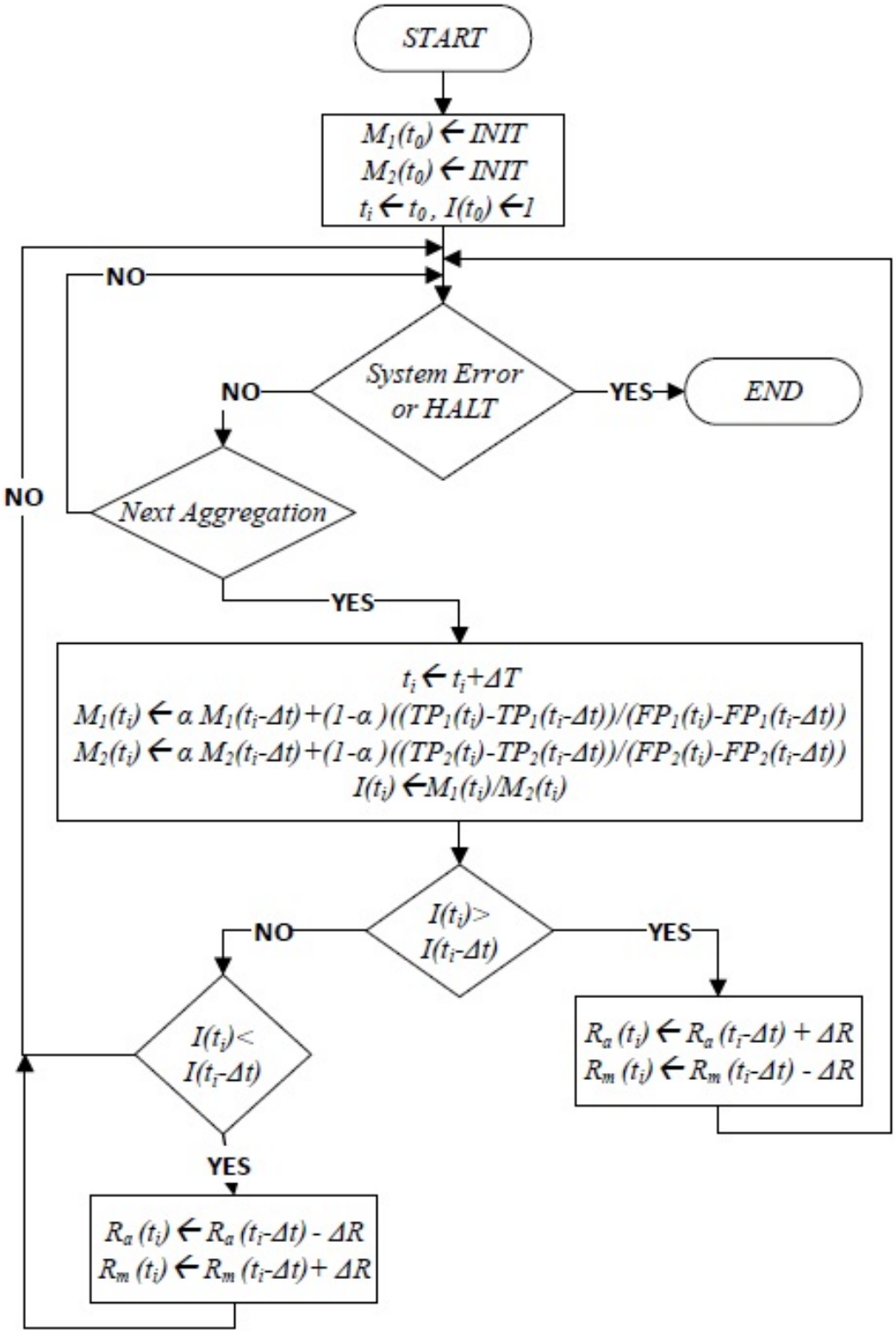

Figure 4.2: ASCH-IDS Flowchart for a single decision making process.

indicator $I(t_i)$ is introduced as shown in Eq.(4.8).

$$I(t_i) = \frac{M_1(t_i)}{M_2(t_i)} \tag{4.8}$$

The relative ROC behavior of the two subsystems is utilized in the decision of forwarding aggregated sensory data as follows: At time t_i, if $I(t_i) > I(t_{i-1})$, ASCH-IDS interprets this situation as better performing of the anomaly detection subsystem when compared to the performance of the misuse detection subsystem. Thus, increasing the sensed data proportion on the anomaly detection subsystem is expected to be beneficial for improving the overall performance. On the other hand, if $I(t_i) < I(t_{i-1})$, ASCH-IDS interprets the situation as better performing of the misuse detection subsystem when compared to the anomaly detection subsystem.

In this case, the intuition is that increasing the data proportion on the misuse detection subsystem will help the IDS system improve the overall performance. For instance, if $I(t_{i+1}) > I(t_i)$, the ASCH-IDS is to increase the proportion of sensory data on M_1 and decrease on M_2 such as: $R_a(t_{i+1}) = R_a(t_i) + \triangle R$ and $R_m(t_{i+1}) = R_m(t_i) - \triangle R$ as formulated in Eqs. (4.9)-(4.10) where $\triangle R$ represents the proportional adjustment of sensor data for each subsystem. An overview of these steps is presented in detail in the flowchart in Fig. 4.2. It is worth noting that the flowchart presents the flow of a continuous decision procedure to adjust the sensory data proportion on each subsystem.

$$R_a(t_{i+1}) = R_a(t_i) \pm \triangle R \tag{4.9}$$

$$R_m(t_{i+1}) = R_m(t_i) \pm \triangle R \tag{4.10}$$

Algorithm 4 Adaptive IDS: (ASCH-IDS)

procedure ROC TRACKING **Input:** $TP_1(t_0)$, $FP_1(t_0)$,$TP_2(t_0)$, $FP_2(t_0)$

Output: $M_1(t_i)$, $M_2(t_i)$ ▷ (1):First subsystem, (2):Second subsystem

Initiate: $M_1(t_0)\leftarrow$INIT , $M_2(t_0)\leftarrow$INIT , $t_i\leftarrow t_0$, $I(t_0)\leftarrow 1$

  for *Allnetwork* **do**

    if HALT **then** END

    else

      Check for $NextAggregation$

    if $NextAggregation$ **then**

$\tau_i \leftarrow t_i + \Delta t$

$M_1(t_i) \leftarrow \alpha M_1(t_i - \Delta t) + (1-\alpha)\frac{TP_1(t_i) - TP_1(t_i - \Delta t)}{FP_1(t_i) - FP_1(t_i - \Delta t)}$

$M_2(t_i) \leftarrow \alpha M_2(t_i - \Delta t) + (1-\alpha)\frac{TP_2(t_i) - TP_2(t_i - \Delta t)}{FP_2(t_i) - FP_2(t_i - \Delta t)}$

    else

      Go to $HALT$

    if $I(t_i)$ **then** $> I(t_i - \Delta t)$

$R_a(t_i) \leftarrow R_a(t_i - \Delta t) + \Delta R$

$R_m(t_i) \leftarrow R_m(t_i - \Delta t) - \Delta R$

Go to Initiate

    if $I(t_i) < I(t_i - \Delta t)$ **then**

$R_a(t_i) \leftarrow R_a(t_i - \Delta t) - \Delta R$

$R_m(t_i) \leftarrow R_m(t_i - \Delta t) + \Delta R$

Go to Initiate

    else

Go to Initiate

4.4 Performance Evaluation

We evaluate the performance of ASCH-IDS to demonstrate the performance improvement of the adaptive IDS solution over its predecessor CHH-IDS in terms of accuracy and detection rates. To this end, we use the Network Simulator version 3 (NS-3) [96] with the simulation settings described in Section 4.4.1. In Section 5.4.3, we present performance results in terms of accuracy, detection rate, ROC, and precision-to-recall characteristics.

4.4.1 Simulation Settings

In the simulation environment, we simulate a WSN of 20 sensors that communicate via the Hierarchical-Dynamic Source Routing (H-DSR) protocol. The sensors are grouped in 4 clusters that spread out in a 100m x 100m area. We repeat each scenario 10 times, and in the figures, we present the average of ten runs with 95% confidence level. Table 4.1 lists a detailed presentation of the simulation settings.

Like its predecessor (CHH-IDS), in ASCH-IDS we have used the Knowledge Discovery in Data mining (KDD) CUP 1999 Data-set in order to validate the efficiency of the proposed ASCH-IDS system on the simulated WSN [112][115] as described in subsection 3.5.1 in Chapter 3.

4.4.2 Numerical Results

- ***Accuracy*** refers to the ratio of the correctly classified occurrences, which are represented by True Positive (TP) and True Negative (TN) as shown in Eq. (4.11) where FN and FP are the False Negative and False Positive cases respectively.

Table 4.1: Simulation settings

Simulation parameter	**Value**
Number of nodes	20
Routing protocol	H-DSR
Number of clusters	4
Simulation time	600s
Packet size	250 bytes
Trust range	[0,1]
Operational area	100m x 100m
Communication range	100m
Attack Types	DoS,Probe,U2R,R2L
$\triangle R$	0.03,0.05,0.1,0.15,0.20 and 0.25
α	0.7
$INIT$	0.5
DoS Attacks	Smurf, land, pod, Neptune, teardrop, back
Probe Attacks	Nmap, portsweep, satan, ipsweep
U2R Attacks	Perl, rootkit, buffer_overflow, loadmodule
R2L Attacks	Imap, guess_passwd, multihop, phf,ftp_write, spy, warezmaster, warezclient

$$AR = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.11}$$

Accuracy has been traced for different scenarios in order to expect the system per-

formance with different data rates as shown in Fig. 4.3. Fig. 4.3 illustrates the accuracy values for the anomaly detection subsystem, misuse detection subsystem, CHH-IDS, and the ASCH-IDS. As seen in the figure, the proposed adaptive methodology results in the highest accuracy of 99.76%. The anomaly decision subsystem achieves better accuracy since the misuse detection subsystem performs with the least accuracy. The best accuracy was achieved by incrementing the data proportion on the anomaly detection subsystem as well as with decrementing the sensory data proportion on misuse detection subsystem with a rate of $\triangle R = 0.25\%$.

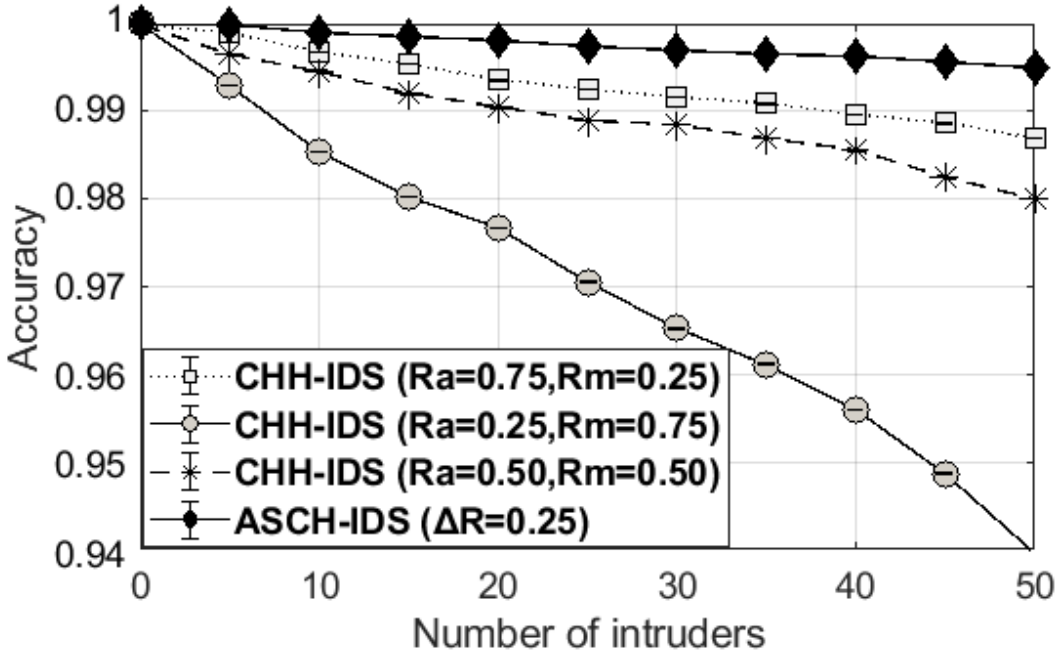

Figure 4.3: Accuracy rates of CHH-IDS under fixed 0.75-0.25, 0.25-0.75 and 0.5-0.5 $R_a - R_m$ distribution for Anomaly and Misuse Detection Subsystems (ADS)(MDS), compared to ASCH-IDS with $\triangle R = 0.25$. By keeping track of the ROC in the anomaly and misuse detection subsystems to adaptively adjust the proportion of sensory data, ASCH-IDS improves the accuracy significantly.

