Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 4731953, 16 pages
http://dx.doi.org/10.1155/2016/4731953

Research Article

WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks

Iman Almomani,[1,2] Bassam Al-Kasasbeh,[2] and Mousa AL-Akhras[2,3]

[1]*Computer Science Department, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia*
[2]*Computer Science Department/Computer Information Systems Department, King Abdullah II School for Information Technology (KASIT), The University of Jordan, Amman, Jordan*
[3]*Computer Science Department, College of Computation and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia*

Correspondence should be addressed to Iman Almomani; imomani@psu.edu.sa

Received 25 March 2016; Accepted 28 August 2016

Academic Editor: Hana Vaisocherova

Wireless Sensor Networks (WSN) have become increasingly one of the hottest research areas in computer science due to their wide range of applications including critical military and civilian applications. Such applications have created various security threats, especially in unattended environments. To ensure the security and dependability of WSN services, an Intrusion Detection System (IDS) should be in place. This IDS has to be compatible with the characteristics of WSNs and capable of detecting the largest possible number of security threats. In this paper a specialized dataset for WSN is developed to help better detect and classify four types of Denial of Service (DoS) attacks: Blackhole, Grayhole, Flooding, and Scheduling attacks. This paper considers the use of LEACH protocol which is one of the most popular hierarchical routing protocols in WSNs. A scheme has been defined to collect data from Network Simulator 2 (NS-2) and then processed to produce 23 features. The collected dataset is called WSN-DS. Artificial Neural Network (ANN) has been trained on the dataset to detect and classify different DoS attacks. The results show that WSN-DS improved the ability of IDS to achieve higher classification accuracy rate. WEKA toolbox was used with holdout and 10-Fold Cross Validation methods. The best results were achieved with 10-Fold Cross Validation with one hidden layer. The classification accuracies of attacks were 92.8%, 99.4%, 92.2%, 75.6%, and 99.8% for Blackhole, Flooding, Scheduling, and Grayhole attacks, in addition to the normal case (without attacks), respectively.

1. Introduction

Wireless Sensor Networks (WSN) have become increasingly an important field of research due to their wide range of real-time applications like critical military surveillance, battlefields, building security monitoring, forest fire monitoring, and healthcare [1]. A WSN consists of large number of autonomous sensor nodes, which are distributed in different areas of interest to gather important data and cooperatively transmit the collected data wirelessly to a more powerful node called sink node or Base Station (BS) [2, 3]. The data transmitted across the network depend on specialized WSN protocols. Therefore, protecting WSN from different security threats is essential. Unfortunately, achieving this objective becomes a major challenge because of the constrained resources of WSNs including battery energy, memory, and processing capabilities [4]. Such limiting characteristics make traditional security measures like cryptography not always sufficient for such networks.

WSNs are highly vulnerable to attacks, due to their open and distributed nature and limited resources of the sensor nodes. Moreover, in WSNs packets broadcasting has to be done frequently, sensor nodes can be deployed randomly in an environment so an attacker adversary can be easily injected to a WSN [5].

An attacker can compromise a sensor node, eavesdrop messages, inject fake messages, alter the integrity of the data, and waste network resources. Denial of Service (DoS) attack is considered one of the most general and dangerous attacks that threaten WSN security. This attack has several forms

and its main objective is to interrupt or suspend the services provided by WSNs [6, 7].

Because the process of avoiding or preventing security threats cannot be always successful, an Intrusion Detection System (IDS) is needed to detect known and unknown attacks and alert sensor nodes about them [3, 4]. IDS allows detecting suspicious or abnormal activities and triggers an alarm when an intrusion occurs. The implementation of IDSs for WSNs are more difficult than other systems because sensor nodes are usually designed to be tiny and cheap, and they do not have enough hardware resources. Additionally, there is no specialized dataset that contains normal profiles and attacks in WSN that can be used to detect an attacker signature [3]. Considering the above challenges, there are mainly two conditions while designing IDS for WSNs: The IDS must be of high degree of accuracy in detecting an intruder that includes unknown attacks, and it also must be lightweight to ensure minimum overhead on the infrastructure of WSNs [8].

In this paper a specialized WSN dataset is constructed to characterize four types of DoS attacks in addition to the normal behavior when no attacks exist. WSNs' characteristics and challenges were considered when Low Energy Aware Cluster Hierarchy (LEACH) [9] routing protocol was used in this study. This choice was made since LEACH is one of the most popular hierarchical routing protocols in WSNs that consumes limited energy and is characterized by its simplicity. The constructed dataset is called WSN-DS.

The rest of paper is organized as follows. Section 2 provides an overview of LEACH protocol, IDSs, and reviews related work. Section 3 analyzes LEACH protocol mathematically; Section 4 describes the extracted features of the constructed dataset. Section 5 models different attacks. Section 6 presents the experimental results obtained from IDS and discusses the importance of the achieved results. Conclusions and avenues for future work are presented in Section 7.

2. Background and Related Works

This section presents an overview of LEACH protocol, LEACH-based protocols, DoS, and IDS in WSNs.

2.1. LEACH Protocol Overview. LEACH is a hierarchical routing protocol used in WSNs to increase the network's lifetime [9–11]. LEACH is a clustering, adaptive, and self-organizing protocol. LEACH assumes that BS is fixed and located far from sensor nodes. Additionally, all sensor nodes are homogeneous and have limited energy and memory. Sensors can communicate among each other and they can communicate directly with the BS. The main idea of LEACH protocol is to organize nodes into clusters to distribute the energy among all nodes in the network. Also, in each cluster there is a node called Cluster Head (CH) which aggregates the data received from sensors within its cluster and forward them to the BS. Figure 1 shows the structure of nodes in LEACH routing protocol. Each round in LEACH protocol consists mainly of two phases: setup phase and steady-state phase. In the setup phase, clusters are formed, whereas in the steady-state phase, sensed data will be transferred to the sink node [12].

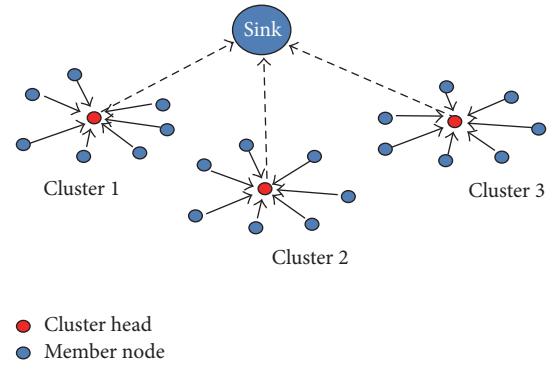

Figure 1: Nodes structure in LEACH routing protocol.

At the beginning of the setup phase, every node generates a random number between 0 and 1, and it then computes a threshold formula $T(n)$, as shown in (1). If the selected random number is less than the threshold value, the node becomes a CH:

$$T(n) = \begin{cases} \dfrac{0}{1 - p \times \left(r \bmod p^{-1}\right)}, & \forall_n \in N \\ 0, & \text{otherwise}, \end{cases} \quad (1)$$

where p is the CH probability (usually in LEACH a node becomes CH with a probability of 0.05), N: is the set of nodes that have not been a CH, in the last $1/p$ rounds, and r is the current round.

CH in the first round cannot be CH again in the next $1/p$ rounds. After $1/p - 1$ rounds, the threshold value becomes 1 for any sensor node that has not been CH yet, and after $1/p$ rounds, all nodes are eligible again to become CHs. Once CHs are assigned for all clusters, each CH will broadcast an advertisement message (ADV_CH) to the rest of nodes using Carrier Sense Multiple Access-Media Access Control (CSMA-MAC) protocol [9]. After receiving ADV_CH message, each node decides to which cluster it belongs by selecting a CH based on the Received Signal Strength Indication (RSSI) of the advertisement message, the node then sends a (JOIN_REQ) message to the selected CH with the highest RSSI. Each node uses CSMA-MAC protocol to transmit its selection [9, 10]. During the setup phase, all CHs keep their receivers ON. After clusters formation, each CH creates a Time Division Multiple Access (TDMA) schedule according to the number of nodes in its cluster called Cluster Members (CM) and broadcasts it to them.

