J JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 977-1000 (2018)
DOI: 10.6688/JISE.201807_34(4).0011

Efficient Denial of Service Attacks Detection in Wireless Sensor Networks

IMAN ALMOMANI[1,2] AND MAMDOUH ALENEZI[1]
[1]*Computer Science Department*
Prince Sultan University
Riyadh, 11586 Saudi Arabia
E-mail: {imomani; malenezi}@psu.edu.sa
[2]*Computer Science Department*
The University of Jordan
Amman, 11942 Jordan
E-mail: {i.momani}@ju.edu.jo

Efficient detection of security attacks in Wireless Sensor Networks (WSNs) is very crucial in their Intrusion Detection Systems (IDSs). This is due to the sensitivity of WSN and its strong presence in the current and future Internet of Things (IoT) services. This paper provides a comprehensive empirical study aims at examining several Data mining techniques (DMTs) using a new specialized, published dataset for WSN networks (named as WSN-DS). The purpose is to provide an efficient IDS for detecting critical Denial of Service (DoS) attacks, which have a serious impact on the services provided by WSNs. Eight DMTs are considered in this study, which were attempted firstly using all existing features in WSN-DS, and evaluated in terms of detection accuracy and time complexity. Moreover, a feature selection algorithm has been applied to reduce around 53% of overall features while attaining a high accuracy rate reaches 98% and reducing the time complexity by up to 78.37% in some techniques. Thorough performance comparison among the studied DMTs before and after the feature selection is provided. Additionally, a deep security analysis has been conducted to make decisions regarding optimizing the attack detection process and consequently protecting WSN applications from different DoS attacks. Such decisions include the best way of integrating DMTs in the IDS of WSN.

Keywords: denial of service, DoS, data mining, wireless sensor networks, WSN, feature selection, intrusion detection systems, IDS, WSN-DS

1. INTRODUCTION

Growing attention is given to Wireless Sensor Network (WSN) due to its wide deployment in many military and civilian services. Such services include area monitoring, battlefields, environmental/earth sensing, manufacturing and industrial monitoring, healthcare, smart homes and smart cities, smart transportation and Internet of Things (IoT). Such important services will heighten the interest of security attackers in such networks and necessitate producing continuous security solutions. These solutions will preclude, detect and limit possible attacks, which could affect severely the provided services by these networks.

Having efficient secure solutions means considering the ongoing security strategies aim to prevent known attacks and detect emerging ones taking into account the charac-

Received August 30, 2017; revised September 30, 2017; accepted November 3, 2017.
Communicated by Changqiao Xu.

teristics of WSNs. The limited resources in WSNs including energy, processing unit, memory, and communication bandwidth in addition to the open-air nature of such networks (wireless) and the lack of centralization make them more vulnerable to security attacks [1, 2]. Especially, when they are running in an unattended environment. Building specialized IDS for WSNs is significant to offer security assurance for services provided by them. Several studies have been proposed by researchers to develop IDSs with high performance in detecting security attacks [3-9]. Data mining techniques (DMTs) are considered one of the powerful tools that could be integrated with IDSs to enhance their capabilities in detecting and classifying dangerous security attackers [10-16].

This paper conducts a comprehensive empirical study that investigates several classification techniques on a new published specialized dataset for WSN networks (named as WSN-DS). The aim is to provide an efficient IDS for detecting critical Denial of Service (DoS) attacks, which have a serious impact on the services provided by WSNs. The goal of this study is to experiment and find the best classification algorithm that can be used in such a scenario to be integrated with the IDS of WSN. The nature of the data and the type of the application usually determines which algorithm to choose. These selected algorithms (DMTs) are compared against well-known comparison measures including accuracy and complexity. The approach then applied on that with feature selection. The security impact of the approach is investigated and explained in this study.

This paper, firstly, surveys previous efforts in developing IDSs mainly in WSNs context and how DMTs are contributing to enhancing the performance of such IDSs. Then it presents the architecture of integrating data mining in the IDS of WSN. This architecture defines the main components of an IDS and how DTMs will be injected into its data analysis component to achieve fast, accurate detection of security attacks.