Another accuracy comparison is the one presented in Figure 4.4 which represents

the accuracy of ASCH-IDS comparing to the previously proposed, in Chapter 3, the (RBC-IDS) with different numbers of hidden layers (H). As the figure shows, the proposed RBC-IDS with $H = 3$ results had the highest AR of 99.91%, followed by ASCH-IDS with 99.83%. RBC-IDS with $H = 1$ performed with the least AR.

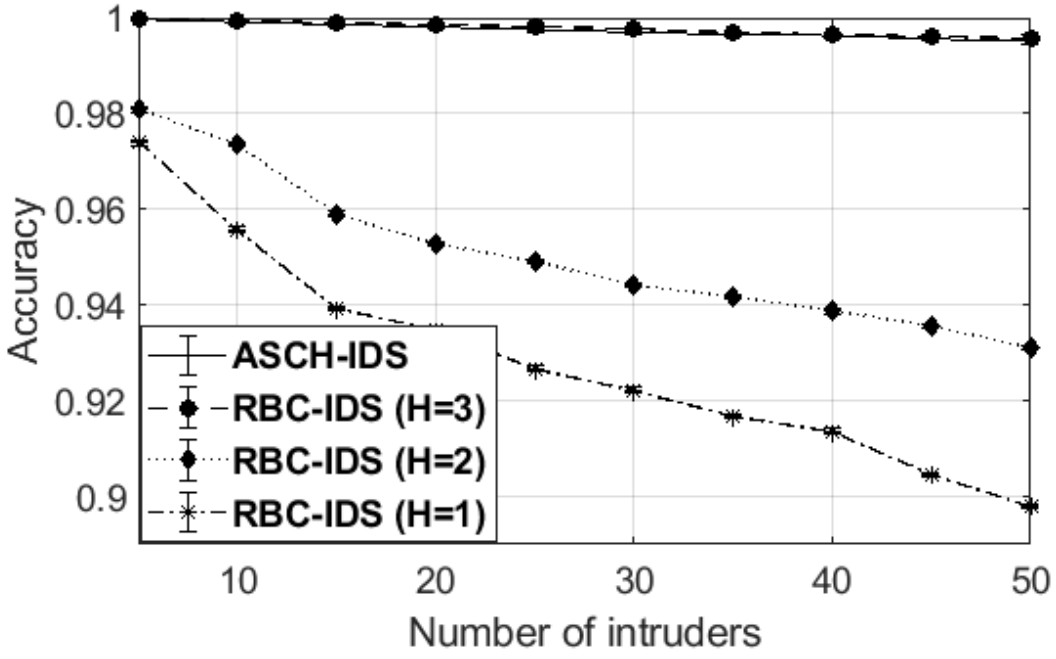

Figure 4.4: Accuracy comparison of RBC-IDS and ASCH-IDS

- ***Detection Rate (DR)*** represents the ratio of sensor behavior that is truly recognized as intrusive. In other words, it represents the True Positive (TP) ratio as shown in Eq. (4.12) where FP refers to False Positive. DR for different data proportions scenarios has been traced in order to expect the system performance with them as shown in Fig. (4.5). Fig. 4.5 illustrates the DRs for the anomaly detection subsystem, misuse detection subsystem, CHH-IDS, and the proposed ASCH-IDS. The proposed ASCH-IDS – as a result of the adaptive decision making by tracking the ROC behavior in each subsystem – leads to the highest DR in detecting the intrusive behavior sensors when compared to each of the individual anomaly detection subsystem and misuse detection subsystem.

$$DR = \frac{TP}{TP + FP} \tag{4.12}$$

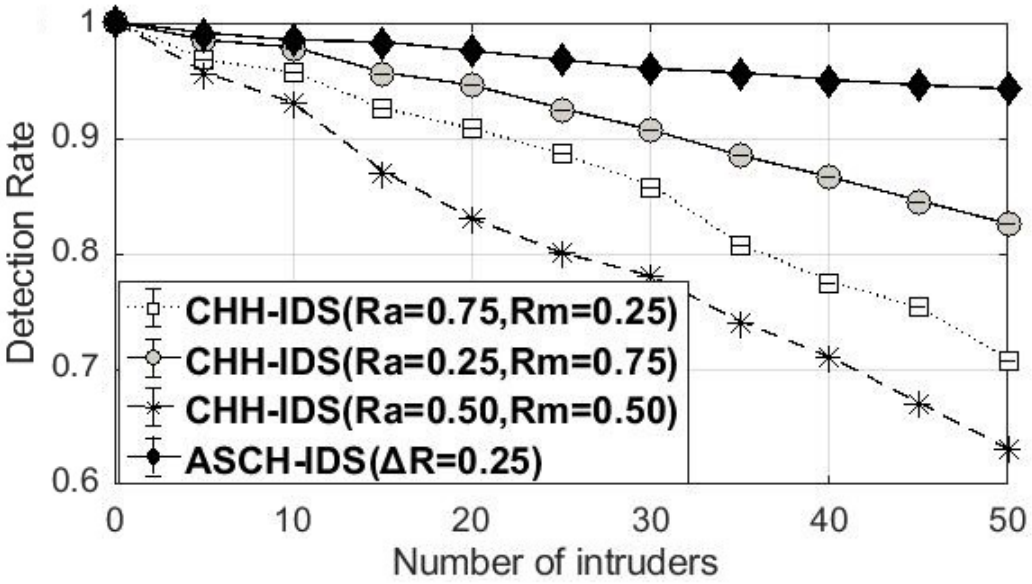

Figure 4.5: Detection rates of CHH-IDS under fixed 0.75-0.25, 0.25-0.75 and 0.5-0.5 $R_a - R_m$ distribution for Anomaly and Misuse Detection Subsystems (ADS)(MDS), compared to ASCH-IDS with $\triangle R = 0.25$. By keeping track of ROC in the anomaly and misuse detection subsystems to adaptively adjust the proportion of sensory data, ASCH-IDS improves the detection rate significantly.

.

Another conclusion that can be made from Fig. 4.5 is that the DR performance decreases by incrementing the data proportion on the anomaly detection subsystem as well as by decrementing the directed data proportion on misuse detection subsystem.

Another detection comparison is the one presented in Figure 4.6 which represents the detection rate of ASCH-IDS compared with the previously proposed, in Chapter 3, RBC-IDS with different hidden layer numbers ($H = 3$, $H = 2$ and $H = 1$).

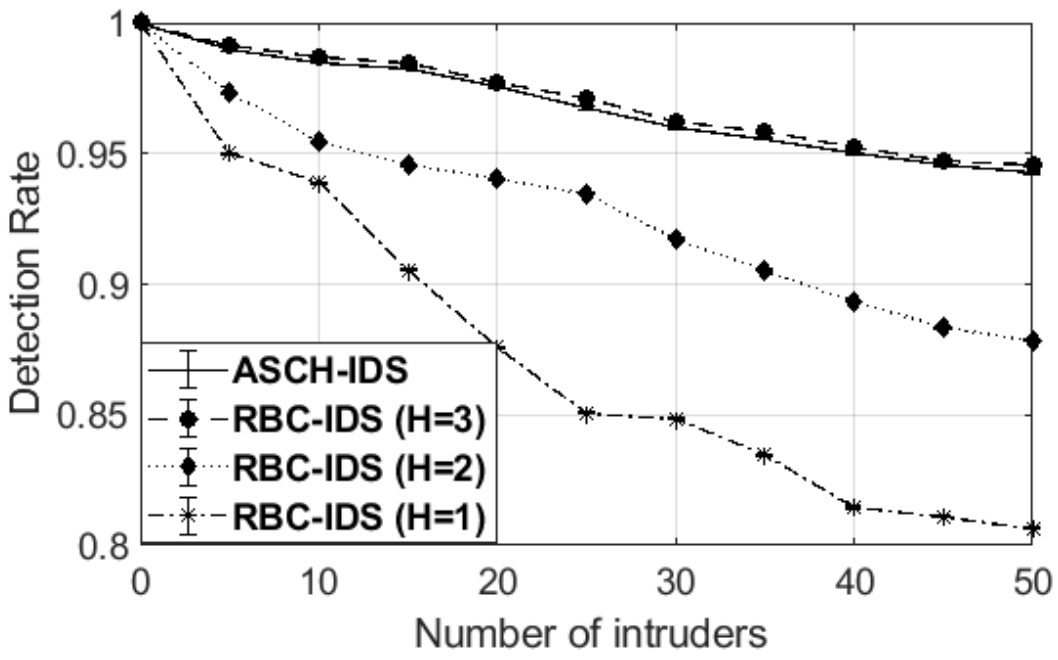

Figure 4.6: Detection comparison of RBC-IDS and ASCH-IDS

It is clear that the RBC-IDS with H=3 achieves the highest detection rate, followed by ASCH-IDS.

- ***Receiver Operating Characteristic (ROC) curve*** characterizes the relationship between TP (Sensitivity) and FP (1-Specificity) for diverse cut-off points. Sensitivity versus specificity adjustment performance denoted by the area under the curve such that better performance is represented by a larger area. ROC curves have been plotted for different scenarios in order to test the system performance with different proportional adjustment of sensor data as shown in Fig. 4.7. According to ROC curves, setting $\triangle R$ at 0.25, the overall performance can be improved effectively.

  Another comparison is the one presented in 4.8 which represents the ROC curves for ASCH-IDS compared to the previously proposed, in Chapter 3, RBC-IDS. As seen in Figure 4.8, the overall RBC-IDS performance can be enhanced when $H = 3$.

- ***Precision-Recall rate*** curve is presented in Fig. 4.9 where recall and precision

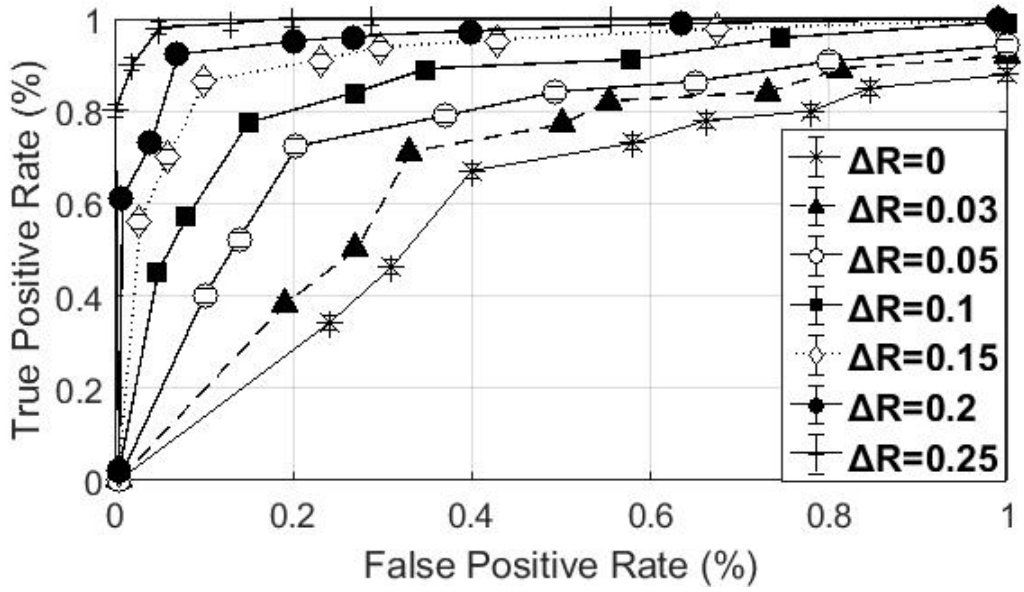

Figure 4.7: ROC curve for different $\triangle R$. Area under the curve is the largest when $\triangle R$ is set to 0.25.