During steady-state phase, each sensor node collects data and transmits them to its CH during its allocated time slot according to the TDMA schedule. CHs receive all the data and aggregate them before sending them to the BS. After a predetermined time, the network starts another round by going back to the setup and steady-state phases again [9].

2.2. LEACH-Based Protocols. LEACH was and still is studied in enormous number of research articles. The authors in [13]

provided a review of 27 clustering and routing techniques based on LEACH protocol for WSNs that includes a comprehensive discussion and comparisons among them. The authors in [14, 15] highlighted LEACH protocol and presented fifteen LEACH improved versions introduced in the literature. The papers have compared some features of several variants of LEACH protocol not empirically but based on their descriptions. In [16] the author proposed and evaluated two new clustering-based protocols for heterogeneous WSNs that were built based on LEACH protocol by considering three types of nodes with different battery energies, which was the source of heterogeneity in the author's protocols.

LEACH-ICE (LEACH Inner Cluster Election) algorithm based on LEACH algorithm was introduced in [17]. The threshold function of the node selected as CH is adjusted. Also, direct communication with the BS occurs when a node is closer to the BS. To improve the clustering mechanism, LEACH-ICE elects a new CH inside the cluster when the resident energy of the current CH is lower than a predefined threshold.

In [18] the authors proposed an energy efficient secondary CH selection algorithm for WSN. By controlling the distances among the CHs, a uniform distribution of CHs is satisfied. Two-level hierarchy mode was applied to transmit data to the BS. LEACH is compared with the improved LEACH-TLCH method. Simulation results show that the improved method can reduce the network consumption of energy and lengthen the network's lifetime.

In [19], a distributive Energy Neutral Clustering (ENC) protocol was proposed to group the network into several clusters, with the goal of providing perpetual network operation. ENC employs a novel Cluster Head Group (CHG) mechanism that allows a cluster to use multiple CHs to share heavy traffic load and to reduce the frequency of cluster reformation. An extension to ENC based on convex optimization techniques of the number of clusters was proposed to group the network into equal-sized clusters to maximize network information gathering. According to the authors' experiments, the proposed protocol can successfully prevent sensors from shutting down due to excessive usage of energy.

2.3. DoS and IDS in WSN. As mentioned earlier, DoS is a common attack that could have a severe impact on WSN's functionalities and services [20]. Many different types of DoS attacks have been identified so far, for example, Blackhole attack, Grayhole attack, Flooding attack, and Wormhole. The seriousness of DoS attack stems from the fact that most WSN applications require the deployment of a sensor node in harsh environments where they are far away and difficult to be controlled [20, 21]. Recently, many researches are going on in an attempt to find solutions for DoS attacks, but mainly they have tackled one or two forms of these attacks but not the majority [2, 22–24]. Moreover, they offer partial solutions and they cannot be applied concurrently because they will consume high energy, which is not practical in WSNs [2, 25]. Therefore, a mechanism should be found to identify different behaviors of DoS attacks and classify them to take effective countermeasures.

Cryptography is a security mechanism that is used for protecting WSN against external attacks. It ensures many security services including integrity and authentication by checking the data packet source and its contents using several techniques such as symmetric encryption, public key cryptography, and hash functions [25]. These techniques cannot be used to detect internal attacks when security keys are exposed to the attacker which uses them to perform encryption and decryption of messages' contents. Consequently, such techniques serve as first line of defense [5]. Attackers always attempt to launch new and unknown attacks in more than one way; therefore, it is necessary to create an efficient IDS, which acts as a second line of defense to detect known and unknown attacks and alert sensor nodes about them. IDS allows detecting suspicious or abnormal activities and triggers alarms when intrusions are detected [26].

The National Institute of Standards and Technology (NIST) [27] categorized intrusion detection into two main approaches: anomaly detection and misuse detection. In anomaly detection the system depends on prior knowledge of normal behavior of the network which will be then compared with its current activities. In misuse detection, the system depends on prior knowledge of attack signatures. It compares the signature with the current activities in the network.

IDS has become an important security component of WSNs; however, the implementation of IDS in WSNs introduces number of challenges that can have negative impact on WSN performance [28]. It is inefficient to use IDS in every sensor node due to the resource-constrained nature of such nodes. IDS components should be installed in places where sensor nodes can be followed to be able to defend against certain threats to the network. IDS is also used in WSNs where huge amount of traffic is transmitted; therefore, there is a possibility an intrusion could be missed as sensor nodes generally have restrictions in handling huge data in the network.

There are two main components of IDS, features extraction and modeling algorithm. Features extraction defines measured attributes that are linked to the IDS functionalities. Modeling algorithm is the main component; the accuracy and the efficiency of detecting and responding to intrusions depend on the modeling algorithm. IDS may have components that depend on the network characteristics and possible intrusions [29]. Most of IDSs have six common components as shown in Figure 2:

(1) Monitoring component: which is used for local activity monitoring or for monitoring neighbor sensor nodes. This component mostly monitors internal activities, traffic patterns, and resource utilization.

(2) Analysis component: which contains all records of normal and abnormal behaviors for all nodes in the network [30].

(3) Detection component: which is the main component that is built based on the modeling algorithm. It works after analyzing network behaviors. Decisions are made to declare such behaviors as malicious or not [31].

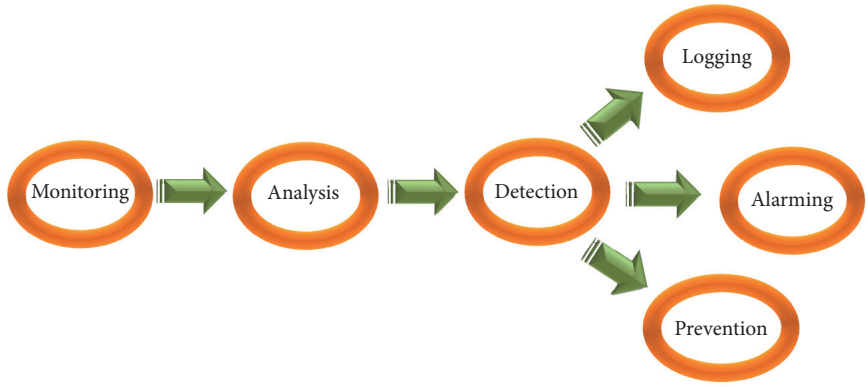

Figure 2: IDS components.

The other three components of IDS consist of actions that can be taken, either one, two, or all of them [32]:

(4) Logging: storing each packet in a log file so that security administrator can use it for later analysis.

(5) Alarming: a responding generating component in case of detection of an intrusion. The response may trigger an alarm to announce the misbehaving node(s).

(6) Prevention: an advanced step that can be added to IDS to enable it to take an action to prevent dealing with an attack once detected. This can be done, for example, by excluding harmful nodes from the network [30].

Designing a specialized dataset for WSN to achieve better detection and classification of DoS attacks is the main aim of this paper. The authors in [30] presented current IDSs and a comparison among them. The authors revised mechanisms, attacks, and evaluation metrics but without mentioning the use of specialized datasets. The comparison depended on the type of IDSs, whether it is anomaly-based, signature-based, hybrid, or cross layer.

Knowledge Discovery and Data Mining Tools Competition (KDD) dataset [33] was constructed for Local Area Network (LAN). KDD is not specialized for wireless in general and WSN in particular, even though many researchers have used it to deal with fraud and intrusion detection [34].

Anomaly, signature, and hybrid-based IDSs have been reviewed in [35]. Mainly KDDCup-99 was used in these IDSs. For example, in the eight studied hybrid-based IDSs, four of them have used KDDCup-99 and the rest have used real data samples.