Dissimilar to previous studies, this research examines a wider spectrum of DMTs; Naive Bayes Classifier (NB), Decision Trees Classifier, Random Forests Classifier, Support Vector Machine (SMO), J48, Artificial Neural Networks, *K*-Nearest Neighbor and Bayesian Networks. Moreover, these eight DMTs are attempted using a recent, specialized dataset for WSN called (WSN-DS). The complete dataset with all its features and instances are experimented. WSN-DS is targeting mainly Denial of Service (DoS) attacks. Four of them are addressed in this paper; Flooding, Scheduling, Grayhole and Blackhole attacks, in addition to the normal behavior. The performance of these DMTs is measured by detection accuracy and time complexity. A detailed comparison of their performances shows high detection rates but with large discrepancies in terms of complexity.

The development of IDS for WSN should always take into account the limited resources of such networks. Therefore, this paper has considered applying a feature selection algorithm to reduce the number of features without affecting the average detection rate. The algorithm succeeded to select only 9 features out of 19, which significantly has reduced the complexity of building DMT models. Many experiments have been conducted to test the performance of DMTs before and after feature selection.

Security analysis has been provided to discuss the results and how security and complexity could be traded off to offer efficient IDS for WSN. This IDS should provide high detection rate of security attacks while taking WSN's characteristics into account. This analysis also elaborates the impact of feeding the IDS with proper DMT to guarantee instant detection and possible exclusion of dangerous attacks from early rounds of network lifetime. This accordingly will prolong the efficient provisioning of services provided by WSN applications.

The rest of paper is organized as follows: Section 2 provides a classified survey of recent existed IDSs utilizing DMTs. Section 3 presents the integration of data mining into WSN's IDS, dataset description and list of DMTs under study. Experiments and results are displayed and discussed in section 4. This also includes the feature selection algorithm and its results in addition to a security analysis part. Finally, the paper is concluded in section 5, where avenues for future work are also presented.

2. RELATED WORK

Many IDSs have been proposed in the literature. This section highlights existing IDSs which have adopted different DMTs in their systems to detect and classify security attacks in general and WSN attacks in particular.

None of the surveyed papers used a mathematically proven specialized WSN dataset. Dhanalakshmi and Kannapiran [4] tried to improve the detection performance of intrusion detection system. They proposed a density-based clustering IDS. The approach with the name of Adaptive Rule-Based Multiagent Intrusion Detection System (ARMA-IDS) uses the combination of distance measurement and Fuzzy c-Means (FCM)). They compared ARMA-IDS with Random forest in terms of Error rate, Recall, and False Detection. They also compared ARMA-IDS and traditional random forest, JRip, AdaBoost, and common path mining algorithms in terms of accuracy, precision, recall, *F*-measure, and the number of classes. The authors adopted unsupervised data mining technique whereas our work uses supervised data mining techniques. Luigi *et al.* [12] proposed a hybrid, lightweight, distributed Intrusion Detection System (IDS) for wireless sensor networks that used both misuse-based and anomaly-based detection techniques. They applied Decision trees in the detection process. They used the FITNESS Research Group dataset, "Cyber security datasets for wireless sensor networks."

Chakchai *et al.* [11] employed different classification algorithms namely, Decision Tree, Ripper Rule, Neural Networks, Naïve Bayes, *k*-Nearest-Neighbour, and Support Vector Machine (SVM) for intrusion detection analysis. The authors used both KDD CUP dataset and HTTP BOTNET attacks. Their results, in general, showed that *k*-Nearest-Neighbour is one of the best candidates considering the lowest computational complexity with good classification accuracy. P. Amudha *et al.* [13] employed data mining techniques for intrusion detection. They evaluated the performance of classification algorithms namely J48, Naïve Bayes, NBTree and Random Forest using KDD CUP'99 dataset. Their results showed that Random Forest outperforms other algorithms in terms of predictive accuracy and detection rate. Yousef *et al.* [16] studied the KDD'99 dataset for improving the results using Data Mining techniques. They normalized the dataset then ran feature selection algorithms. Then, they employed K-means, Naïve-Bayes, SVM and Random Forest algorithms for classification purposes. Their experimental results showed that the random forest methods provide high detection rate and reduce false alarm rate. The work of [3, 10, 19] used the KDD CUP dataset, which is not specific to WSN whereas our work is investigating a specialized WSN dataset.

Table 1 summarizes, classifies and compares the surveyed IDSs in terms of used datasets, applied DMTs, addressed attacks, evaluation metrics, and simulation environment. These IDSs are either proposed or reviewed for the purpose of comparison and analysis.

Table 1. Comparison of existing DMT-based IDSs.