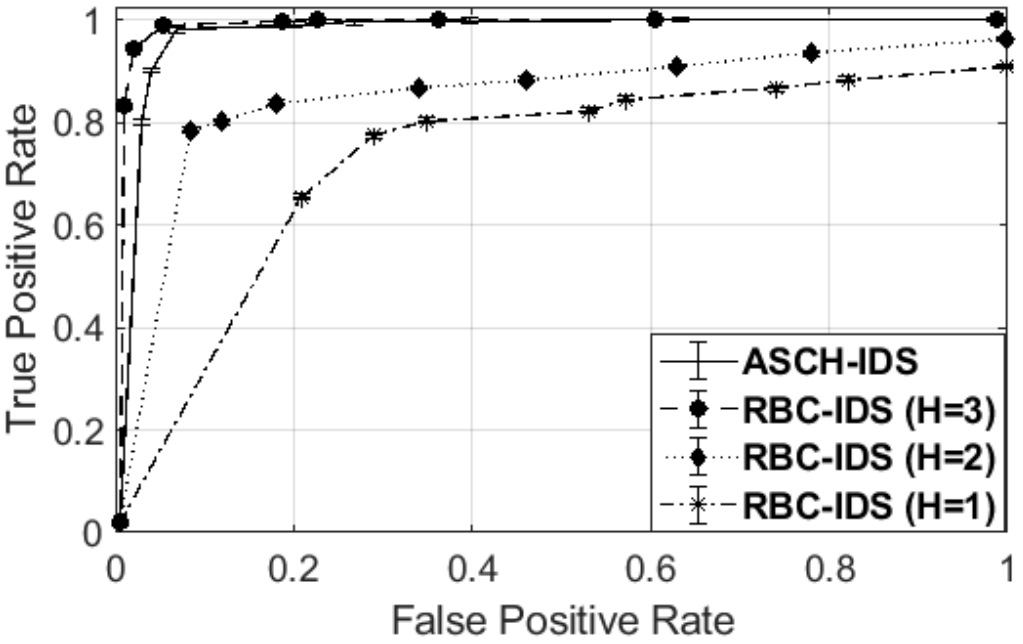

Figure 4.8: ROC comparison of RBC-IDS and ASCH-IDS

are formulated as $TP/(TP + FN)$ and $TP/(TP + FP)$, respectively. High recall and high precision are required to achieve high performance. Therefore the closer the precision-recall rate to one, the better the system performance [125]. Figure 4.9 shows that ASCH-IDS with $\triangle R = 0.25$ achieves a relatively high precision to recall ratio compared to other proportional adjustments of sensor data (i.e. other $\triangle R$ values). Therefore, setting $\triangle R$ at 0.25 achieves highly effective performance with a recall of 99.8% and precision of 90.1%.

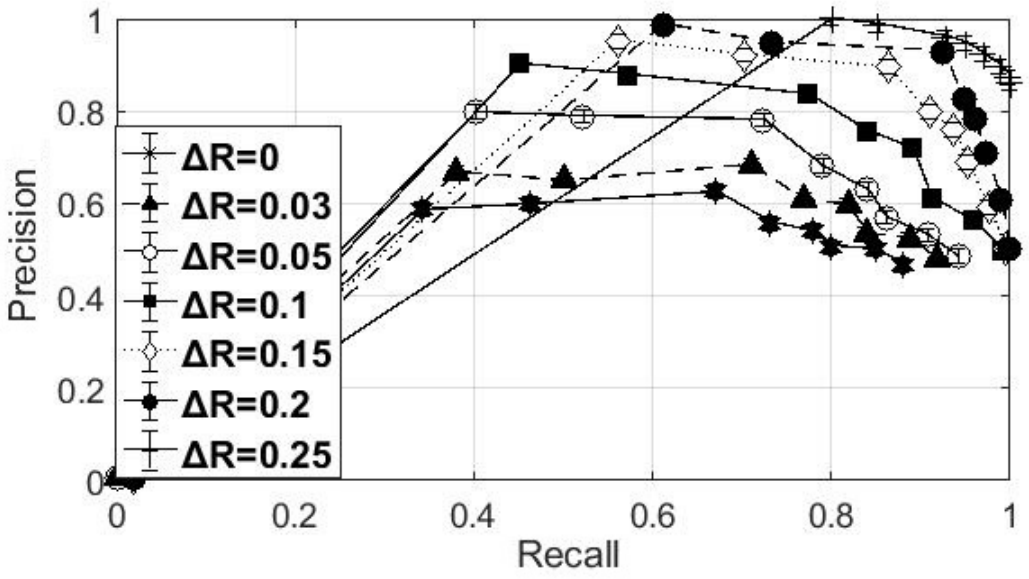

Figure 4.9: Precision - Recall for different $\triangle R$

Figures 4.4, 4.6 and 4.8 show that the adaptive machine learning-based solution (ASCH-IDS) performs at the same rate as the deep learning-based solution (RBC-IDS), whereas adopting a machine learning-based IDS framework (ASCH-IDS) leads to approximately half the detection time of the deep learning-based (RBC-IDS) framework as shown in Table 4.2.

Table 4.2 shows the detection time (training and testing) times of RBC-IDS and ASCH-IDS procedures. It is clear that the detection time (training and testing) for the RBC-IDS procedure is higher than that of the ASCH-IDS procedure.

Table 4.2: Training and testing time comparison

Tested procedure	**Training time (s)**	**Testing time (s)**
RBC-IDS	31.5	1.62
ASCH-IDS	17.1	0.86

4.5 Summary

We have proposed an Adaptive Machine Learning-based Intrusion Detection System namely the Adaptively Supervised and Intrusion-Aware Data Aggregation for Wireless Sensor Clusters in Critical Infrastructures. The proposed scheme is called Adaptively Supervised and Clustered Hybrid - IDS (ASCH-IDS).

ASCH-IDS adopts the previously proposed Clustered Hybrid and Hierarchical IDS (CHH-IDS) which consists of Misuse and Anomaly Detection subsystems (MDS and ADS). ASCH-IDS dynamically adjusts the proportions of sensory data directed to the ADS and MDS based on an indicator that keeps track of the receiver operating characteristic (ROC) behavior in each subsystem. The predecessor, CHH-IDS exhibits a detection rate and accuracy trade-off depending on the sensor data proportion forwarded to one of the two subsystems.

The proposed ASCH-IDS undertook the intrusion problem by using adaptation strategy using different data proportions on ADS and MDS to detect dynamically known and unknown intrusions via supervised and unsupervised machine learning techniques, respectively. Thus, the adjustment on the data proportions leads to adapting the probability of calling supervised (or unsupervised) learning to detect intrusive behaviors. We have evaluated the performance of ASCH-IDS through simulations and demonstrated that the proposed method performs with $\approx$ 99% detection rate and $\approx$ 99.80% accuracy in the presence of known and unknown malicious behavior in the WSN. Furthermore, we have pursued an empirical study on the ROC curve of the ASCH-IDS under various step values to increment/decrement the sensory data on a subsystem. We have shown that setting the incremental/decremental step value aggressively to 25% can ensure the best performance.

Chapter 5

Reinforcement Learning-based IDS for WSNs

5.1 Introduction

Machine learning (ML)-based detection procedures have been used in recognizing numerous attacks and help in preventing intrusions. Nevertheless, most of such procedures fall under the shallow learning concept where they cannot successfully resolve the intrusion data classification problem that faces critical applications. As a promising solution, the authors in [126] propose Latent Dirichlet Allocation in order to recognize and identify the latent semantics of raw data; however, in the context of sensor networks and sensed big data, semantic analysis on message payloads may not be feasible. Nevertheless, a distributed IDS requires an analytics engine which is capable of running a MapReduce cluster [127].

In the analytics engine of a distributed IDS system, ensemble methods –such as in [20]– are hosted to exploit various machine learning algorithms for high detection and accuracy performance. On the other hand, reinforcement learning procedures have the potential

to improve data representations to generate better IDS models. Furthermore, based on the related in [128], it is expected that reinforcement learning can outperform ensemble methods in detection performance in several cases [128].

In this chapter, we present a Reinforcement Learning-based Intrusion Detection System (RL-IDS) for the analytics engine of an IDS of a sensor network that is deployed for monitoring of critical infrastructures.

Reinforcement Learning (RL) considered as an extension to machine learning and involves agents that take actions to maximize the notion of rewards. As an RL solution, we present a Q-Learning based IDS model.

5.1.1 Reinforcement learning (RL)

Reinforcement learning is the learning through the interaction between a decision-making agent and its environment where the agent executes an action by mapping the environmental input to state in discrete time steps (state-to-action mapping). The environment shows the new state and replies with positive or negative feedback namely reward which evaluate the effectiveness of the performed action. The selection mechanism of an action depends on a quantity factor such as the average reward with time with the main objective to maximize numerical rewards. The agent then updates its corresponding policy targeting the optimized future rewards. RL technique may be adopted by the agent who interacts with the dynamic environment as trial-and-error [129]. This agent-environment interaction is shown in Figure 5.1.

5.1.2 Markov Decision Process (MDP)

MDP is a discrete time process that provides a mathematical framework for modelling decision making of an environment under different actions. It enables the prediction of the

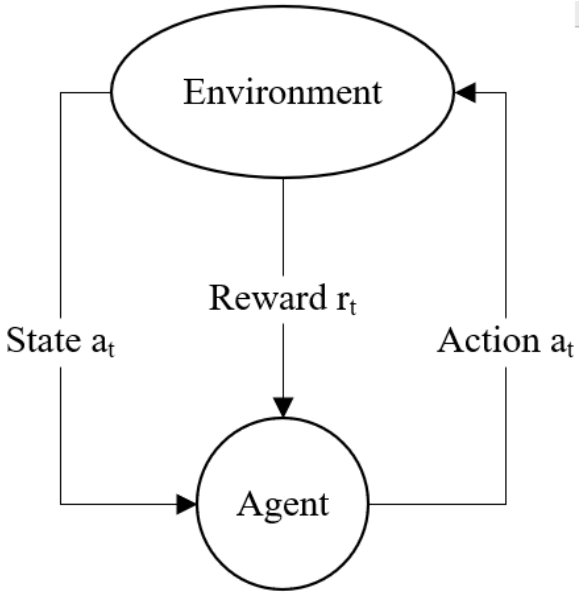

Figure 5.1: The interaction between the Agent and the Environment in Reinforcement Learning (RL)

next state s' and next reward r according to the current state s and action a. A MDP is a 4-tuple (S, A, R_a, P_a) where:

- S is a finite set of states.
- A is a finite set of actions, A_s is the finite set of actions available from state s.
- $(P_a(s, s')$=pr$(s_{t+1} = s'|s_t = s, a_t = a)$ is the transition probability function that action a in state s and time t will lead to state s' at time $t + 1$.
- $R_a(s, s')$ is the immediate reward that is received after transition from state s to state s' due to action a.
- P_a and R_a are the environmental dynamics.

A **finite MDP** is the MDP with a finite set of states and actions where each action consumes the same duration and the agent can observe the environment at all times. Other MDPs such as the Semi-MDPs (SMDPs) where actions have different duration's and Partially Observable MDPs (POMDPs) where actions observe the environment partially

are outside the scope of our works.

In the performance evaluation, as a benchmark to this study, we identify our previously presented -in Chapter 4 - Adaptive Machine Learning-based IDS (AML-IDS) namely an Adaptively Supervised and Clustered Hybrid (ASCH-IDS) methodology [20]. We compare AML-IDS and RL-IDS via simulations and show that accuracy under RL-IDS approach performs with $\approx 100\%$ detection rate and $\approx 100\%$ accuracy rate while AML-IDS performs with a detection rate that is slightly above 99%. Based on our research findings, we report that a RL-based IDS system is preferable over an AML-based IDS system for the analytics engine of an IDS for critical infrastructure monitoring sensor networks.