Other researches which also considered KDD in their analysis and classifications can be found in [36–38].

It can be concluded that there is no specialized dataset for WSN that has been reported in the literature for detecting and classifying as many DoS as possible. Therefore, there is an urgent need to define a labelled, specialized dataset that successfully characterizes WSN to help in studying normal and anomaly behaviors. The construction and testing of such dataset are proposed in this paper.

3. LEACH Mathematical Analysis

To ensure the correctness of the constructed dataset called WSN-DS, a mathematical analysis has been conducted to all LEACH phases and then has been compared to the results of simulation in case of normal situation when there is no DoS attack. The terms used in LEACH's mathematical model are listed as follows:

LEACH Mathematical Model Terms

N: number of sensor nodes in WSN

Si: senor node i

NC: number of CHs

CM: number of members within a cluster

ADV-CH-SENT: number of advertisement messages sent by CH

ADV-CH-RCVD: number of advertisement messages received by sensor nodes

JOIN-REQ-SENT: number of join request messages sent by sensor nodes

JOIN-REQ-RCVD: number of join request messages received by CHs

TDMA-SENT: number of TDMA schedules sent by CHs

TDMA-RCVD: number of TDMA schedules received by sensor nodes

NO-DATA-PKT: number of data packets received by a CH

3.1. Advertisement Phase. Theorem 1 calculates the number of advertisement messages that are sent by CHs and received by CMs in a specific round as follows.

Theorem 1. *In the advertisement phase of LEACH, maximum ADV-CH-SENT in a specific round is NC and the maximum ADV-CH-RCVD is* $(N - 1) * NC$.

Table 1: Comparison between the mathematical model and simulation results.

Round	Number of clusters	ADV-CH-Sent		ADV-CH-Rcvd		Join-Req-Sent		Join-Req-Rcvd		BS receives	
		Math	Sim.	Math	Sim.	Math	Sim.	Math	Sim.	Math	Sim.
1	4	4	4	396	396	96	96	96	96	238	238
2	2	2	2	198	198	98	98	98	98	53	53
3	3	3	3	297	297	97	97	97	97	126	126
4	2	2	2	198	198	98	98	98	98	59	59
5	7	7	7	693	693	93	93	93	93	563	563
6	6	6	6	594	594	94	94	94	94	516	516
7	4	4	4	396	396	96	96	96	96	268	268
8	4	4	4	396	396	96	96	96	96	291	291
9	5	5	5	495	495	95	95	95	95	447	447
10	7	7	7	693	693	93	93	93	93	695	695
11	6	6	6	594	594	94	94	94	94	456	456
12	6	6	6	594	594	94	94	94	94	363	363
13	1	1	1	99	99	99	99	99	99	13	13
14	7	7	7	693	693	93	93	93	93	629	629

Table 2: Applying Theorem 3 equation to round 1 of simulation round.

Cluster number	Number of nodes within CH	Number of packets received(No-DATA-PKT)	Number of packets sent to BS
Cluster 1	25	1200	48
Cluster 2	30	1230	41
Cluster 3	8	880	111
Cluster 4	33	1254	38
Applying Theorem 3 equation $\sum_{i=1}^{NC}$(NO-DATA-PKT/CM of CHi)			Total: 238

Proof. According to LEACH, each CH in each round is supposed to broadcast an advertisement message to the rest of nodes. Therefore, in case of having NC cluster heads, then ADV-CH-SENT equals NC. On the other hand, these advertisement messages (NC) will be received by all sensor nodes (N) except the CH node itself which equals $(N - 1) *$ NC. □

3.2. Cluster Setup Phase. Theorem 2 calculates the number of join request messages sent by sensor nodes and received by CHs in order to associate with them.

Theorem 2. *In clusters setup phase of LEACH, the maximum JOIN-REQ-SENT equals JOIN-REQ-RCVD which is N − NC.*

Proof. According to LEACH, once each sensor node has decided to which cluster it will belong, then it informs its CH by sending a (JOIN_REQ) message. Therefore, all sensor nodes (N) except CHs (NC) will send (JOIN_REQ) messages (N − NC) and these messages will also be received by CHs. □

3.3. Data Transmission Phase. Theorem 3 calculates the amount of sensed data packets that are delivered to the BS at the end of each round.

Theorem 3. *In the data transmission phase of LEACH, at the end of each round, BS receives* $\sum_{i=1}^{NC}$*(NO-DATA-PKT/CM of CH*i*) packets.*

Proof. According to LEACH, when the CH receives the sensed data from the sensors nodes (CMs) according to their time slots assigned by TDMA schedule, it aggregates them into one packet and sends it to the BS. Throughout the round, the number of packets sent to the CH from CMs is (NO-DATA-PKT) but due to the aggregation process only (NO-DATA-PKT/CMs of CHi) packets will be sent to the BS. Having NC of CHs, then the overall data packets received by BS are $\sum_{i=1}^{NC}$(NO-DATA-PKT/CM of CHi). □

3.4. Comparison between Mathematical Model and Simulation Results. To confirm the correctness of the simulation which is used to collect data to construct the dataset, a comparison is performed between the mathematical analysis and simulation results. The comparison will be based on sample of the simulation results representing the first 14 rounds as after this round nodes start to die. In the first 14 rounds, the number of alive nodes is 100. Table 1 shows this comparison. The mathematical results were obtained by applying the equations in Theorems 1–3, while the simulation results were obtained from Network Simulator 2 (NS-2) simulator.

For more clarification, Table 2 presents how the mathematical formula of Theorem 3 is applied to a sample round (Round 1) in one of the simulation scenarios to calculate the number of received data packets by BS.

Table 1 shows 100% match between the mathematical model and the simulation results. This is due to the behavior of LEACH protocol which implements dynamic

Table 3: Observations for five different simulation scenarios (A–E) when determining the number of nodes monitored by each node.

Number of neighbors to watch	Max number of monitors for a specific node					Min number of monitors for a specific node					Number of overall monitored nodes				
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
3	6	7	7	6	7	0	0	0	1	0	97	99	99	100	97
4	7	9	8	8	9	0	0	0	1	0	99	99	99	100	99
5	10	9	10	10	10	1	1	1	1	2	100	100	100	100	100
6	11	12	11	10	13	1	1	1	2	2	100	100	100	100	100

TDMA Scheduling technique at the data transmission level. Additionally, it uses both Code Division Multiple Access (CDMA) and CSMA codes to avoid and reduce collisions and interferences that may exist in the network.

4. WSN-DS Dataset Description and Creation

In order to build the dataset and collect the required data from the sent and received packets within WSN, a monitoring service is needed with minimum cost. On the other hand, we need to guarantee that necessary data related to the network which help in detecting, classifying, and then preventing different possible attacks are collected. In this research, to distribute the load among sensor nodes, each sensor will take part in the monitoring process and should be able to monitor set of its neighbors. The challenge was how to find the suitable number of nodes to be watched by a sensor node in order to monitor all network sensors. Many experiments have been conducted to decide on this number and the summary of the results is shown in Table 3.

When each sensor node has watched 3 nodes of its neighbors, it has been noticed that the largest number of sensor nodes which could be monitored by a single node was seven. In other words, the BS has received seven different reports about the same node from seven different watching nodes. To make sure that the received information are correct, these reports could be checked for consistency. In some scenarios, some sensor nodes were not monitored by any sensor. This indicates that monitoring 3 neighboring nodes is not enough to get information about all network sensor nodes.

Additionally, an improvement has occurred when 4 neighbors are being watched. But only when the number is 5, all sensor nodes are being watched in all 5 scenarios. Similar results have been obtained when a sensor node was watching 6 of its neighbors. Consequently, it has been found that monitoring 5 neighbors is enough to get information about all nodes in the network and there is no need to increase the computational complexity by going further.