No	Datasets	Data Mining Technique	Attack Model	Evaluation Metrics	Simulation Environment
Dhanalakshmi and Kannapiran, 2016 [4]	KDDCup99 SCADA	Authors proposed a density-based clustering IDS (**ARMA-IDS)** which uses the combination of distance measurement and Fuzzy c-Means (FCM))	**KDD attacks:** DoS, Probe, U2R, R2L **SCADA:** Context-specific attackers Including DoS	(1) compare **ARMA-IDS** with **Random forest** in terms of: **Error rate & Recall & False Detection** (2) compare **ARMA-IDS** and traditional **random forest, JRip, AdaBoost**, and **common path mining** algorithms in terms of **accuracy**, **precision, recall**, ***F*-measur**e, and the **number of classes** (3) compare between **KDD** and **SCADA** in terms of **Detection rate**	Ns-2
Chakchai *et al.*, 2014 [11]	KDDCup99 Extended with HTTP botnet attacks	Decision Tree, Ripper Rule, Neural Networks, Naïve Bayes, *k*–Nearest–Neighbour, and Support Vector Machine (SVM)	**KDD attacks:** DoS, Probe U2R, R2L	classification accuracy and complexity	WEKA
P. Amudha *et al.*, 2013 [19]	KDDCup99	Naïve Bayes, Decision Trees Support Vector Machine, In addition to Ensemble Classifier approach which considers more than model	**KDD attacks:** DoS, Probe U2R, R2L	Predictive accuracy: TP, FN, FP, TN Receiver Operating Characteristics (ROC)	Review Study
Abhaya *et al.*, 2014 [10]	KDDCup99 "not directly stated"	Decision Tree, K-Nearest Neighbor, Naive Bayes classifier Support Vector Machine	**KDD attacks:** DoS, Probe U2R, R2L	Execution time Accuracy rate (TP, TN, FP, FN)	Review Study
Chih-Fong *et al.*, 2009 [20]	KDD99 DARPA1998 DARPA1999 UNM, CNNJU CUCS, RWND PACCT, Windows System Network, TCP dump data	**Single Classifier** Support vector machines, Artificial neural networks, Self-organizing maps, Decision trees Naïve Bayes networks, Genetic algorithms, Fuzzy logic **Hybrid Classifier** **Ensemble Classifier**	Not mentioned	Year-wise distribution of articles for the types of classifier design/ single classifiers/hybrid classifiers/baseline classifiers/datasets used/feature selection considered.	Review Study
P Amudha *et al.*, 2011 [13]	KDDCup99	J48, Naïve Bayes, NBTree Random Forest	**KDD attacks:** DoS, Probe U2R, R2L	Predictive accuracy, detection rate, and time complexity	WEKA
Luigi *et al.*, 2013 [12]	FITNESS Research Group, "Cyber security datasets for wireless sensor networks	Decision Tree, Classification And Regression Tree (CART) CHi-squared, Automatic Interaction Detection (CHAID) C5.0, Logistic Regression, Bayesian Network	Sinkhole sleep deprivation	FPR (False Positive Rate) FNR (False Negative Rate) AC (Accuracy Detection)	NS-3
Sonu and Padmavati, 2016 [9]	Not mentioned	Not mentioned	General attacks, No linking to IDSs or DMTs	Compare DMT based IDS including ones based on in terms of communication overhead, scalability, detection speed, false alarms and computational complexity	Review Study
Anush *et al.*, 2015 [3]	KDDCup99 DARPA,1999	Support Vector Mechanism (SVM), Bayesian classifier program	**KDD attacks** & DoS attacks	Detection accuracy and False detection rate	Review Study
Yousef *et al.* 2015 [16]	KDD'99 with more selected features.	K-means, Naïve-Bayes, SVM, Random Forest	**KDD attacks:** DoS, Probe, U2R, R2L	Confusion Matrix, Classification Rate	WEKA

As can be concluded from the table, Knowledge Discovery and Data Mining (KDD) dataset, KDD attacks, classification accuracy and complexity metrics, and WEKA tool are the mostly used by the developed IDSs among others. Many techniques other than DMTs have been also considered while designing IDSs for WSN. For example in [7], out of 46 reviewed IDSs, only two have adopted DMTs in their solutions.