The remaining of this chapter is organized as follows: First, we start by listing some related works in section 5.2. Second, our proposed Reinforcement Learning-based Intrusion Detection System (RL-IDS) is presented in Section 5.3. In Section 5.4, we present the performance evaluation and finally, we summarize the chapter in Section 5.5.

5.2 Related Work

detection of intrusions is crucial for any networks' reliable operations. Recently, RL has been used for intrusions detection' purposes such as the work presented in [130].

In [130], the authors formulated the online attack detection issue as a Partially Observable Markov Decision Process problem (POMDP) and proposed an RL-based robust online detection algorithm. Their results showed the effectiveness of the proposed RL-based solution for the cyber-attacks', that targeted the studied smart grid network, detection.

A recent work that adopted reinforcement learning for anomaly detection is presented in [131] where the author proposed a platform which is an AI based 6-layer anomaly detection system that combined the behavioral bio-metrics with contextual, social and other signals and covered cases such as user verification, account takeover, remote attacks, and bot attacks.

5.3 Reinforcement Learning-based Clustered IDS Solution (RL-IDS)

The **RL-IDS** solution for WSNs builds on our previously proposed AML-IDS research that utilizes supervised and unsupervised learning modules [20]. As AML-IDS, the RL-IDS use the Weighted Cluster Head Selection (WCHS) algorithm mentioned in Section 3.3.1 where the cluster head (CH) is elected based on the weight's evaluation of each node with the other nodes in the corresponding cluster.

In RL-IDS, as AML-IDS, each CH collects the sensed data from clusters and directs them to the sink by adopting the –previously mentioned in Section 3.3.2– trust based-aggregation method [122]. The method calculates the CH's trust value by tracking the trust of the nodes TR_n with the trust evaluation between the CH and its corresponding nodes $TR^n_{aggregator}$ [122].

In ML, the examined environment is characterized as Markov Decision Process *(MDP)* where RL methods, for example, Q-learning, utilize dynamic programming techniques. *MDP* introduces with the optimum policy idea in order to accomplish the most extreme rewards with time [132].

Fundamentals of RL can be listed as follows:

- The agent cooperates with the environment and takes action A_τ in each state S_τ and waits for the response.
- The environment issues a reward (R_τ) for the accomplished actions, which can have either positive (R^+) or negative value (R^-).
- The agent observes the environment for any changes and optimize the received rewards by updating the policies.

Figure 5.2 represents the proposed RL-IDS architecture where the CHs are responsible for aggregating the sensed data and directing them to the central analytics engine where RL-based Q-learning model is called.

5.3.1 Q-learning method

Q-learning is a model-free reinforcement learning technique which is able to learn without following the current policy [133]. Besides, when Q-learning is applied, starting from the present state, an optimal policy can be found in order to expand the total rewards' expected value.

Q-Learning builds on the concept of state value iteration where an agent aims to estimate the state value function $V(S)$ to update all states $\mathcal{S}$ and actions $\mathcal{A}$ for each iteration in order to know which A results in higher reward R. Q-learning is a simple table-based RL where the rows represent the states whereas the columns represent the actions [134]. In every S, the agent takes an action A, watches the reward for this action R and also the next state $\hat{S}$, and updates the estimated Q ($EST - Q$) value.

In other way, It keeps a Q-value Q(s, a) in a Q table for each state-action. Let a_t and s_t refer to the action and state executed by an agent at time t and r_{t+1} refers to the reinforcement signal that the environment has generated for executing action a_t in state s_t . When the agent accepts the returned reward r_{t+1}, it updates the Q-value that corresponds to s_t and a_t.

Algorithm 5 represents the Q-learning procedure in details [135]. Besides its model-free nature, Q-learning is attractive because when applied, it is able to learn without following the current policy [133].

Equation (5.1) presents the EST-Q value [136] where S_τ represents the state, A_τ stands for the action, and R_τ denotes the reward at time τ. Lastly, Γ and β represent a constant

for the relative value of rewards and the learning rate, respectively, such that $0 < \Gamma < 1$ and $0 < \beta < 1$.

$$\hat{Q}(S_\tau, A_\tau) \leftarrow (1-\beta)Q(S_\tau, A_\tau) + \beta(R_\tau + \Gamma max Q(\hat{S}_\tau, \hat{A}_\tau)) \tag{5.1}$$

Algorithm 5 Q-learning

procedure Q-LEARNING (S$\in \mathcal{S}$,$A \in \mathcal{A}$,$R \in \mathcal{R}$,T,β,Γ)

Input:

States $\mathcal{S} = \{1, 2, ..., n_{\mathcal{S}}\}$

Actions $\mathcal{A} = \{1, 2, ..., n_{\mathcal{A}}\}$

Rewards $\mathcal{R} = \{1, 2, ..., n_{\mathcal{R}}\}$

Learning rate $\beta \in [0, 1]$

Relative value of rewards $\Gamma \in [0, 1]$

Output:

Transition T: $S \times A \rightarrow \hat{S}$

Q $(\hat{S} \times A)$

Initialize Q value : $S \times A \rightarrow R$

    for $S \in \mathcal{S}$ **do**

$\hat{Q}(S, A) \leftarrow (1-\beta)\ Q(S, A) + \beta(R + \Gamma max Q(\hat{S}, \hat{A}))$

$S \leftarrow \hat{S}$

Return Q

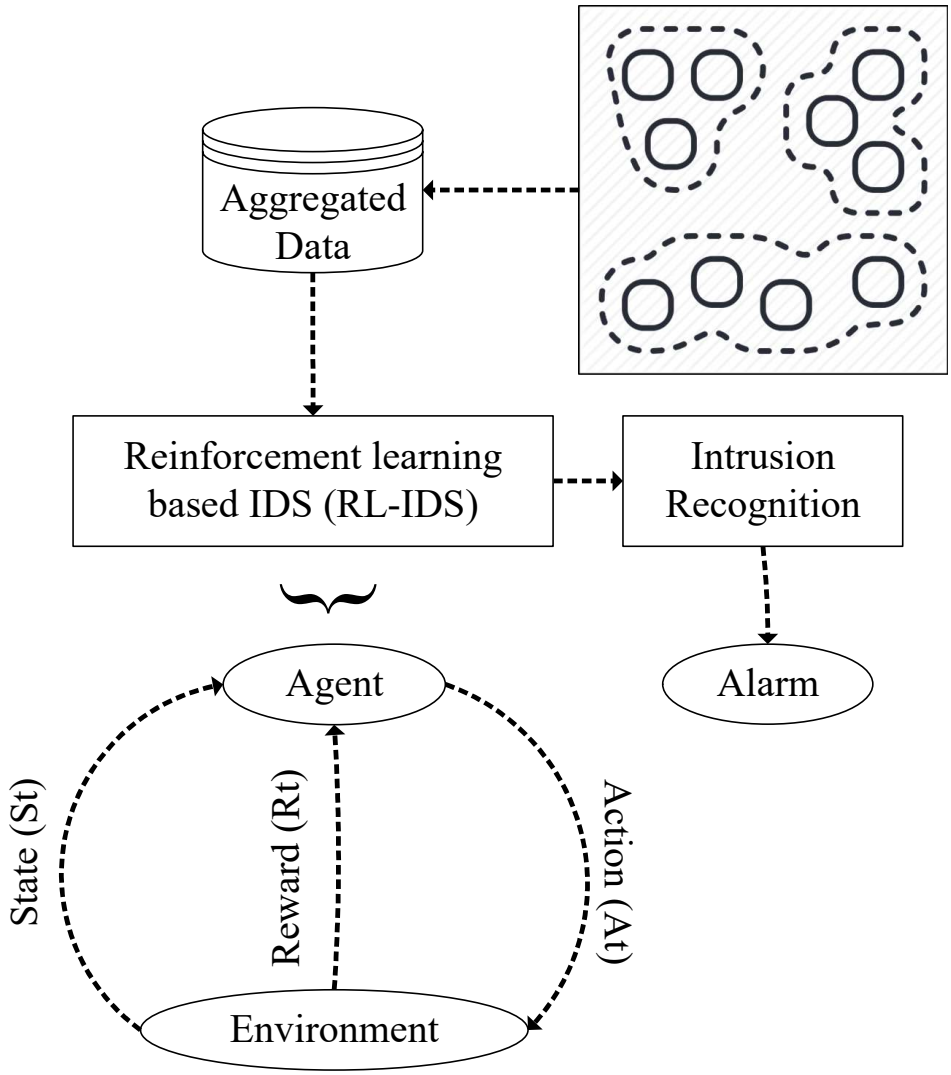

Figure 5.2: The presented RL-IDS framework where the agent interprets the present state (S) of the environment and fulfills an action (A) which eventually will result in some reward (R) to the agent.

5.3.2 Value Iteration

The value iteration procedure is a successive approximation algorithm [137] where the optimal value function is computed by successively expanding the horizon. The policy associated with the value function converges to the optimal policy after a number of iterations

[138], [139].

Equation (5.2) formulates the estimated function $V(S)$ which is the expectation of R that will be achieved in S as an initial state [140]. In the equation $P_{S\hat{S}}(A)$ stands for the transitional probability from state S to state $\hat{S}$ at action A. Additionally, $R(S, \hat{S}, A)$ is the reward gained by transitioning from S to $\hat{S}$ at A, and d refers to the discounted factor weight from future rewards to current rewards [140].

The method used for value iteration is formulated in Eq.(5.3) below. Algorithm 6 describes the value iteration method in details.

The complexity for each update to a single V(S) estimate is O(SxA) since it iterates over all actions to perform the max_A and over all next states for $\sum_{S'}$.

For $V_{I+1}(S)$ where $\forall S \in \mathcal{S} : V_{I+1}(S) \leftarrow max_A \sum_{S'} T$, the overall complexity will be $O(|SxSxA|)$ or $O(|S^2A|)$. The complexity, C of the Algorithm 6 is $O(S^2A)$.

$$V_{I+1}(S) = \sum_{A} \lambda(S, A) \sum_{\hat{S}} P_{S\hat{S}}(R(S, \hat{S}, A) + dV_I(\hat{S})) \tag{5.2}$$

$$V_{I+1}(S) = max_A \sum_{\hat{S}} P_{S\hat{S}}(R(S, \hat{S}, A) + dV_I(\hat{S})) \tag{5.3}$$

Algorithm 6 Value Iteration

procedure V-ITERATION (S$\in \mathcal{S}, A \in \mathcal{A}, R \in \mathcal{R}$,P,$\theta$)

Input:

$\mathcal{S} = \{1, 2, ..., n_{\mathcal{S}}\}$

$\mathcal{A} = \{1, 2, ..., n_{\mathcal{A}}\}$

$\mathcal{R} = \{1, 2, ..., n_{\mathcal{R}}\}$

P $= P_{S\hat{S}}(A)$ ▷ Transition from S to $\hat{S}$ at action A

$\theta \Rightarrow (\theta > 0)$ ▷ θ is a threshold

$I \leftarrow 0$

Output:

$V(S)$ ▷ is the value function

Repeat:

$I \leftarrow I + 1$

For each state S

$V_{I+1}(S) = max_A \sum_{\hat{S}} P_{S\hat{S}}(R(S, \hat{S}, A) + dV_I(\hat{S}))$

until $|V_{I+1}(S) - V_I(S)| < \theta$

Return V_I

EndFor

5.4 Performance Evaluation

The performance of RL-IDS has been evaluated in comparison to the previously presented, in Chapter 4, AML-IDS in terms of Accuracy Rate (AR), Detection Rate (DR), ROC, False Negative Rate (FNR) and F_1 score characteristics. NS-3 [96] has been used with the simulation inputs. The performance results are presented in Section 5.4.3.

5.4.1 Tested environment

A WSN of twenty sensors that communicate through the Dynamic Source Routing protocol for Hierarchical representation networks (H-DSR) had been simulated. The tested sensor nodes are deployed in four clusters in an area of 100m x 100m. Figures present the average of 10 runs for each scenario with a confidence level of 95%.

The rewards generated by the environment are as follows:

1. $R^+ \leftarrow +1$ if :
    - S is normal and A is not sending an alert.
    - S is malicious and A is sending an alert.
2. $R^- \leftarrow -1$ if :
    - S is malicious and A is not sending an alert.
    - S is normal and A is sending an alert.