Choosing 5 neighbors to be monitored is done at the beginning of the simulation. All nodes broadcast a *Hello* message. Accordingly, each node selects the first 5 nodes it heard from. Then it monitors them over the simulation period, so that each node sends a report to its CH at the end of each round. Then the CH sends the received reports to the BS. For security purposes and in case of suspecting the CH and having one monitor for this node (one report), these reports could be sent directly to the BS at the expense of consuming more energy if this node is further from the BS than the CH. After deep study of LEACH routing protocol, we have succeeded to extract 23 attributes to help in identifying the status of each node in the network, These attributes are listed as follows.

WSN-DS Dataset Attributes

Node ID: a unique ID to distinguish the sensor node in any round and at any stage. For example, node number 25 in the third round and in the first stage is to be symbolized as 001 003 025.

Time: the current simulation time of the node.

Is CH? A flag to distinguish whether the node is CH with value 1 or normal node with value 0.

Who CH? The ID of the CH in the current round.

RSSI: Received Signal Strength Indication between the node and its CH in the current round.

Distance to CH: the distance between the node and its CH in the current round.

Max distance to CH: the maximum distance between the CH and the nodes within the cluster.

Average distance to CH: the average distance between nodes in the cluster to their CH.

Current energy: the current energy for the node in the current round.

Energy consumption: the amount of energy consumed in the previous round.

ADV_CH send: the number of advertise CH's broadcast messages sent to the nodes.

ADV_CH receives: the number of advertise CH messages received from CHs

Join_REQ send: the number of join request messages sent by the nodes to the CH.

Join_REQ receive: the number of join request messages received by the CH from the nodes.

ADV_SCH send: the number of advertise TDMA schedule broadcast messages sent to the nodes.

ADV_SCH receives: the number of TDMA schedule messages received from CHs.

Rank: the order of this node within the TDMA schedule.

$N \rightarrow$ Network Size
SN $\rightarrow$ Sensor Node
MN $\rightarrow$ Malicious Node
CH $\rightarrow$ Cluster Head
BS $\rightarrow$ Base Station
CM $\rightarrow$ Cluster Member
NC $\rightarrow$ Cluster Heads list
$x \rightarrow$ Integer value between 0 and $N - 1$
$\forall$ SN$_i$, $0 < i \leq N$, compute T(SN$_i$) and random r_{SN_i}
IF ($r_{\text{SN}_i} < T(\text{SN}_i)$) THEN
    SN$_i$ = CH
ELSE
    SN$_i$ = CM
ENDIF
$\forall$ CH j, $j \in$ NC
{
 CH j broadcasts the advertisement message (ADV_CH)
 x CMs will join CH j
 CH j creates TDMA schedule
 x CMs send data to CH j in the corresponding TDMA time slot
}
IF CH j = MN THEN
    Performs the attack by dropping all packets
ELSE
    Sends aggregated data to BS
ENDIF

Algorithm 1: Model of Blackhole attack.

Data sent: the number of data packets sent from a sensor to its CH.

Data received: the number of data packets received from CH.

Data sent to BS: the number of data packets sent to the BS.

Distance CH to BS: the distance between the CH and the BS.

Send Code: the cluster sending code.

Attack Type: type of the node. It is a class of five possible values, namely, Blackhole, Grayhole, Flooding, and Scheduling, in addition to normal, if the node is not an attacker.

5. Attack Models

Four types of DoS attacks in LEACH protocol are implemented in the constructed dataset; Blackhole, Grayhole, Flooding, and Scheduling attacks. This section models each one of these attacks. To ensure proper distribution of the attacker nodes, the network terrain has been divided into 10 regions. Then the attackers' ratios according to the simulation scenario were distributed randomly within these regions.

5.1. Blackhole Attack. Blackhole attack is a type of DoS attack where the attacker affects LEACH protocol by advertising itself as a CH at the beginning of the round. Thus, any node that has joined this CH during this round will send the data packets to it in order to be forwarded to the BS. The Blackhole attacker assumes the role of CH and it will keep dropping these data packets and not forwarding them to the BS [39–41]. Algorithm 1 shows the algorithm of Blackhole attack.

To implement this attack in the simulation environment, several attackers' intensities (10%, 30%, and 50%) have been injected randomly to perform Blackhole attack. These attackers which act as CHs will drop all the packets relayed through them in their way to the BS.

5.2. Grayhole Attack. Grayhole attack is a type of DoS attack where the attacker affects LEACH protocol by advertising itself as a CH for other nodes. Therefore, when the forged CH receives data packets from other nodes, it drops some packets (randomly or selectively) and prevents them from reaching the BS [40–42]. Algorithm 2 shows the algorithm of Grayhole attack.

Similar to Blackhole attack, 10%, 30%, and 50% of the sensor nodes are injected randomly to implement the Grayhole attack. The decision whether to forward a specific packet or not is also devised randomly. But the decision can be done selectively based on the sensitivity of the sensed data carried by the packet.

5.3. Flooding Attack. Flooding attack is a type of DoS attack where the attacker affects LEACH protocol in more than one way. This research studies the impact of Flooding attack by sending large number of advertising CH massages (ADV_CH) with high transmission power. Consequently, when sensors receive large number of ADV_CH messages,

$N \rightarrow$ Network Size
SN $\rightarrow$ Sensor Node
MN $\rightarrow$ Malicious Node
CH $\rightarrow$ Cluster Head
BS $\rightarrow$ Base Station
CM $\rightarrow$ Cluster Member
NC $\rightarrow$ Cluster Heads list
$x \rightarrow$ Integer value between 0 and $N - 1$
$\forall$ SN$_i$, $0 < i \leq N$, compute T(SN$_i$) and random r_{SN_i}
IF ($r_{\text{SN}_i} < T(\text{SN}_i)$) THEN
    SN$_i$ = CH
ELSE
    SN$_i$ = CM
ENDIF
$\forall$ CHj, $j \in$ NC
{
CHj broadcasts the advertisement message (ADV_CH)
x CMs will join CHj
 CHj creates TDMA schedule
 x CMs send data to CHj in the corresponding TDMA time slot
}
IF CH = MN THEN
    Performs the attack by dropping some packets (randomly or selectively)
ELSE
    Sends aggregated data to BS
ENDIF

Algorithm 2: Model of Grayhole attack.

this will consume sensors' energy and waste more time to determine which CH to join. Moreover, the attacker attempts to cheat victims to choose it as a CH, especially those nodes that are located on a far distance from it in order to consume their energy [40, 43]. Algorithm 3 shows the algorithm of Flooding attack.

Flooding attack has been implemented in several ways in the simulation environment. In some experiments 10 ADV_CH messages were sent by the attacker; other scenarios consider 50 ADV_CH messages to be sent or a random number between 10 and 50. The idea is when more ADV_CH messages are sent, more messages will be received and more energy will be consumed. We have already studied in [44] the impact of Flooding attack on WSN lifetime. The energy consumption was shown in each round using several attackers' ratios.

5.4. Scheduling Attack. Scheduling attack was introduced in a previous study of the authors [44]. Scheduling attack occurs during the setup phase of LEACH protocol, when CHs set up TDMA schedules for the data transmission time slots. The attacker which acts as a CH will assign all nodes the same time slot to send data. This is done by changing the behavior from broadcast to unicast TDMA schedule. Such change will cause packets collision which leads to data loss. Algorithm 4 shows the algorithm of Scheduling attack.

The implementation of Scheduling attack is performed by setting the same time for all Cluster Members to send their data packets. Other scenarios assign every two nodes the same time or every five nodes the same time.

In [44] it has been shown that the risk of DoS attackers on LEACH protocol services could be significant. The attackers can influence the network in more than one way, through wasting the nodes' energy or dropping their data packets. This badly affects the services provided by WSN. Therefore, a methodology to detect such attacks and protect different services provided by WSN is urgently required.