Therefore, this paper attempts to provide a profound study of the impact of applying different DMTs on the performance of IDSs. This is in the context of WSN being targeted especially from DoS attacks. In order to achieve more realistic results, a specialized dataset for WSN needs to be used to run the DMTs under study. KDD dataset [17] was constructed for Local Area Network (LAN). KDD is not specialized for wireless networks in general and WSN in particular, even though it has been considered by many researchers in the scope of IDSs as shown in Table 1 and also presented in [8, 14, 16, 18].

The following sections present the architecture of the proposed IDS for WSN and how DMTs are injected, examined, evaluated and compared. Consequently, draw recommendations for optimizing the detection process of security attacks in WSN.

3. THE INTEGRATION OF DATA MINING IN THE IDS OF WSN

In order to build an efficient IDS for WSN, the normal behavior of the network should be studied carefully to be able to identify any anomalous activities that might occur. Moreover, attacks' signatures will be also defined to characterize the coexisted attacks. To achieve that, several components need to be included in the IDS. Fig. 1 shows main components in such systems starting from having monitoring services to collect data about each sensor in the network. This could be implemented by running a Monitor Agent (MA) at each sensor regardless of its role in the network. This agent monitors group of neighbor sensors especially their packets' transmission and reception. The collected data by these monitors will be sent to the sink to be processed and analyzed at either the sink or the Admin side, which could be approached through the Internet network.

Detection and possible classification of security attacks are the key results of this analysis. Here DMTs could be injected to ensure high detection rate of security attacks.

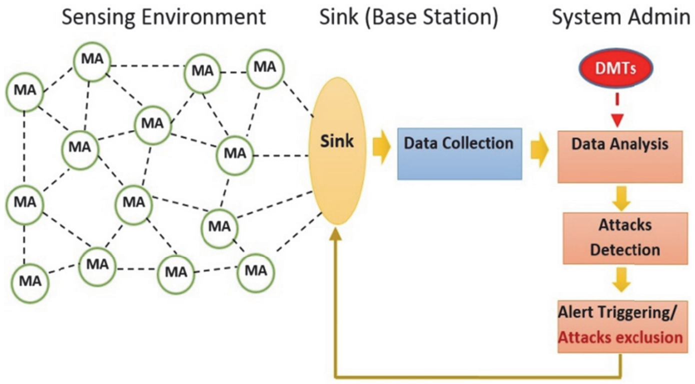

Fig. 1. DMTs based intrusion detection system for WSN.

Then the system admin has to alert sensor nodes through the sink of the existence of such attacks and also request to exclude attacking nodes from the next round of WSN lifetime.

In the following subsections, the collected dataset is described. Also, the selected DMTs, which take place in the analysis phase and contribute to detecting different DoS attacks are presented. Furthermore, the performance results of implementing these DMTs according to a set of evaluation metrics are analyzed. Finally, the impact of excluding the detected attacks is discussed.

3.1 Dataset Description

This paper is utilizing a specialized dataset for WSN (WSN-DS), which has been evaluated and published recently in [21]. WSN-DS was constructed to characterize the normal behavior and four types of DoS attacks when Low Energy Aware Cluster Hierarchy (LEACH) is running [22]. LEACH has been chosen since it is one of the most popular, simple and hierarchical routing protocol in WSNs that consumes limited energy. Many protocols are developed based on LEACH [23-29]. The energy balancing in LEACH protocol is achieved by organizing the nodes into clusters. In each cluster, there is a node called Cluster Head (CH), which aggregates the data received from sensors (Cluster Members) within its cluster and forward them to the Base Station (BS). The purpose of clustering is to limit the direct communication with the sink (BS), which could be located far from the sensors, especially in large-scale WSN. This, in turn, will save the sensors' energy and consequently prolong the network lifetime and its provided services.

LEACH protocol has many rounds depending on the network lifetime. Each round in LEACH protocol consists mainly of two phases: a setup phase and steady-state phase. In the setup phase, clusters are formed, whereas in the steady-state phase, sensed data will be forwarded to the sink node. The cluster heads are not fixed throughout the network lifetime; they are changed at each round to avoid overwhelming specific nodes and disseminate the network load among all sensors.

The vital role of CH in controlling the sensed data transmission and delivery makes it the main target to security attackers. With the absence of security countermeasures in the original LEACH, a malicious node (MN) can easily get the role of CH. Fig. 2 shows the structure of nodes in LEACH routing protocol with the existence of attacks.