5.4.2 Tested Dataset

The Knowledge Discovery in Data mining CUP 1999 (KDDCup99) dataset is a subset of the Defense Advanced Research Projects Agency (DARPA) dataset [141]. The KDDCup99

is used to test the presented IDS' efficiency on the simulated WSN, where each connection record contains 41 features and is labeled as normal or attack. The attack types in these datasets are classified into the following pre-defined groups of intrusive behavior: Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe [141].

5.4.3 Numerical results

Accuracy

Accuracy denotes the ratio of the truly classified incidences that return to True Positive (TP) and True Negative (TN) incidences as formulated in Eq. (5.4) [113].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

Accuracy has been presented to trace the AML-IDS performance and compare it with RL-IDS ones. Fig.5.3 presents the accuracy performance for the AML-IDS and the RL-IDS. Where, the proposed RL-IDS results with the highest *accuracy* of ≈ 1, whereas AML-IDS performs at 0.998. RL-IDS, by adopting a Q-learning mechanism, achieves the highest accuracy for the following reason: It depends on the exemplars of the training data sets and also on decision making while the system is running. In addition, in RL-IDS, the agent interacts with the environment to optimize (R^+) by learning the best actions through feedback.

Detection Rate (DR)

DR denotes the behavioral patterns that are accurately recognized as intrusive. It signifies the (TP) ratio as displayed in Eq. (5.5) [113]. DR for RL-IDS compared to AML-IDS as illustrated in Fig. (5.4). The proposed RL-IDS achieves the highest DR followed by

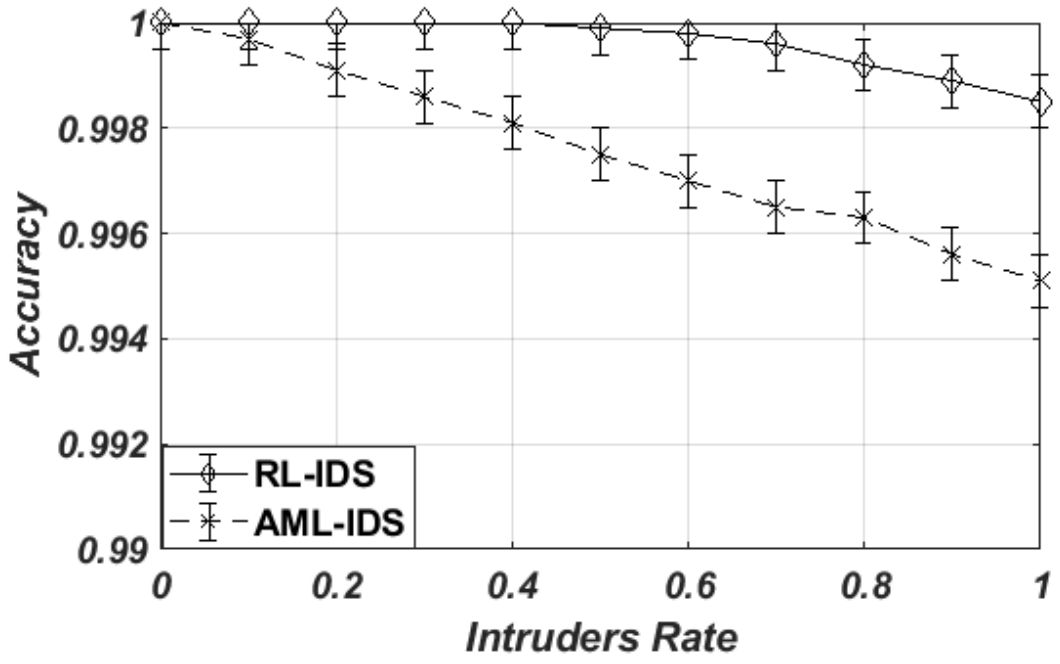

Figure 5.3: Accuracy rates of Reinforcement Learning-IDS (RL-IDS) and Adaptive Machine Learning-IDS (AML-IDS)

AML-IDS as a response to the increase of TP along with a decrease of FP, which would improve DR.

$$DR = \frac{TP}{TP + FP} \tag{5.5}$$

False Negative Rate (FNR)

FNR stands for the ratio of malicious sensor behavior that has been classified as non malicious [19], as formulated in Eq. 5.6 [113]. Thus, FN denotes failure to detect intrusive behaviour. FNR of RL-IDS compared to AML-IDS are shown in Fig. 5.5. In RL-IDS, FNR is lower when compared to that under AML-IDS as a response to the increase of TP which improves both AR and DR.

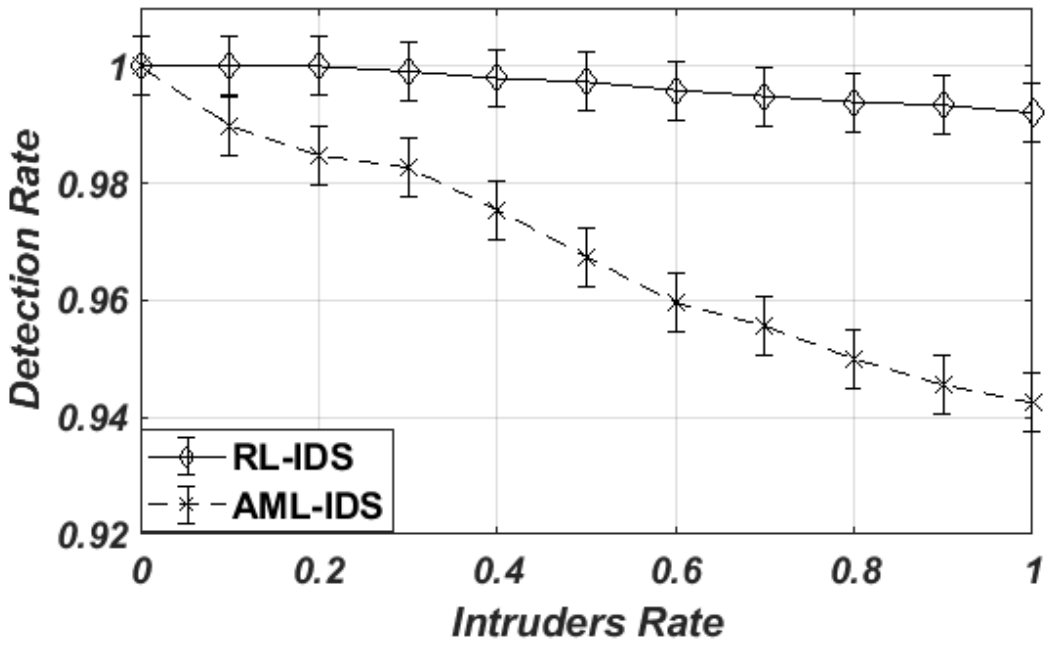

Figure 5.4: Detection rates of Reinforcement Learning-IDS (RL-IDS) compared to detection rate of Adaptive Machine Learning-IDS (AML-IDS)

$$FNR = \frac{FN}{TP + FN + FP + TN} \tag{5.6}$$

(ROC) curve

The Receiver Operating Characteristic (ROC) curve displays the ratio between Sensitivity (TP) and the FP ($1 - Specificity$). Sensitivity-specificity is represented by the area under the curve where the larger area reflects better performance. ROC curves for RL-IDS and AML-IDS are depicted in Fig. 5.6. It is clearly seen that RL-IDS achieves better performance as a result of its higher TP compared to AML-IDS.

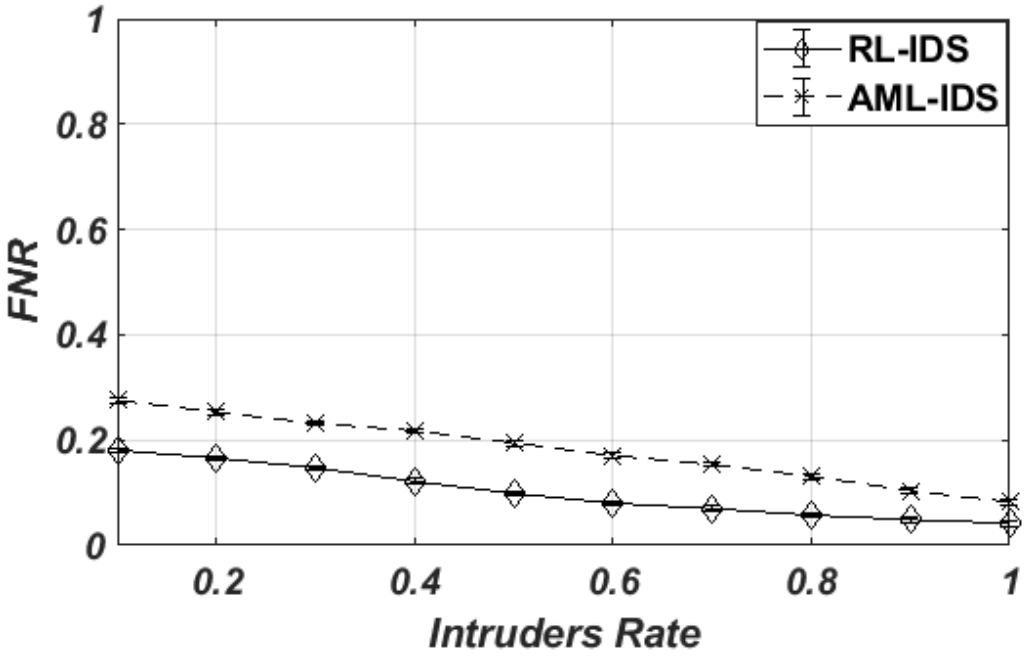

Figure 5.5: FN rates of Reinforcement Learning-IDS (RL-IDS) compared to FNR of Adaptive Machine Learning-IDS (AML-IDS)

F_1 Score curve

F_1 score exploits the precision-recall of the prediction in order to calculate its F score [142]. The **precision** is the number of TP's divided by all positives which is formulated as $TP/(TP + FP)$. **Recall** is formulated as $TP/(TP + FN)$. Figure 5.7 represents the precision-recall rates. The closer precision-recall to 1, the better the performance is achieved [125].

To start with, as shown in Fig. 5.7, RL-IDS outperforms AML-IDS as a result of its high precision and high recall. Precision-recall depends on TP performance which is the reason behind RL-IDS better performance.

F_1 score represents the harmonic mean of precision and recall as shown in equation 5.7 [142]. F_1-score of RL-IDS and AML-IDS are plotted in Fig. 5.8 [142]. Since F_1 score is

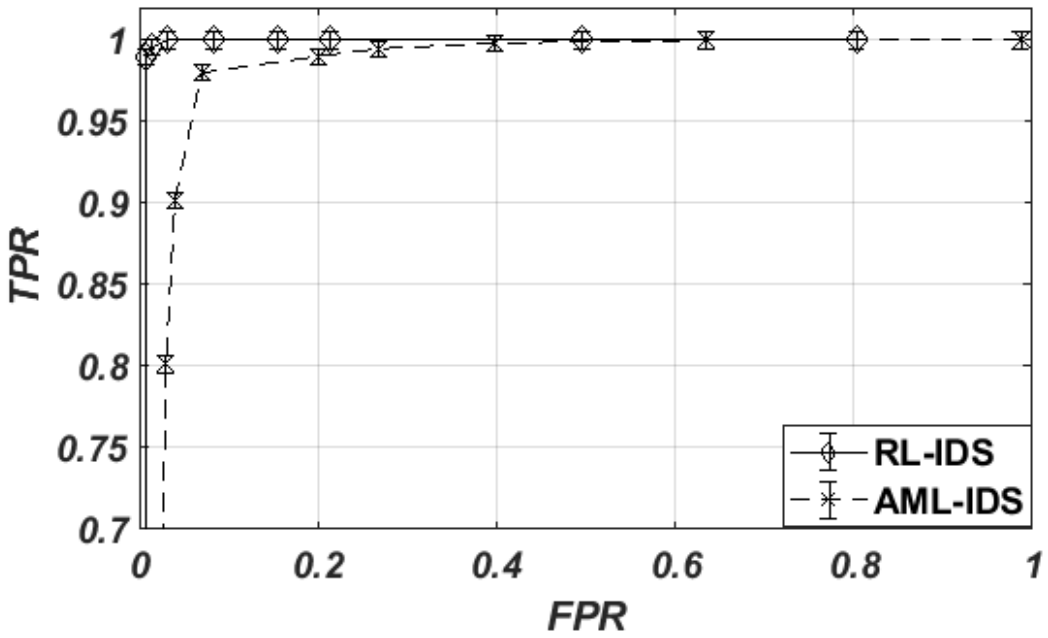

Figure 5.6: ROC of Reinforcement Learning-IDS (RL-IDS) compared to ROC representation of Adaptive Machine Learning-IDS (AML-IDS)

used to assess if the test cases need a balance between precision and recall, the higher the precision-recall ratio the better the F_1 score.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.7}$$