Section 6 illustrates the importance of studying normal and anomaly (under attack) behaviors of WSN protocols and presenting them through a specialized dataset (WSN-DS). WSN-DS allows several intelligent and data mining approaches to be applied for the aim of better detection and classification of DoS attacks. As a result, sensor nodes will be more experienced with the normal behaviors and attackers' signatures and will be able to make proper decisions at the right time. In this research ANN is applied to test the constructed dataset and measure its accuracy in detecting and classifying four types of DoS attacks.

6. Experiments and Results

In this paper, WSN-DS, a specialized dataset for WSN to detect DoS attacks, was constructed. LEACH protocol was used to collect the dataset because it is one of the most common and widely used routing protocols in WSNs. WSN-DS contains 374661 records that represent four types of DoS attacks: Blackhole, Grayhole, Flooding, and Scheduling attack, in addition to the normal behavior (no-attack) records. Table 4 shows sample from WSN-DS dataset to help in detecting and classifying DoS attacks.

N → Network Size
SN → Sensor Node
MN → Malicious Node
CH → Cluster Head
BS → Base Station
CM → Cluster Member
NC → Cluster Heads list
x → Integer value between 0 and $N - 1$
$\forall \text{SN}_i, 0 < i \leq N$, compute $T(\text{SN}_i)$ and random r_{SN_i}
IF ($r_{\text{SN}_i} < T(\text{SN}_i)$) THEN
    SN_i = CH
ELSE
    SN_i = CM
ENDIF
$\forall$ CHj, $j \in$ NC
{
 IF CHj = MN THEN
    CHj broadcasts a lot of advertisement messages (ADV_CH) with high transmitting power.
 ELSE
    CHj broadcasts normal advertisement message (ADV_CH)
 ENDIF
x CMs will join CHj
 CHj creates TDMA schedule
 x CMs send data to CHj in the corresponding TDMA time slot
}

Algorithm 3: Model of Flooding attack.

N → Network Size
SN → Sensor Node
MN → Malicious Node
CH → Cluster Head
BS → Base Station
CM → Cluster Member
NC → Cluster Heads list
x → Integer value between 0 and $N - 1$
$\forall \text{SN}_i, 0 < i \leq N$, compute $T(\text{SN}_i)$ and random r_{SN_i}
IF ($r_{\text{SN}_i} < T(\text{SN}_i)$) THEN
    SN_i = CH
ELSE
    SN_i = CM
ENDIF
$\forall$ CHj, $j \in$ NC
{
CHj broadcasts the advertisement message (ADV_CH)
x CMs will join CHj
IF CHj = MN THEN
    CHj performs the attack by creating the TDMA schedule and give all nodes same time slot to send data
ELSE
    CHj creates normal TDMA schedule
ENDIF
x CMs send data to CHj in the corresponding TDMA time slot
CHj sends aggregated data to BS
}

Algorithm 4: Model of Scheduling attack.

Table 4: Sample from WSN-DS dataset.

Id	Time	Is_CH	Who CH	Dist_To_CH	ADV_S	ADV_R	JOIN_S	JOIN_R	SCH_S	SCH_R	Rank	DATA_S	DATA_R	Data_Sent_To_BS	Dist_CH_To_BS	Send_code	Consumed energy	Attack type
106079	303	1	106079	0	1	3	0	75	1	0	0	0	1350	7	108.34705	0	1.64035	Grayhole
107033	353	1	107033	0	1	3	0	71	1	0	0	0	1349	9	162.5505	0	2.03296	Grayhole
115021	753	1	115021	0	1	5	0	59	1	0	0	0	1298	0	0	0	0.00721	Blackhole
117044	853	1	117044	0	1	4	0	54	54	0	0	0	0	0	0	0	0.00723	Scheduling
103043	153	1	103043	0	1	4	0	47	1	0	0	0	1269	14	145.08942	0	1.88023	Grayhole
105005	253	1	105005	0	1	9	0	47	1	0	0	0	1170	7	137.59248	0	0.92063	Grayhole
110024	503	1	110024	0	1	9	0	35	1	0	0	0	1200	15	113.27654	0	2.0577	Grayhole
101041	53	1	101041	0	1	0	0	34	1	0	0	0	1258	0	0	0	0.00225	Blackhole
102040	103	1	102040	0	1	2	0	31	1	0	0	0	1240	0	0	0	0.00728	Blackhole
201061	1003	1	201061	0	1	7	0	31	1	0	0	0	1240	0	0	0	0.00719	Blackhole
118058	903	1	118058	0	1	5	0	27	27	0	0	0	0	0	0	0	0.00724	Scheduling
103003	153	1	103003	0	1	4	0	22	1	0	0	0	1166	29	85.19787	0	2.06959	Grayhole
111050	553	0	111093	15.17406	0	2	1	0	0	1	10	22	0	0	0	1	0.04156	Normal
111057	553	0	111093	15.91573	0	2	1	0	0	1	3	22	0	0	0	1	0.04172	Normal
402054	1253	1	402054	0	6	22	0	0	0	0	0	0	0	13	142.10787	0	0.24255	Flooding
402063	1253	1	402063	0	6	28	0	0	0	0	0	0	0	13	123.96292	0	0.23082	Flooding
402069	1253	1	402069	0	6	22	0	0	0	0	0	0	0	13	93.93772	0	0.21998	Flooding
118046	903	1	118046	0	1	5	0	21	21	0	0	0	0	0	0	0	0.00722	Scheduling
110044	503	1	110044	0	1	9	0	20	1	0	0	0	1087	23	121.40806	0	1.92349	Grayhole
117061	853	1	117061	0	1	9	0	20	1	0	0	0	1131	0	0	0	0.00728	Blackhole
201021	1003	1	201021	0	1	7	0	20	1	0	0	0	1140	0	0	0	0.0072	Blackhole
101021	53	1	101021	0	1	0	0	17	1	0	0	0	1105	0	0	0	0.00225	Blackhole
117039	853	1	117039	0	1	4	0	14	14	0	0	0	0	0	0	0	0.00723	Scheduling
117095	853	1	117095	0	1	4	0	14	14	0	0	0	0	0	0	0	0.00722	Scheduling
103029	153	1	103029	0	1	3	0	10	1	0	0	0	960	0	0	0	0.00724	Blackhole
118031	903	1	118031	0	1	5	0	5	5	0	0	0	0	0	0	0	0.00736	Scheduling
111053	553	0	111028	19.42763	0	2	1	0	0	1	37	32	0	0	0	2	0.1789	Normal
111051	553	0	111028	21.35118	0	2	1	0	0	1	33	32	0	0	0	2	0.057	Normal
111055	553	0	111028	36.99519	0	2	1	0	0	1	31	32	0	0	0	2	0.0582	Normal
111054	553	0	111028	43.03687	0	2	1	0	0	1	24	32	0	0	0	2	0.05904	Normal
111060	553	0	111028	40.20187	0	2	1	0	0	1	20	32	0	0	0	2	0.05894	Normal

Table 5: Ns-2 simulation parameters.

Parameter	Value
Number of nodes	100 nodes
Number of clusters	5
Network area	100 m × 100 m
Base station location	(50, 175)
Size of data packet	500 bytes
Size of packet header	25 bytes
Maximum transmission range	200 m
Routing protocol	LEACH
MAC protocol	CSMA/TDMA
Simulation time	3600 s
Initial energy (in joule)	5, 50
Attackers' intensities	10%, 30%, 50%

In order to gather the required data, NS-2 was used [45]. Simulation parameters are summarized in Table 5.

This section shows the results obtained from the dataset collected as described in Section 4. Waikato Environment for Knowledge Analysis (WEKA) Toolbox was used in the simulation experiments to evaluate the proposed dataset. WEKA is an open source data mining software suite built using Java programming language and developed at the University of Waikato in New Zealand. Data mining algorithms in WEKA could be applied to datasets and be called using either WEKA's interface or user customized Java code. WEKA contains a lot of algorithms for data preprocessing, clustering, classification, association rules, regression, and visualization [46, 47].