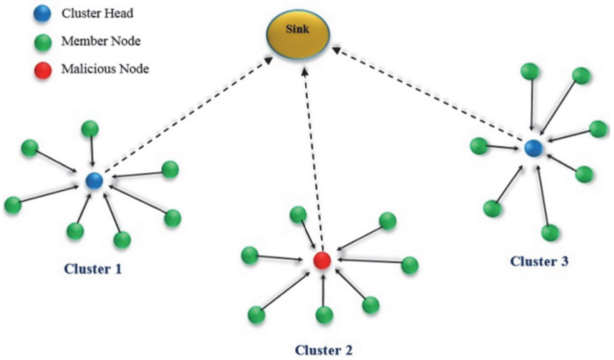

Fig. 2. Nodes structure in LEACH routing protocol with the existence of attacks.

The MN can perform different DoS attacks in LEACH protocol. Four types of them are addressed in this study; Blackhole, Grayhole, Flooding, and Scheduling attacks. The behaviors of these attacks are described in the following algorithm and mathematically analyzed in Theorems 1-4 to illustrate their impacts on LEACH protocol phases. Table 2 lists the notations used in the algorithm and the theorems' equations and their meanings.

Table 2. Equations and algorithm's notations and their meanings.

Notation	Meaning
N	Network Size
SN	Sensor Node
CH	Cluster Head
MN	Malicious Node (attacker)
BS	Base Station
NC	Number of Cluster Heads (CHs) within a specific round
P	Percentage of attackers (MN) acting as CHs
NC'	*P* * *NC*
CM	Number of Members (sensors) in a specific Cluster
CNum	Cluster Head numbers
SP	Percentage of packets that are dropped selectively
T	Threshold function
X	Integer value between 0 and $N-1$
NFlood	Number of advertisement packets an attacker will send to flood the network during a specific round
NO-DATA-PKT	Number of data packets received by a *CH*
JOIN-REQ-SENT	Number of join request messages sent by sensor nodes
JOIN-REQ-RCVD	Number of join request messages received by CHs

```
$\forall SN_i$, $0 < i \le N$, compute $T(SN_i)$ and random $r_{SN_i}$
IF ($r_{SN_i} < T(SN_i)$) THEN
        $\mathbf{SN_i = CH}$
ELSE
        $\mathbf{SN_i = CM}$
ENDIF
$\forall CH_j, j \in CNum$
{
  Step 1: $CH_j$ broadcasts the advertisement message (ADV_CH)
  Step 2: $x$ $CM_s$ join $CH_j$
  Step 3: $CH_j$ creates TDMA schedule
  Step 4: $x$ $CM_s$ send data to $CH_j$ during assigned TDMA time slot
  Step 5: $CH_j$ sends aggregated data to BS
}

// In case of compromising CH, the functionalities of these steps will be changed as follows:
IF $CH_j$ = Blackhole THEN
        Step 5: $CH_j$ drops all received packets
ELSE IF $CH_j$ = Grayhole THEN
        Step 5: $CH_j$ drops some of the received packets (randomly or selectively)
ELSE IF $CH_j$ = Flooding THEN
        Step 1: $CH_j$ broadcasts excessive number of advertisement messages with high transmitting
                power to drain sensors' energy
ELSE IF $CH_j$ = Scheduling THEN
      Step 4: $CH_j$ creates the TDMA schedule and assigns all nodes the same time slot to send their
                data to the Cluster Head which causes data collision
End IF
```

Theorem 1: Blackhole attack costs the network $\sum_{i=1}^{NC'}\frac{NO-DATA-PKT}{CMofCHi}$ of its data packet within a specific round and prevents them from reaching the sink (BS).

Proof**:** In reference to LEACH, once the CH receives the sensed data from the sensors nodes (NO-DATA-PKT) according to their time slots assigned by TDMA schedule, it aggregates them into one packet and sends it to the BS ($\frac{NO-DATA-PKT}{CMofCHi}$). Having NC of CHs, and knowing the percentage of MN; P, then the overall data packets received and dropped by the malicious CHs and prevented from reaching the BS are $\sum_{i=1}^{NC'}\frac{NO-DATA-PKT}{CMofCHi}$.

Theorem 2: Grayhole attack costs the network $SP*\sum_{i=1}^{NC'}\frac{NO-DATA-PKT}{CMofCHi}$ of its data packet within a specific round and prevents them from reaching the sink (BS).