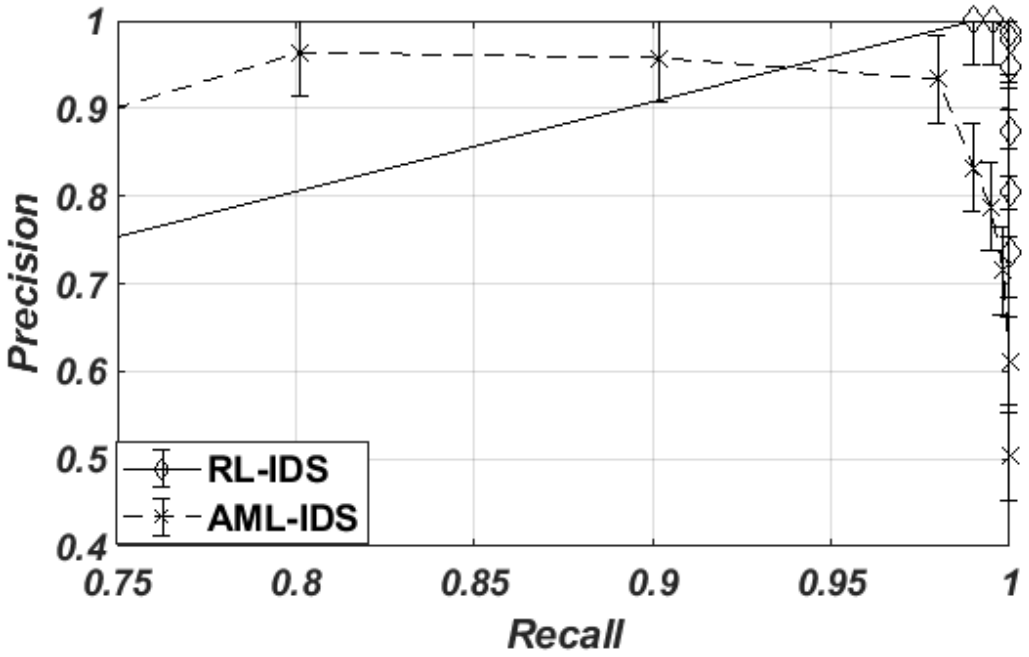

Figure 5.7: Precision-recall representation of Reinforcement Learning-IDS (RL-IDS) compared to precision-recall representation of Adaptive Machine Learning-IDS (AML-IDS)

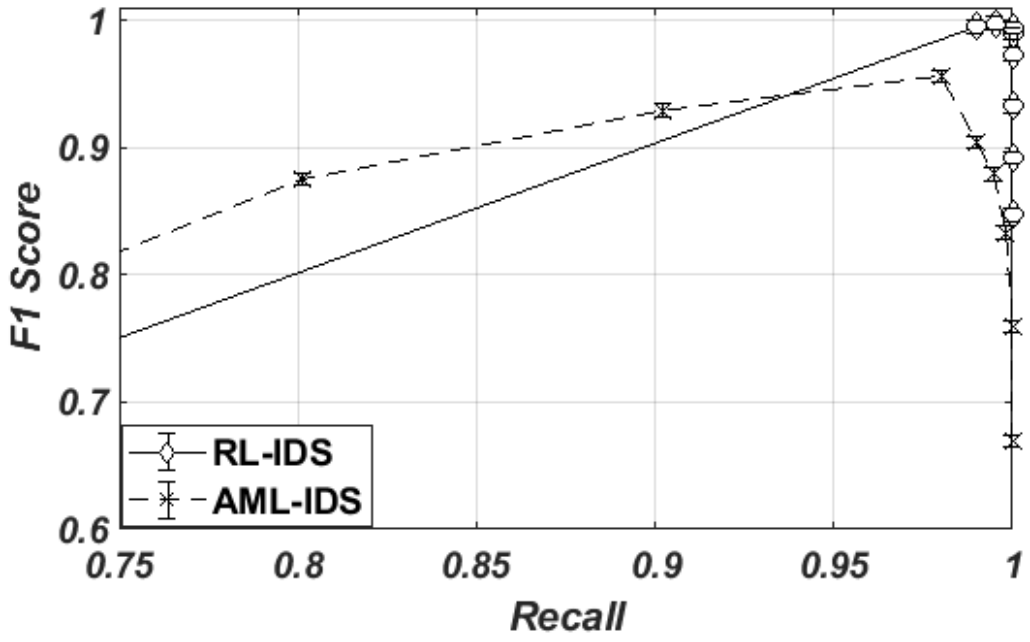

Figure 5.8: F_1 Score representation of Reinforcement Learning-IDS (RL-IDS) compared to F_1 score representation of Adaptive Machine Learning-IDS (AML-IDS)

5.5 Summary

In this chapter, we have proposed an IDS approach by harnessing reinforcement learning techniques on a hybrid IDS framework for Wireless Sensor Networks (WSNs) that monitor critical infrastructures.

We adopted Q-Learning method as a model-free reinforcement learning technique which has been reported to be beneficial.

Through simulations, we have compared the RL-IDS to the previously proposed adaptive machine learning-based IDS namely (AML-IDS) and verified that RL-IDS performs with ≈ 1 detection, accuracy, Precision-recall as well as F_1 score in the existence of intrusive behaviour in the tested WSN. Our performance evaluation reports that the RL-based solution outperforms the AML-based solution with better accuracy, detection, and with the largest area under the curve (ROC).

Chapter 6

Conclusion and Future Work

This chapter summarizes the major contributions of the thesis. It also highlights the road-map for future research directions in the field of sensor network security.

6.1 Concluding Remarks

Critical infrastructure monitoring is considered as one of the vital services in smart cities. Monitoring the networks components and detecting malicious behaviours are fundamental functions to ensure the security of the monitoring operation. Since Wireless Sensor Networks (WSNs) are deployed in open and uncontrolled environments, monitoring through WSNs leads to huge vulnerabilities. Therefore, the robustness of Intrusion Detection Systems (IDSs) in WSNs is a must.

In this thesis, we present a comparative study between Machine Learning-based IDS (ML-IDS), Adaptive Machine Learning-based IDS (AML-IDS), Deep Learning-based IDS (DL-IDS) and Reinforcement Learning-based IDS (RL-IDS) for critical infrastructure monitoring WSNs. We specifically investigate our proposed ML-IDS namely Clustered Hierarchical Hybrid IDS (CHH-IDS), Adaptively Supervised and Clustered Hybrid IDS (ASCH-IDS) with a Restricted Boltzmann Machine-based and Clustered IDS (RBC-IDS), and Q-Learning-based IDS (QL-IDS), which is a reinforcement learning solution. The purpose of this research is to test and determine the impact of using different learning techniques on WSNs' security metrics (Accuracy Rate (AR), Detection Rate (DR), False Negative Rate (FNR), Receiver Operating Characteristics (ROC) curves and F_1 score).

Our aim is to increase the AR, DR and decrease the generated FNR and compare the results of different learning techniques.

Proposed IDSs implementation is accomplished using the Network Simulator-3 (NS3) simulator with the adaption of KDD'99 dataset.

Through simulations, we have verified that Reinforcement Learning-based IDS (RL-IDS), namely QL-IDS, works with $\approx$ 100% detection rate and $\approx$ 100% accuracy rate in the existence of intrusive behaviour in the tested WSN.

From the performance analysis, we have also shown that the AML-based solution performs

in same rates as the DL-based solution whereas adopting an ML-based IDS framework leads to approximately half the detection time of the DL-based RBM-IDS framework.

We have also shown that the RL-based solution performs with the best precision-recall, F_1 score with ≈ 1 which represents the best performance and with the largest area under the curve (ROC).

6.2 Future Research Directions

Based on the investigations and conclusions in this thesis, there are several areas that could be considered for future research. These are summarized as follows:

1. Apply the presented IDSs on different networks.
2. Evaluate the presented IDSs with different input traffic (Datasets) such as DARPA and NSL-KDD.
3. Study the delay metric (time matter) in order to reduce the learning' training and testing times.
4. Apply our proposed intrusion detection method on high speed big data environments.

References

[1] L. Dhanabal and Dr. S. P. Shantharajah. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. 2015.

[2] A. Gouveia and M. Correia. *A Systematic Approach for the Application of Restricted Boltzmann Machines in Network Intrusion Detection*, volume 10305. 05 2017.

[3] S. Seo, S. Park, and J. Kim. Improvement of network intrusion detection accuracy by using restricted boltzmann machine. In *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 413–417, Dec 2016.

[4] S. R. J. Ramson and D. J. Moni. Applications of wireless sensor networks a survey. In *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, pages 325–329, Feb 2017.

[5] S. Otoum, M. Ahmed, and H. T. Mouftah. Sensor medium access control (smac)-based epilepsy patients monitoring system. In *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1109–1114, May 2015.

[6] R. Kaur and P. Singh. Review of black hole and grey hole attack. 6:35–45, 12 2014.

[7] X. Zhang and J. Wang. An efficient key management scheme in hierarchical wireless sensor networks. In *2015 International Conference on Computing, Communication and Security (ICCCS)*, pages 1–7, Dec 2015.

[8] E. Hernndez-Pereira, J.A. Surez-Romero, O. Fontenla-Romero, and A. Alonso-Betanzos. Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. *Expert Systems with Applications*, 36(7):10612 – 10617, 2009.

[9] G. Kumar, K. Kumar, and M. Sachdeva. The use of artificial intelligence based techniques for intrusion detection: a review. *Artificial Intelligence Review*, 34(4):369–387, Dec 2010.

[10] M. Asaka, S. Okazawa, A. Taguchi, and S. GOTO. A method of tracing intruders by use of mobile agents, 1999.

[11] Ch. Kruegel, G. Vigna, and W. Robertson. A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48:717–738, 08 2005.

[12] I. Butun, S. D. Morgera, and R. Sankar. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys Tutorials*, 16(1):266–282, First 2014.

[13] L. Buttyan, D. Gessner, A. Hessler, and P. Langendoerfer. Application of wireless sensor networks in critical infrastructure protection: challenges and design options [security and privacy in emerging wireless networks]. *IEEE Wireless Communications*, 17(5):44–49, October 2010.

[14] X. Zhang, J. Du, C. Fan, D. Liu, J. Fang, and L. Wang. A wireless sensor monitoring node based on automatic tracking solar-powered panel for paddy field environment. *IEEE Internet of Things Journal*, 4(5):1304–1311, Oct 2017.

[15] E. N. Ganesh. Iot based environment monitoring using wireless sensor network. 5:964–970, 02 2017.

[16] J. Sriwan and W. Suntiamorntut. Human activity monitoring system based on wsns. In *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 247–250, July 2015.

[17] D. Rojas and J. Barrett. A hardware-software wsn platform for machine and structural monitoring. In *2017 28th Irish Signals and Systems Conference (ISSC)*, pages 1–6, June 2017.

[18] P. Guo, J. Cao, and X. Liu. Lossless in-network processing in wsns for domain-specific monitoring applications. *IEEE Transactions on Industrial Informatics*, 13(5):2130–2139, Oct 2017.

[19] S. Otoum, B. Kantarci, and H. T. Mouftah. Detection of known and unknown intrusive sensor behavior in critical applications. *IEEE Sensors Letters*, 1(5):1–4, Oct 2017.

[20] S. Otoum, B. Kantraci, and H. T. Mouftah. Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures. In *IEEE International Conference on Communications (ICC)*, pages 1–6, May 2018.