Experiments were conducted on an Intel® Core™ i5-4210U CPU @ 1.70 GHz 2.40 GHz, 8.00 GB RAM with Windows 8.1 64-bit Operating System.

Because different performance metrics are appropriate in different settings, in this paper seven performance metrics are used: True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Overall Accuracy (A), Precision (P), and Root Mean Square Error (RMSE).

TPR represents the rate of attack cases identified correctly, TNR represents the rate of normal (no-attack) cases identified correctly, FPR represents the rate of no-attack cases identified as attacks by the system, and FNR represents the rate of attack cases identified as normal ones. A is the total rate of correct decisions whether identifying an attack correctly or deciding there is no attack when really there is no attack. P represents the predicted positive cases that were correctly classified; RMSE provides information on the efficiency that indicates the difference between the outputs and the targets. Lower values of RMSE indicates more accurate evaluation. Zero means no error:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{2}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FN} + \mathrm{TP}}, \tag{3}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{4}$$

Table 6: Dataset separated 60% training set and 40 testing sets using holdout method.

The attack type	Training set (60%)	Testing set (40%)
Blackhole	6029	4020
Grayhole	8758	5838
Flooding	1988	1324
Scheduling	3982	2656
Normal	204039	136027
Sum	*224796*	*149865*

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}, \tag{5}$$

$$A = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{6}$$

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (O_i - T_i)^2}{n}}, \tag{8}$$

where TP is the number of attack cases classified correctly as attacks. TN is the number of normal (no-attack) cases classified correctly as normal (no-attack). FP is the number of normal (no-attack) cases classified incorrectly as attacks. FN is the number of attack cases classified incorrectly as normal (no-attack). O_i and T_i are the output and target values, respectively, and n is the total number of data points.

The classification results of this dataset were obtained through a number of test cases applied using Artificial Neural Networks (ANNs), which can be built in several ways. ANN is used as a classifier were the 23 attributes extracted from the simulation experiments are used as inputs and the type of attack, including the normal case, is used as output. ANN training algorithm includes a built-in procedure to help minimizing the error between the neural network output and the desired output. Its iterative training procedure terminates when that error reaches a value below a predetermined threshold. After the training phase, the trained neural network is used on the test dataset to check its generalization accuracy.

We are extracting different results with two ANNs test options. The first one is by using holdout method where the dataset is separated to 60% training data and 40% testing data. Table 6 shows data separation using holdout method.

The second option is by using 10-Fold Cross Validation which separates the training dataset into 10 equal parts. This method trains ANN using nine of the 10 parts and evaluates it with the remaining part. The same process is repeated for all 10 parts using a sliding window to determine the test set and the remaining parts are used for training the ANN. After the completion of the 10 iterations, the results are compiled and averages are computed. The main advantage of the 10-Fold Cross Validation is using all records in the dataset alternately for both training and testing. On the other hand, it is computationally expensive.

Table 7: Parameters for MLP neural network classifier.

Parameter	Explanation	Used value
L	Learning rate: used for weight adjustment on each iteration. (The value should be between 0 and 1.)	0.3
M	Momentum: used for weight adjustment during backpropagation, in order to speed up convergence and avoid local minima. (The value should be between 0 and 1.)	0.2
N	The number of epochs or passes through training data.	500
V	The percentage of the validation set from the training data.	20%
S	Seed for random number generator. Random numbers are used for setting initial weights for the connections between nodes. (The value should be ≥0.)	0
E	Threshold for consecutive errors allowed during validation testing before the neural network terminates. (The value should be >0.)	20
H	Number of nodes in the hidden layer which is represented as follows: number of hidden layers (number of neurons in each layer).	1 (11) 2 (11, 5) 3 (11, 5, 2)

An important parameter of ANNs is the used transfer function. In this study the most common activation (transfer) function which is the logistic sigmoid function was used. This function is also called log-sigmoid. The function is defined as

$$a = \frac{1}{1 + e^{-n}}. \quad (9)$$

The logistic sigmoid function accepts any value and returns a value between 0 and 1. Because of the nonlinear characteristics of this function, it allows ANNs to model complex data with possible built-in nonlinearities.

Table 7 shows the parameters and the values used in this paper for WEKA toolbox Multilayer Perceptron (MLP) ANN classifier configuration. MLP is the most popular ANN variation that allows configuration of multilayer ANN which is able to model complex relations between the input and output parameters.

Several ANN architectures were attempted in this paper, an ANN with one hidden layer and 11 neurons is used. Moreover, an ANN with two hidden layers with 11 neurons in the first layer and 5 neurons in the second hidden layer was used. Finally, ANN with three hidden layers with 11 neurons in the first layer, 5 neurons in the second hidden layer, and two neurons in the third hidden layer was also attempted.

By using the holdout method to train the ANN with one hidden layer, an overall classification accuracy of 97.5431% was achieved. This corresponds to correctly classifying 146184 out of 149865 in the testing set as can be noticed from Table 6.

Table 8 shows the Confusion matrix for this method. For example, there are 2656 records in the testing set for Scheduling attack as shown in Table 6. 2620 records were classified correctly as Scheduling attack, 23 records were classified as no-attack, 3 records were classified as Grayhole attacks, and 10 records were classified as Blackhole attack. This means that the percentage of positive classification of Scheduling attack is 98.6%. The percentage of samples that were incorrectly classified as positive while they are normal is 0.4%.

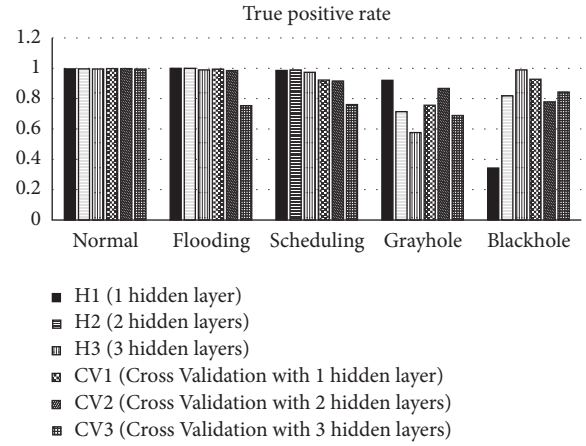

Figure 3: True positive results.

Table 9 shows the results of the remaining metrics for the holdout method. RMSE as calculated in (8) is 0.073 which is an acceptable value.

From Table 9, it can be concluded that the accuracy of detecting Blackhole attack was (34.3%), which is a low percentage. For that reason, an ANN architecture that has two hidden layers was attempted. In this case, 98% (avg. of TPR) of DoS cases were correctly classified with an error of 0.0817. Table 10 shows summary of the metrics of using this architecture. From Table 10, it can be shown that the accuracy rate decreased for Grayhole attack and significantly increased for Blackhole attacks.

When the ANN was trained on the dataset with three hidden layers, 97.8% of DoS cases were correctly classified with an error of 0.0791. Table 11 shows summary of results of using holdout method with three hidden layers.

More decrease in the accuracy rate of Grayhole attack can be seen in Table 11.

An ANN was trained on the WSN-DS dataset using 10-Fold Cross Validation method with one hidden layer. In this case, 98.52% of DoS attacks were correctly classified with an error of 0.0636. Table 12 shows the summary results of using this method with one hidden layer.

Table 12 shows an improvement in the results for all types of attacks. We have trained the ANN using 10-Fold Cross Validation with two hidden layers. Having two hidden layers, 98.53% of the DoS cases were classified correctly with an error of 0.0643. Table 13 summarizes the results of using this method.

Using 10-Fold Cross Validation to train an ANN architecture that has three hidden layers on WSN-DS dataset, 97.18% of the cases were correctly classified with an error of 0.0914. Table 14 summarizes the results of using this method.

Figures 3, 4, and 5 summarize the previous results. Figure 3 shows the True positive rate. On average the best

Table 8: Confusion matrix of holdout method with one hidden layer.