Proof**:** In reference to LEACH, once the CH receives the sensed data from the sensors nodes (NO-DATA-PKT) according to their time slots assigned by TDMA schedule, it aggregates them into one packet and sends it to the BS ($\frac{NO-DATA-PKT}{CMofCHi}$). Having NC of CHs, and knowing the percentages of MN; P, and number of packets that are dropped selectively by malicious CHs; SP, then the overall data packets received and dropped by these malicious CHs and prevented from reaching the BS are $SP*\sum_{i=1}^{NC'}\frac{NO-DATA-PKT}{CMofCHi}$.

Theorem 3: Flooding attack in the advertisement phase and within a specific round overwhelms the network with a maximum $(NC - NC') + NC'*NFlood$ of ADV-CH-SENT packets and a maximum $(N - 1)[(NC - NC') + (NC'*NFlood)]$ of ADV-CH-RCVD packets.

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 to NC. But having NC' of attacks out of NC, normal nodes will send $(NC - NC')$ADV-CH-SENT whereas MN will send $NC'*NFlood$ means $(NC - NC') + NC'*NFlood$ of ADV-CH-SENT in total. On the other hand, these advertisement messages will be received by all sensor nodes (N) except the CH node itself which equals to $(N - 1)*(NC - NC') + (N - 1)*NC'*NFlood$, that equivalents to $(N - 1)[(NC - NC') + (NC'*NFlood)]$.

Theorem 4: Scheduling attack costs the network $\sum_{i=1}^{NC'} NO-DATA-PKT$ of its sensors' data packets within a specific round due to data collision.

Proof**:** According to LEACH, sensor nodes' within a cluster are assigned specific time slots by their associated CHs to send their data packets (NO-DATA-PKT). Knowing the percentage of malicious CHs; P, which usually assigns same time slots to all sensors' within their clusters and causes data collision to all data packets received by them; in total: $\sum_{i=1}^{NC'} NO-DATA-PKT\cdot$

A scheme has been defined in [21] to collect data from Network Simulator-2 (NS-2) and then process them to produce 23 features after implementing both the normal and DoS attacks scenarios. These features are listed in Table 3.

Table 3. WSN-DS features.

Features	Description
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 at 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	Number of Advertise CH's broadcast messages sent to the nodes.
ADV_CH receives	Number of Advertise CH messages received from CHs
Join_REQ send	Number of Join request messages sent by the nodes to the CH.
Join_REQ receive	Number of Join request messages received by the CH from the nodes.
ADV_SCH send	Number of Advertise TDMA schedule broadcast message sent to the nodes.
ADV_SCH receives	Number of TDMA schedule messages received from CHs.
Rank	The order of this node within the TDMA schedule.
Data sent	Number of data packets sent from a sensor to its CH.
Data received	Number of data packets received from CH.
Data sent to BS	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 (Class label)	Type of the node. It is a class of five possible values, namely: Blackhole, Grayhole, Flooding, Scheduling, in addition to Normal, if the node is not an attacker.

3.2 Data Mining Techniques (DMTs)

Laskov *et al.* [30] found in their experiments that supervised learning methods (classification) considerably outperform unsupervised learning (clustering) when the test data contains known attacks as the case in this study. Classification is one of the mostly used machine learning techniques [31] with a broad range of applications, including sentiment analysis, risk assessment, spam detection, medical diagnosis, intrusion detection, and image classification. It is also known as supervised statistical learning. In supervised learning, the model needs to be first trained using data with predetermined classes. This data is used to train the learning algorithm, which creates a model that can then be used to label/classify the testing instances, where the values of the class labels are unknown. Eight techniques have been considered in this paper including Artificial Neural Network (ANN) which has been used in [21] to examine the constructed WSN-DS. These techniques are [32]:

- **Naive Bayes (NB)** classifier is based on Bayes theorem in which it assumes that all features are independent. NB discovers the class with maximum probability given a set of features values using the Bayes theorem. One advantage of NB is the fact it needs an only small portion of training data to estimate the classification parameters.