[21] S. Otoum, B. Kantraci, and H. T. Mouftah. Hierarchical trust-based black-hole detection in wsn-based smart grid monitoring. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[22] S. Otoum, B. Kantraci, and H. T. Mouftah. Mitigating false negative intruder decisions in wsn-based smart grid monitoring. In *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*.

[23] S. Otoum, B. Kantarci, and H. T. Mouftah. On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, pages 1–1, 2019.

[24] V. Kumar and R. Kumar. An adaptive approach for detection of blackhole attack in mobile ad hoc network. *Procedia Computer Science*, 48:472 – 479, 2015. International Conference on Computer, Communication and Convergence (ICCC 2015).

[25] M. Pouryazdan, B. Kantarci, T. Soyata, and H. Song. Anchor-assisted and vote-based trustworthiness assurance in smart city crowdsensing. *IEEE Access*, 4:529–541, 2016.

[26] J. Sen. Detection of cooperative black hole attack in wireless ad hoc networks. 12:26 – 33, 02 2012.

[27] J. Yin and S. K. Madria. A hierarchical secure routing protocol against black hole attacks in sensor networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, volume 1, pages 8 pp.–, June 2006.

[28] G. Pathak, S. Patil, and J. Tryambake. Efficient and trust based black hole attack detection and prevention in wsn. In *International journal of computer science and business informatics*, volume 14, pages 93–103, September 2014.

[29] F. Belabed and R. Bouallegue. An optimized weight-based clustering algorithm in wireless sensor networks. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 757–762, Sept 2016.

[30] H. Sedjelmaci, S. M. Senouci, and M. Abu-Rgheff. An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks. In *IEEE Internet of Things Journal*, volume 1, pages 570–577, 2014.

[31] Y. K. Penya, I. Ruiz-Agndez, and P. G. Bringas. Integral misuse and anomaly detection and prevention system. In Pawel Skrobanek, editor, *Intrusion Detection Systems*, chapter 9. IntechOpen, Rijeka, 2011.

[32] E. Cole, R. L. Krutz, and J. Conley. *Network Security Bible*. John Wiley #38; Sons, Inc., USA, 2005.

[33] M. Sourour, B. Adel, and A. Tarek. Environmental awareness intrusion detection and prevention system toward reducing false positives and false negatives. In *2009 IEEE Symposium on Computational Intelligence in Cyber Security*, pages 107–114, March 2009.

[34] Sh. Wu and W. Banzhaf. The use of computational intelligence in intrusion detection systems: A review. In *Applied Soft Computing 10 (2010)*, volume 10, page 135, 2010.

[35] H. T. Elshoush and I. M. Osman. Reducing false positives through fuzzy alert correlation in collaborative intelligent intrusion detection systems a review. In *International Conference on Fuzzy Systems*, pages 1–8, July 2010.

[36] G. P. Spathoulas and S. K. Katsikas. Using a fuzzy inference system to reduce false positives in intrusion detection. In *2009 16th International Conference on Systems, Signals and Image Processing*, pages 1–4, June 2009.

[37] I. . Chen, P. . Lin, C. . Luo, T. . Cheng, Y. . Lin, Y. . Lai, and F. C. Lin. Extracting attack sessions from real traffic with intrusion prevention systems. In *2009 IEEE International Conference on Communications*, pages 1–5, June 2009.

[38] Y. Lin, P. Lin, S. Wang, I. Chen, and Y. Lai. Pcaplib: A system of extracting, classifying, and anonymizing real packet traces. *IEEE Systems Journal*, 10(2):520–531, June 2016.

[39] M. Sharma, K. Jindal, and A. Kumar. Intrusion detection system using bayesian approach for wireless network. *International Journal of Computer Applications*, 48(5):2933, 2012.

[40] R. Elbasiony, S. Elsayed, T. Eltobely, and M. Fahmy. A hybrid network intrusion detection framework based on random forests and weighted k-means. In *Ain Shams Engineering Journal*, volume 4, page 753762, 2013.

[41] X. Sun, B. Yan, X. Zhang, and Ch. Rong. An integrated intrusion detection model of cluster-based wireless sensor network. *PLOS ONE*, 10(10):1–16, 10 2015.

[42] F. Bao, I. Chen, M. Chang, and J. Cho. Trust-based intrusion detection in wireless sensor networks. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–6, June 2011.

[43] S. Mutlu and G. Yilmaz. A distributed cooperative trust based intrusion detection framework for manets. In *The Seventh International Conference on Networking and Services (ICNS 2011)*, 2011.

[44] K. Mukesh and A. K. Sheik. Trust based intrusion detection system for mobile ad hoc network. In *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, volume 1, pages 106–109, 2013.

[45] P. Dharmesh and P. Yask. Intrusion detection systems for trust based routing in ad-hoc networks. In *International Journal of Computer Science and Information Technology and Security (IJCSITS)*, volume 2, 2012.

[46] N. F. Haq, A. R. Onik, and F. M. Shah. An ensemble framework of anomaly detection using hybridized feature selection approach (hfsa). In *2015 SAI Intelligent Systems Conference (IntelliSys)*, pages 989–995, Nov 2015.

[47] A. Hadri, K. Chougdali, and R. Touahni. Intrusion detection system using pca and fuzzy pca techniques. In *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, pages 1–7, Oct 2016.

[48] B. Subba, S. Biswas, and S. Karmakar. Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, Nov 2016.

[49] A. R. Syarif and W. Gata. Intrusion detection system using hybrid binary pso and k-nearest neighborhood algorithm. In *2017 11th International Conference on Information Communication Technology and System (ICTS)*, pages 181–186, Oct 2017.

[50] P. Nskh, M. N. Varma, and R. R. Naik. Principle component analysis based intrusion detection system using support vector machine. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 1344–1350, May 2016.

[51] S. Zhao, W. Li, T. Zia, and A. Y. Zomaya. A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things. In

2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), pages 836–843, Nov 2017.

[52] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli. A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access*, 6:20255–20261, 2018.

[53] J. Straub. Testing automation for an intrusion detection system. In *2017 IEEE AUTOTESTCON*, pages 1–6, Sept 2017.

[54] A. Honig, A. Howard, E. Eskin, and S. Stolfo. Adaptive model generation: An architecture for deployment of data mining-based intrusion detection systems. In *IN*, pages 153–194. Kluwer Academic Publishers, 2002.

[55] H. Sedjelmaci, S. M. Senouci, and T. Taleb. An accurate security game for low-resource iot devices. *IEEE Transactions on Vehicular Technology*, 66(10):9381–9393, Oct 2017.

[56] F. Jemili, M. Zaghdoud, and M. B. Ahmed. A framework for an adaptive intrusion detection system using bayesian network. In *IEEE Intelligence and Security Informatics*, pages 66–70, May 2007.

[57] P. A. A. Resende and A. C. Drummond. Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling. *Security and Privacy*, 1(4):e36.

[58] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri. Real-time multi-agent system for an adaptive intrusion detection system. *Pattern Recognition Letters*, 85:56 – 64, 2017.

[59] D. Krishnan. A distributed self-adaptive intrusion detection system for mobile ad-hoc networks using tamper evident mobile agents. *Procedia Computer Science*, 46:1203 –

1208, 2015. Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India.

[60] M. Hossain and S. M. Bridges. A framework for an adaptive intrusion detection system with data mining. 2001.

[61] A. Nadeem and M. Howarth. Adaptive intrusion detection and prevention of denial of service attacks in manets. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, IWCMC '09, pages 926–930, New York, NY, USA, 2009. ACM.

[62] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien. Hybrid intelligent intrusion detection scheme. In António Gaspar-Cunha, Ricardo Takahashi, Gerald Schaefer, and Lino Costa, editors, *Soft Computing in Industrial Applications*, pages 293–303, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[63] N. D. Lane and P. Georgiev. Can deep learning revolutionize mobile sensing? In *Intl. Workshop on Mobile Computing Systems and Applications*, HotMobile '15, pages 117–122. ACM, 2015.

[64] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova. Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security and Applications*, 41:1 – 11, 2018.

[65] M. Z. Alom, V. Bontupalli, and T. M. Taha. Intrusion detection using deep belief networks. In *2015 National Aerospace and Electronics Conference (NAECON)*, pages 339–344, June 2015.

[66] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, Feb 2018.

[67] Ch. Yin, Y. Zhu, J. Fei, and X. He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[68] A. Y. Javaid, Q. Niyaz, W. Sun, and M. Alam. A deep learning approach for network intrusion detection system. *ICST Trans. Security Safety*, 3:e2, 2015.

[69] TA. Tang, L. Mhamdi, D. McLernon, S. Zaidi, and M. Ghogho. Deep learning approach for network intrusion detection in software defined networking, December 2016.

[70] U. Fiore, F. Palmieri, A. Castiglione, and A. D. Santis. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, 122:13 – 23, 2013.

[71] Y. Li, R. Ma, and R. Jiao. A hybrid malicious code detection method based on deep learning. 9:205–216, 05 2015.

[72] K. Alrawashdeh and C. Purdy. Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 195–200, Dec 2016.

[73] M. Haque and T. M. Alkharobi. Adaptive hybrid model for network intrusion detection and comparison among machine learning algorithms. 5, 02 2015.

[74] M. Sheikhan, Z. Jadidi, and A. Farrokhi. Intrusion detection using reduced-size rnn based on feature grouping. *Neural Computing and Applications*, 21:1185–1190, 2010.

[75] E. Hodo, X. Bellekens, A. Hamilton, P. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson. Threat analysis of iot networks using artificial neural network intrusion detection system. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6, May 2016.

[76] C. Yin, Y. Zhu, J. Fei, and X. He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[77] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, Feb 2018.

[78] A. Servin and D. Kudenko. Multi-agent reinforcement learning for intrusion detection. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pages 211–223, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[79] X. Xu and T. Xie. A reinforcement learning approach for host-based intrusion detection using sequences of system calls. In *Advances in Intelligent Computing*, pages 995–1003, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[80] I. Tiyas, A. Barakbah, T. Harsono, and A. Sudarsono. Reinforced intrusion detection using pursuit reinforcement competitive learning. *EMITTER International Journal of Engineering Technology*, 2(1):39–49, 2014.

[81] J. C. G. Next generation intrusion detection: Autonomous reinforcement learning of network attacks. In *In Proceedings of the 23rd National Information Systems Secuity Conference*, pages 1–12, 2000.

[82] A. Servin. Towards traffic anomaly detection via reinforcement learning and data flow. pages 81–88.

[83] N. Sengupta and J. Sil. Comparison of supervised learning and reinforcement learning in intrusion domain. In K. R. Venugopal and L. M. Patnaik, editors, *Wireless Networks and Computational Intelligence*, pages 546–551, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[84] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, and M. C. Govil. A comparative analysis of svm and its stacking with other classification algorithm for intrusion detection. In *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)*, pages 1–6, April 2016.

[85] F. Karata and S. A. Korkmaz. Big data: Controlling fraud by using machine learning libraries on spark. *International Journal of Applied Mathematics, Electronics and Computers*, 6(1):1–5, 2018.

[86] K. Peng, V. C. M. Leung, and Q. Huang. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, 6:11897–11906, 2018.

[87] K. Peng, V. C. M. Leung, L. Zheng, Sh. Wang, Ch. Huang, and T. Lin. Intrusion detection system based on decision tree over big data in fog environment. *Wirel. Commun. Mob. Comput.*, 2018:13–, March 2018.

[88] M. Belouch, S. El Hadaj, and M. Idhammad. Performance evaluation of intrusion detection based on machine learning using apache spark. *Procedia Comput. Sci.*, 127(C):1–6, May 2018.

[89] M. A. Manzoor and Y. Morgan. Real-time support vector machine based network intrusion detection system using apache storm. pages 1–5, 10 2016.

[90] J. Schmidhuber. Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828, 2014.

[91] L. Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3:e2, 2014.

[92] Z. Wang. Deep learning-based intrusion detection with adversaries. *IEEE Access*, 6:38367–38384, 2018.

[93] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017.

[94] F. Bao, I. Chen, M. Chang, and J. Cho. Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. *IEEE Transactions on Network and Service Management*, 9(2):169–183, June 2012.