	Normal	Flooding	Scheduling	Grayhole	Blackhole
Normal	135483	350	32	152	10
Flooding	0	1325	0	0	0
Scheduling	23	0	2620	3	10
Grayhole	29	0	9	5379	421
Blackhole	0	0	3	2640	1377

Table 9: Summary results of holdout method with one hidden layer.

	TPR	FPR	FNR	TNR	*P*
Normal	0.996	0.004	0.004	0.996	1.000
Flooding	1.000	0.002	0	0.998	0.791
Scheduling	0.986	0.000	0.014	1	0.983
Grayhole	0.921	0.003	0.079	0.997	0.658
Blackhole	0.343	0.004	0.657	0.996	0.757
Avg.	*0.975*	*0.021*	*0.025*	*0.979*	*0.978*

Table 10: Summary results of holdout method with two hidden layers.

	TPR	FPR	FNR	TNR	*P*
Normal	0.996	0.008	0.004	0.992	0.999
Flooding	1	0.003	0	0.997	0.753
Scheduling	0.984	0	0.016	1	0.991
Grayhole	0.714	0.006	0.286	0.994	0.838
Blackhole	0.818	0.011	0.182	0.989	0.669
Avg.	*0.98*	*0.008*	*0.02*	*0.992*	*0.982*

Table 11: Summary results of holdout method with three hidden layers.

	TPR	FPR	FNR	TNR	*P*
Normal	0.995	0.016	0.005	0.984	0.998
Flooding	0.989	0.003	0.011	0.997	0.734
Scheduling	0.973	0.001	0.027	0.999	0.954
Grayhole	0.576	0.001	0.424	0.999	0.965
Blackhole	0.989	0.016	0.011	0.984	0.631
Avg.	*0.978*	*0.015*	*0.022*	*0.985*	*0.984*

Table 12: Summary results of 10-Fold Cross Validation with one hidden layer.

	TPR	FPR	FNR	TNR	*P*
Normal	0.998	0.018	0.002	0.982	0.998
Flooding	0.994	0.001	0.006	0.999	0.904
Scheduling	0.922	0	0.078	1	0.995
Grayhole	0.756	0.003	0.244	0.997	0.911
Blackhole	0.928	0.009	0.072	0.991	0.730
Avg.	*0.985*	*0.017*	*0.015*	*0.983*	*0.987*

method for classifying the attacks is Cross Validation with one hidden layer (CV1). It was the best in classifying all attacks except for Scheduling and Grayhole attack where it was slightly more accurate to use holdout method with one hidden layer (H1).

Table 13: Summary results of 10-Fold Cross Validation with two hidden layers.

	TPR	FPR	FNR	TNR	*P*
Normal	0.998	0.02	0.002	0.98	0.998
Flooding	0.985	0.001	0.015	0.999	0.900
Scheduling	0.915	0	0.085	1	0.992
Grayhole	0.867	0.007	0.133	0.993	0.832
Blackhole	0.778	0.005	0.222	0.995	0.810
Avg.	*0.985*	*0.019*	*0.015*	*0.981*	*0.985*

Table 14: Summary results of 10-Fold Cross Validation with three hidden layers.

	TPR	FPR	FNR	TNR	*P*
Normal	0.994	0.045	0.006	0.955	0.995
Flooding	0.754	0.001	0.246	0.999	0.855
Scheduling	0.761	0.001	0.239	0.999	0.946
Grayhole	0.689	0.01	0.311	0.99	0.743
Blackhole	0.843	0.013	0.157	0.987	0.638
Avg.	*0.972*	*0.041*	*0.028*	*0.959*	*0.974*

Figure 4 shows FPR. In FPR the smaller the rate, the better the performance. On average H1 was the best method; it is slightly better than CV1; however, CV1 was better than H1 in classifying Flooding, Scheduling, and Grayhole attacks. H1 was better in classifying the normal behavior and the Blackhole attack.

Figure 5 shows the error rate of all methods using Root Mean Squared Error (RMSE).

Figure 5 shows that CV1 was the best in terms of RMSE. From the results of TPR, FPR, and RMSE in Figures 3–5, it is concluded that the use of CV1 architecture outperforms other ANN architectures in classifying DoS attacks in WSN.

From the previous results obtained from applying ANN to WSN-DS dataset, high accuracy was achieved in the task of classifying four DoS attacks to determine whether the protocol is in its normal mode or exposed to any type of attack.

7. Conclusions and Future Work

The aim of this paper is to design an intelligent intrusion detection and prevention mechanism that could work efficiently to limit DoS attacks with reasonable cost in terms

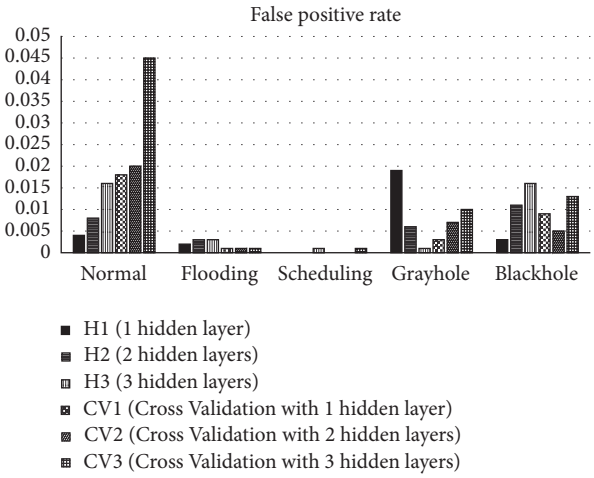

Figure 4: False positive results.

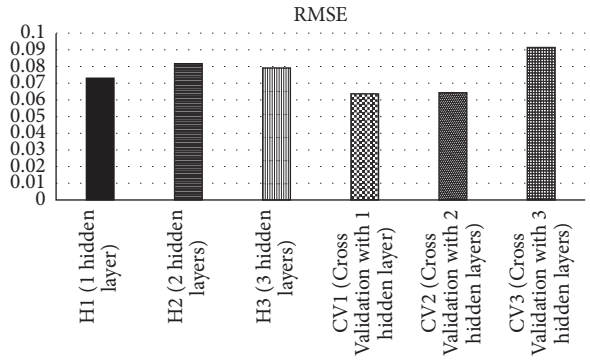

Figure 5: Root Mean Squared Error (RMSE) in each method.

of processing and energy. To achieve this aim, a specialized dataset for WSN was constructed to classify four types of DoS attacks. The considered attacks are Blackhole, Grayhole, Flooding, and Scheduling attacks. The data were collected using NS-2. In addition to including normal behavior, it was also able to collect 374661 records containing the signatures of these four attacks. The dataset containing normal and malicious network traffic was used to obtain the experimental results shown. In this paper, mathematical validation of the created dataset has been provided to ensure its correctness. The constructed dataset is called WSN-DS.

ANN-MLP model using WEKA toolbox was built; attacks were classified using two methods, holdout and 10-Fold Cross Validation, with one, two, and three hidden layers used in each case. We have found that, using 10-Fold Cross Validation with one hidden layer, the percentages of classification accuracies of attacks were 92.8%, 99.4%, 92.2%, 75.6%, and 99.8 in Blackhole, Flooding, Scheduling, and Grayhole attacks, in addition to the normal case (without attacks), respectively. From these results, it can be concluded that ANN trained using WSN-DS dataset is very useful in classifying DoS attacks as it was able to achieve high classification accuracy in the presence of more than one attack.

This work, which compares a number of distinct DoS attacking models, provides additional insights. Specifically, it would draw conclusions in terms of selecting the best protocol to be employed in a precisely predefined real-time application in WSN. This research reemphasizes the importance of considering security early in the network protocol development process. Without this, inherited vulnerabilities in these network protocols and other software will increasingly become targets for malicious attacks.

In future, this work can be extended to include other types of DoS attacks in data link layer such as Wormhole or Sybil. In addition, attacks on protocols other than LEACH and in different layers of WSN can be considered. It is also possible to attempt the use of other classifiers and data mining approaches. The current and future versions of WSN-DS will be posted to the researchers.