Naive Bayes classifier selects the classification Θ_t that is most likely for the features $f_1, f_2, f_3, \ldots, f_n$ such as shown in Eq. (1).

$$\Theta_t = \arg\max_{\Theta_j \in \Theta} P(\Theta_j) \prod P(f_i \mid \Theta_j) \tag{1}$$

Where $P(f_i|\Theta_j)$ is an estimate using κ estimates such as shown in Eq. (2).

$$P(f_i \mid \Theta_j) = \frac{e_s + \kappa\rho}{e + \kappa} \tag{2}$$

Where as

- ♦ κ is the total sample size
- ♦ e_s the total sample that have $f = f_i$ and $\Theta = \Theta_j$
- ♦ e is total number of sample where $\Theta = \Theta_j$
- ♦ ρ is priori estimator for $P(f_i|\Theta_j)$

• **Decision Trees (DT)** classifier uses a tree-like structure to represent the attributes and possible outcomes. It decides the dependent value of a new sample based on diverse attribute values of the existing data. Each internal node in the tree represents a single attribute while leaf nodes represent class labels. Decision trees classify each instance by starting at the root of the tree and moving through it until a leaf node.

For decision tree after looking at impurity two things are measured (a) Entropy and (b) Information Gain

Entropy E: is a degree about the randomness of elements *el* or it is usually referred as the degree of impurity that is described in Eq. (3).

$$\mathrm{E} = -\rho(el)*\log\rho(el) \tag{3}$$

Where ρ is referred as probability of element and entropy is product of ρ and logρ.

Overall entropy is calculated by summing all entropies as E calculated in Eq. (4).

$$\mathrm{E} = \sum_{i}^{n} -\rho(el_i) * \log \rho(el_i) \tag{4}$$

Information Gain (G): is based on entropy whenever the sets are split according to attributes. Attribute with higher G is a task of building a decision tree. That means higher the G of the attribute refers to higher contribution of decision making. G is calculated in Eq. (5).

$$G(P, c) = E(P) - E(P, c) \tag{5}$$

Where

- ♦ E(P) is entropy of parent P.
- ♦ E(P, c) is weighted sum entropy of child

• **Random Forests** Classifier is an ensemble classification and regression algorithm. It generates several decision trees at training time. Each tree gives a class label. The Random Forests classifier selects the class label that has the mode of the classes output by individual trees. Random forests algorithm has been used extensively in different applications. It is made by making simple trees that have numeric values. The set of predictor is set randomly and root-mean-square ξ is calculated using Eq. (6)

$$\xi = (O_i - \Gamma). \tag{6}$$

Where O_i is observation and Γ are tree responses. The prediction ρ_ϱ for the random forest is defined in Eq. (7).

$$\rho_\varrho = \frac{1}{k}\sum_{j=1}^{k}\Gamma_j \tag{7}$$

Where ρ_ϱ is predication based on random forest and Γ_j is jth tree-response and are randomly distributed samples while k is a total number of runs.

- **Support Vector Machine (SVM)** is one of the most widely used data mining classification techniques. It works with feature space instead of the data space. It finds the optimal hyper-plane, which maximally separates samples in two different classes. SVM represents the examples as points in the space to divide them by a clear gap so the new examples can be mapped into the same predicated category. The Sequential Minimal Optimization (SMO) algorithm is a fast simple method for training an SVM. SVM works on hyperplane represented using Bias ϑ and weighted vector ω. Distances on vector are calculated using following Eq. (8)

$$d = \frac{|\vartheta_0 + \vartheta^T N|}{||\vartheta||}. \tag{8}$$

Where N represents training examples while ϑ is known as the weighted vector. ϑ_0 is referred as bias. ϑ_0 is a trainings sample weighted vector to be taken from N.

- **J48** uses a divide and conquer approach to growing decision trees. It forms a tree structure and decides the dependent value of a new sample based on diverse attribute values of the existing data. J48 work on Gain, Entropy, and pruning Entropy E and Information Gain G are already defined in the decision tree, while pruning is calculated through finding the error rate ς and error rate ς is greater than parent then pruning is done else splitting is done so ς will be calculated of error rates and the total number of samples in Eq. (9).

$$\varsigma = \frac{\gamma + \frac{\xi^2}{2N} + \sqrt[\xi]{\frac{\gamma}{N} + \frac{\gamma^2}{N} + \frac{\xi^2}{4N^2}}}{1 + \frac{\xi^2}{N}} \tag{9}$$

whereas γ is referred as error rate and N are the total number of samples while ξ will be calculated through confidence level as shown in Eq. (10).

$$\xi = \phi^{-1}(\mathfrak{h}) \tag{10}$$

where $\mathfrak{h}$ is referred as the confidence level.