[95] Ana Paula R. da Silva, Marcelo H. T. Martins, Bruno P. S. Rocha, Antonio A. F. Loureiro, Linnyer B. Ruiz, and Hao Chi Wong. Decentralized intrusion detection in wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Quality of Service &Amp; Security in Wireless and Mobile Networks*, Q2SWinet '05, pages 16–23, New York, NY, USA, 2005. ACM.

[96] *ns-3 Tutorial*. Available at https://www.nsnam.org/docs/tutorial/html/, Last Visit: April.12.2018.

[97] T. T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys Tutorials*, 10(4):56–76, Fourth 2008.

[98] F. Belabed and R. Bouallegue. An optimized weight-based clustering algorithm in wireless sensor networks. *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016.

[99] W. Zhang, S. Das, and Y. Liu. A trust based framework for secure data aggregation in wireless sensor networks. *IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2006.

[100] D. Ma and A. Zhang. An adaptive density-based clustering algorithm for spatial database with noise. *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 467–470, 2004.

[101] M. F. Jaing, S. S. Tseng, and C. M. Su. Two-phase clustering process for outliers detection. *Pattern Recogn. Lett.*, 22(6-7):691–700, May 2001.

[102] A. Ram, A. Sharma, A. S. Jalal, A. Agrawal, and R. Singh. An enhanced density based spatial clustering of applications with noise. In *IEEE Intl. Advance Computing Conference*, pages 1475–1478, March 2009.

[103] J. Zhang, M. Zulkernine, and A. Haque. Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38:649–659, 2008.

[104] Random forests, leo breiman and adele cutler.

[105] L. Deng and D. Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3–4):197–387, June 2014.

[106] *Are we there yet?* Available at http://rodrigob.github.io/are_we_there_yet/build/, Last Visit: Dec.30.2018.

[107] G. E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[108] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[109] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 153–160, Cambridge, MA, USA, 2006. MIT Press.

[110] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[111] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[112] *KDD Cup 1999 Data.* Available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Last Visit: December.17.2018.

[113] B. M. Beigh and M. A. Peer. Performance evaluation of different intrusion detection system: An empirical approach. In *Intl. Conf. on Computer Communication and Informatics*, pages 1–7, Jan 2014.

[114] S. Otoum, B. Kantarci, and H. T. Mouftah. Hierarchical trust-based black-hole detection in wsn-based smart grid monitoring. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[115] S.Otoum, Burak Kantarci, and Hussein T. Mouftah. Mitigating false negative intruder decisions in wsn-based smart grid monitoring. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 153–158, June 2017.

[116] A. M. Chandrasekhar and K. Raghuveer. Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers. In *2013 International Conference on Computer Communication and Informatics*, pages 1–7, Jan 2013.

[117] A. George. Article: Anomaly detection based on machine learning: Dimensionality reduction using pca and classification using svm. *International Journal of Computer Applications*, 47(21):5–8, June 2012.

[118] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer-Verlag New York, 2 edition, 2009.

[119] E. Anthi, L. Williams, and P. Burnap. Pulse: An adaptive intrusion detection for the internet of things. 05 2018.

[120] H. Qu, Z. Qiu, X. Tang, M. Xiang, and P. Wang. An adaptive intrusion detection method for wireless sensor networks. *International Journal of Advanced Computer Science and Applications*, 8, 01 2017.

[121] P. Techateerawat and A. Jennings. Adaptive intrusion detection in wireless sensor networks. In *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, pages 23–28, Oct 2007.

[122] W. Zhang, S. Das, and Y. Liu. A trust based framework for secure data aggregation in wireless sensor networks. *IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2006.

[123] D. Ma and A. Zhang. An adaptive density-based clustering algorithm for spatial database with noise. *IEEE Intl Conf on Data Mining (ICDM'04)*.

[124] J. Zhang, M. Zulkernine, and A. Haque. Random-forests-based network intrusion detection systems. *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, 38/5:649659, 2008.

[125] M. Elhamahmy, H. Elmahdy, and I. A. Saroit. A new approach for evaluating intrusion detection system. In *Artificial Intelligent Systems and Machine Learning*, volume 2, pages 290–298, November 2010.

[126] J. Huang, Z. Kalbarczyk, and D. M. Nicol. Knowledge discovery from big data for intrusion detection using lda. In *IEEE International Congress on Big Data*, pages 760–761, June 2014.

[127] V. P. Janeja, A. Azari, J. M. Namayanja, and B. Heilig. B-dids: Mining anomalies in a big-distributed intrusion detection system. In *IEEE International Conference on Big Data (Big Data)*, pages 32–34, Oct 2014.

[128] A. Servin and D. Kudenko. Multi-agent reinforcement learning for intrusion detection. pages 211–223, 02 2008.

[129] L. P. Kaelbling, M. Littman, and A. P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 04 1996.

[130] M. N. Kurt, O. E. Ogundijo, Ch. Li, and X. Wang. Online cyber-attack detection in smart grid: A reinforcement learning approach. *CoRR*, abs/1809.05258, 2018.

[131] H. Koduvely. Anomaly detection through reinforcement learning, 01 2018.

[132] S. Doltsinis, P. Ferreira, and N. Lohse. An mdp model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(9):1125–1138, Sept 2014.

[133] Ch. Gaskett, D. Wettergreen, and A. Zelinsky. Q-learning in continuous state and action spaces. In Norman Foo, editor, *Advanced Topics in Artificial Intelligence*, pages 417–428, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[134] M. Pouyan, A. Mousavi, S. Golzari, and A. Hatam. Improving the performance of q-learning using simultanouse q-values updating. In *2014 International Congress on Technology, Communication and Knowledge (ICTCK)*, pages 1–6, Nov 2014.

[135] S. Araghi, A. Khosravi, M. Johnstone, and D. C. Creighton. Q-learning method for controlling traffic signal phase time in a single intersection. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1261–1265, 2013.

[136] Ch. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992.

[137] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515, 11 1954.

[138] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.

[139] M. L. Littman, Th. L. Dean, and L. P. Kaelbling. On the complexity of solving markov decision problems. *CoRR*, abs/1302.4971, 2013.

[140] X. Du and J. Zhai. Algorithm trading using q-learning and recurrent reinforcement learning.

[141] I. The UCI KDD Archive, University of California. *KDD Cup 1999 Data*. Available at http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99/html/, Last Visit: April.10.2018.

[142] M. Elhamahmy, N. Hesham, and A. Imane. A new approach for evaluating intrusion detection system. In *Artificial Intelligent Systems and Machine Learning*, volume 2, pages 290–298, Dec 2010.

[143] J. Nitin. ns-3-tutorial, 07 2013.

[144] *ns-3 Introduction*. Available at https://www.nsnam.org/docs/ns-3-overview-july-2014.pdf, Last Visit: Dec.27.2018.

[145] P. Natesan and P. Balasubramanie. Multi stage filter using enhanced adaboost for network intrusion detection. *International Journal of Network security and its Applications*, 4:121–135, 05 2012.

[146] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, July 2009.

[147] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman. Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, volume 2, pages 12–26 vol.2, Jan 2000.

APPENDICES

Appendix A

Confidence Intervals

The Quantitative methods have been used to calculate the result's confidence intervals of the proposed Intrusion Detection Systems' (IDSs') solutions.

The most popular technique is the Confidence Interval (CI). CI is a measure to quantify uncertainty over any collected sample of data. It is defined as the estimated range of values within which a generated data lies with a specific probability. For instance, a probability of 95% implies that confidence of 95% that the collected data lies in a certain confidence interval (CI).

Simulated metrics such as Accuracy Rate (AR), Detection Rate (DR) and False Negative Rate (FNR) are measured by calculating the mean of a successive of n runs, with different simulation seed to ensure that there no correlation in the presented results.

The n simulation reads (Rs) are represented by $R_1, R_2, R_3, ..., R_n$ and their mean is represented in Equation A.1:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^{n} R_i \tag{A.1}$$

In order to calculate the CIs, it is a need to calculate the variance V_r^2 and the standard deviation σ which are represented in Equations A.2 and A.3 below:

$$V_r^2 = \frac{1}{n-1}\sum_{i=1}^{n}(R_i - \bar{R})^2 \tag{A.2}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_i - \bar{R})^2} \tag{A.3}$$

After finding the variance and the standard deviation, the z-statistic can be used to calculate the Reliability Factor (RF). For establishing CIs with the variance, the interval is created with the equation in Equation A.4 below where $Z_{\frac{\alpha}{2}}$ is the RF:

$$CI = \bar{R} \pm Z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \tag{A.4}$$

For α of 5% which is equivalent to a 95% CI, the RF is 1.96.

The upper and lower values of the 95% CI can be calculated as follows:

$$U(R) = \bar{R} + Z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \tag{A.5}$$

$$L(R) = \bar{R} - Z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \tag{A.6}$$

Based on our results, we took a sample of 10 simulation runs and mathematically tested them to determine the CIs.

Through the z-statistic, the RF and Equations (A.5) and (A.6), it is found that the collected results were within the calculated value of CI (95%).

Appendix B

Network Simulator-3 (NS-3)

NS-3 is a discrete-event in time network simulator for networked systems, targeted primarily for research, experimentation and educational use. It is free software, licensed under the GNU GPLv2 license, mainly supported for Linux, OS X, and FreeBSD and is publicly available for research, development, and use [96] [143].

The NS-3 simulation starts by defining the overall network topology includes the nodes, the links, the sources, the destinations, and attach the routing protocols to the desired nodes.

NS-3 simulation basics as mentioned in [144]:

- C++ functions schedule events to occur at specific simulation times with bindings available for Python.
- A simulation scheduler orders the event execution.
- Simulation time advances in discrete jumps from event to event.
- Simulation stops at a specific time or when events end.
- Simulation::Run() gets it all started

NS-3 is a free open-source software to be used by users; accordingly, the user can modify arguments, create different applications and add new protocols.

Appendix C

KDD CUP'99 Dataset

MIT Lincoln laboratory has collected real datasets for Intrusion Detection Systems (IDSs) evaluation purposes under the sponsorship of the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) [145].

The KDDCup99 dataset is subsets of the DARPA benchmark dataset and considered as the most widely used dataset for IDSs' evaluation [146][141].

KDDCup99 was prepared by the data captured in DARPA98 IDS evaluation program [147] which was ≈ four Gigabytes (GB) of compressed raw tcpdump data of seven weeks of network traffic which contains ≈ five million connection records with 100 bytes each [146].

KDD dataset consists of single connection vectors with 41 features in each and is labeled as normal or attack with one specific attack type.

Numerous attack types in KDDCup'99 datasets are grouped into four attack categories to put similar attack types into a single category in order to improve the detection rate such as the training and test sets contain 24 and 38 attacks respectively [146].

The attacks fall under four categories which are [146]:

1. Denial of Service Attack (DoS): Where the attacker makes the computing and memory resources busy in order to handle the legitimate requests as well as denies the users access to the machine.
2. User to Root Attack (U2R): When the attacker starts with access to a user account on the system (such as: sniffing passwords) as well as exploits vulnerabilities to get root access to the system.
3. Remote to Local Attack (R2L): Happens when an attacker exploits vulnerabilities to get access to a machine which does not have an account on it.
4. Probing Attack: It is an attack for gathering information about a network in order to circumvent its security controls.

It is important to mention that the test dataset includes attack types not in the training dataset which make the attacks' classification task more realistic.

KDDCup99 vectors features can be categorized into three sets [146]:

1. Basic features: It refers to all attributes that can be extracted from the TCP/IP connection. Such features cause some delay in the attacks' detection process.
2. Traffic features: Refers to features that depend on the window interval and is categorized into two types:
   - Same host features: Test the connections which belong to the same destination host as the current connection. It calculates statistics that belong to services, protocol behaviors, etc.
   - same service features: Test the connections which belong to the same service as the current connection.

   Both the same host and same service features are time based features.
3. Content features: Refers to features in order to look for suspicious behaviors in the data parts (such as a number of failed log-in attempts). It can be used specifically

in detecting R2L and U2R attacks where these attacks dont have any intrusion sequential patterns, as well as they, are embedded in the data parts of the packets.