Competing Interests

The authors declare that they have no competing interests.

References

[1] N. Marriwala and P. Rathee, "An approach to increase the wireless sensor network lifetime," in *Proceedings of the World Congress on Information and Communication Technologies (WICT '12)*, pp. 495–499, IEEE, Trivandrum, India, October-November 2012.

[2] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.

[3] M. A. Rassam, M. A. Maarof, and A. Zainal, "A survey of intrusion detection schemes in wireless sensor networks," *American Journal of Applied Sciences*, vol. 9, no. 10, pp. 1636–1652, 2012.

[4] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[5] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSim '11)*, pp. 308–311, September 2011.

[6] J. Sen, "Security in wireless sensor networks," in *Wireless Sensor Networks: Current Status and Future Trends*, S. Khan, A.-S. K. Pathan, and N. A. Alrajeh, Eds., CRC Press, New York, NY, USA, 2012.

[7] N. Farooq, I. Zahoor, S. Mandal, and T. Gulzar, "Systematic analysis of DoS attacks in wireless sensor networks with wormhole injection," *International Journal of Information and Computation Technology*, vol. 4, no. 2, pp. 173–182, 2014.

[8] A. Mitrokotsa and T. Karygiannis, "Intrusion detection techniques in sensor networks," in *Wireless Sensor Network Security*, Cryptology and Information Security Series, pp. 251–272, IOS Press, 2008.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd IEEE Annual Hawaii International Conference on System Sciences*, pp. 1–10, Maui, Hawaii, USA, January 2000.

[10] H. Liu, L. Li, and S. Jin, "Cluster number variability problem in LEACH," in *Ubiquitous Intelligence and Computing*, pp. 429–437, Springer, Berlin, Germany, 2006.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[12] S. B. Alla, A. Ezzati, and A. Mohsen, "Hierarchical adaptive balanced routing protocol for energy efficiency in heterogeneous wireless sensor networks," in *Energy Efficiency—The Innovative Ways for Smart Energy, the Future Towards Modern Utilities*, InTech, 2012.

[13] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 623–645, 2013.

[14] A. Braman and G. R. Umapathi, "A comparative study on advances in LEACH Routing protocol for wireless sensor networks: a survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 2, pp. 5883–5890, 2014.

[15] H. Dhawan and S. Waraich, "A comparative study on LEACH routing protocol and its variants in wireless sensor networks: a survey," *International Journal of Computer Applications*, vol. 95, no. 8, pp. 21–27, 2014.

[16] D. Kumar, "Performance analysis of energy efficient clustering protocols for maximizing lifetime of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 4, no. 1, pp. 9–16, 2014.

[17] Y. M. Miao, "Cluster-head election algorithm for wireless sensor networks based on LEACH protocol," *Applied Mechanics and Materials*, vol. 738-739, pp. 19–22, 2015.

[18] S. Taneja, "An energy efficient approach using load distribution through LEACH-TLCH protocol," *Journal of Network Communications and Emerging Technologies (JNCET)*, vol. 5, no. 3, pp. 20–23, 2015.

[19] S. Peng, T. Wang, and C. P. Low, "Energy neutral clustering for energy harvesting wireless sensors networks," *Ad Hoc Networks*, vol. 28, pp. 1–16, 2015.

[20] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1-2, 2009.

[21] D. Mansouri, L. Mokdad, J. Ben-Othman, and M. Ioualalen, "Detecting DoS attacks in WSN based on clustering technique," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 2214–2219, IEEE, Shanghai, China, April 2013.

[22] A. Garofalo, C. Di Sarno, and V. Formicola, "Enhancing intrusion detection in wireless sensor networks through decision trees," in *Dependable Computing*, pp. 1–15, Springer, Berlin, Germany, 2013.

[23] S.-S. Wang, K.-Q. Yan, S.-C. Wang, and C.-W. Liu, "An integrated intrusion detection system for cluster-based wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15234–15243, 2011.

[24] D. Wu, G. Hu, and G. Ni, "Research and improve on secure routing protocols in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Circuits and Systems for Communications (ICCSC '08)*, pp. 853–856, IEEE, Shanghai, China, May 2008.

[25] G. Wang, J. Hao, J. Mab, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[26] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[27] R. Bace and P. Mell, *NIST Special Publication on Intrusion Detection Systems*, Booz-Allen and Hamilton, McLean, Va, USA, 2001.

[28] J. Xu, J. Wang, S. Xie, W. Chen, and J.-U. Kim, "Study on intrusion detection policy for wireless sensor networks," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 1–6, 2013.

[29] S. Khan and K.-K. Loo, "Real-time cross-layer design for a large-scale flood detection and attack trace-back mechanism in IEEE 802.11 wireless mesh networks," *Network Security*, vol. 2009, no. 5, pp. 9–16, 2009.

[30] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, pp. 1–7, 2013.

[31] A. Abid, A. Kachouri, and A. Mahfoudhi, "Anomaly detection in WSN: critical study with new vision," in *Proceedings of the International Conference on Automation, Control, Engineering and Computer Science (ACECS '14)*, pp. 37–46, 2014.

[32] H. Jadidoleslamy, "A high-level architecture for intrusion detection on heterogeneous wireless sensor networks: hierarchical, scalable and dynamic reconfigurable," *Wireless Sensor Network*, vol. 3, no. 7, pp. 241–261, 2011.

[33] KDD, https://kdd.ics.uci.edu.

[34] J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection," Results from the JAM Project by Salvatore, 2000.

[35] A. Ananthakumar, T. Ganediwal, and A. Kunte, "Intrusion detection system in wireless sensor networks: a review," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 12, pp. 131–139, 2015.

[36] A. Alsadhan and N. Khan, "A proposed optimized and efficient intrusion detection system for wireless sensor network," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 12, pp. 1621–1624, 2013.

[37] Y. El Mourabit, A. Bouirden, A. Toumanari, and N. E. Moussaid, "Intrusion detection techniques in wireless sensor network using data mining algorithms: comparative evaluation based on attacks detection," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 9, pp. 164–172, 2015.

[38] S. Sumitha Pandit and B. Kalpana, "Hybrid technique for detection of denial of service (DOS) attack in wireless sensor network," *International Journal of Advanced Networking and Applications*, vol. 7, no. 2, pp. 2674–2681, 2015.

[39] S. Athmani, D. E. Boubiche, and A. Bilami, "Hierarchical energy efficient intrusion detection system for black hole attacks in WSNs," in *Proceedings of the World Congress on Computer and Information Technology (WCCIT '13)*, pp. 1–5, IEEE, Sousse, Tunisia, June 2013.

[40] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[41] M. Tripathi, M. S. Gaur, and V. Laxmi, "Comparing the impact of black hole and gray hole attack on LEACH in WSN," *Procedia Computer Science*, vol. 19, pp. 1101–1107, 2013.

[42] A. P. Renold, R. Poongothai, and R. Parthasarathy, "Performance analysis of LEACH with gray hole attack in Wireless Sensor Networks," in *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI '12)*, pp. 1–4, January 2012.

[43] S. Magotra and K. Kumar, "Detection of HELLO flood attack on LEACH protocol," in *Proceedings of the 4th IEEE International Advance Computing Conference (IACC '14)*, pp. 193–198, IEEE, Gurgaon, India, February 2014.

[44] I. Almomani and B. Al-Kasasbeh, "Performance analysis of LEACH protocol under Denial of Service attacks," in *Proceedings of the 6th IEEE International Conference on Information and Communication Systems (ICICS '15)*, pp. 292–297, Amman, Jordan, April 2015.

[45] The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/.

[46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[47] R. R. Bouckaert, E. Frank, M. A. Hall et al., "WEKA—experiences with a Java open-source project," *The Journal of Machine Learning Research*, vol. 11, pp. 2533–2541, 2010.