- **Artificial Neural Networks (ANN)** is inspired by the neurons in human brain system. It makes a structure or a network of numerous interconnected units (artificial neurons). Each one comprises input/output characteristics that implement a local function. The function can be computing weighted sums of inputs, which produces an output if it exceeds a given threshold. The function output could serve as an input to other neurons in the network. This requires multiple iterations before reaching a final answer. ANN works on propagation function represented by $\mathbb{P}$ based on neurons $\mathbb{N}$ as shown in following Eq. (11).

$$\mathbb{P}_{\mathbb{N}}(T) = \sum_j \mathcal{O}_j(T)\omega_{\mathbb{N}_j} \tag{11}$$

Where as $\omega_{\mathbb{N}j}$ are the weight for each Neuron $\mathbb{N}$ and its predecessor j. $\mathcal{O}_j$ is an output function based on activation a at some Time T as shown in following Eq. (12).

$$\mathcal{O}_j(T) = \Xi(a_{\mathbb{N}}(T)) \tag{12}$$

Where are Ξ is an output function that computes the activation a at some time T.

- ***K*-Nearest Neighbour (KNN)** classifies instances based on their similarity of their neighbors. The neighbors are selected from a set of objects for which the correct classification is known. The training instances are defined by n-dimensional numeric attributes. Each instance represents a point in this n-dimensional space. All of the training instances are stored in an n-dimensional pattern space. When given an unknown instance, a k-nearest neighbor classifier searches the pattern space for the k training instances that are closest to the unknown instance. KNN is based on weighted distance $\mathbb{W}$ that is calculated using a query point q along with instance c as shown in following Eq. (13).

$$\mathbb{W}(q, c_i) = \frac{\exp(-D(q, c))}{\sum_{i=1}^{n} \exp(-D(q, c_i))} \tag{13}$$

And k-nearest prediction outcome ψ is calculated using following Eq. (14),

$$\psi = \frac{1}{K}\sum_{i=1}^{n} \psi_i \tag{14}$$

where ψ_i is the ith mark in the sample of size n. The final equation of only real-valued results will be as shown in Eq. (15).

$$\mathbb{K}(q_i) = \frac{\sum_{i=1}^{n} \mathbb{W}_i c_i}{\sum_{i=1}^{n} \mathbb{W}_i} \tag{15}$$

- **Bayesian Networks** represents probabilistic graphical models, which allow the representation of dependencies among subsets of attributes. These networks depict conditional probability distributions allowing class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. Probabilistic inference can be performed to predict the outcome of some variables based on the observations of others. The goal is to find posterior probability conditional cost $\mathcal{C}$ that is given as follows in Eq. (16).

$$\mathcal{C}[\text{э} \mid \gamma] = \mathcal{C}[\gamma \mid \text{э}] \cdot \frac{\mathcal{C}[\text{э}]}{\mathcal{C}[\gamma]} \tag{16}$$

Where γ is the evidence either observer or unobserved while э is the reason for which evidence γ is required.

Bayesian network follows Bayesian theorem that is stated below in Eq. (17).

$$\rho(\text{э} \mid \gamma) = \frac{\rho(\gamma \mid \text{э}) * \rho(\text{э})}{\rho(\gamma \mid \text{э}) * \rho(\text{э}) + \rho(\gamma \mid \text{э}') * \rho(\text{э}')} \tag{17}$$

Where as э′ is the posterior probability and ρ is probability based on reason э and evidence γ.

These DMTs are applied to WSN-DS and evaluated considering all features and also selected features. The evaluation results of such application are discussed in the following sections.

4. EXPERIMENTS AND RESULTS

In order to test the introduced DMT-based IDS for WSN, WEKA tool has been used. Table 4 illustrates the software and the hardware specifications used to build our experiments. In addition, the experiments were performed using the complete WSN-DS dataset, which contains 374661 records and 19 features including class type.

Table 4. Software and Hardware specifications.

Software and Hardware Information	
Weka Version	3.8.0
java.class.version	52.0
java.runtime.name	Java(TM) SE Runtime Environment
java.runtime.version	1.8.0_112-b15
java.specification.name	Java Platform API Specification
java.specification.vendor	Oracle Corporation
java.specification.version	1.8
memory.initial	128MB (134217728)
memory.max	3577MB (3750756352)
os.name	Windows 10
os.version	10.0

Processor Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz, 3601 Mhz, 4 Core(s), 8 Logical Processor(s)
BIOS Version/Date Hewlett-Packard L01 v02.57, 12/11/2014
SMBIOS Version 2.7
Embedded Controller Version 255.255
Installed Physical Memory (RAM) 8.00 GB with SSD

10-fold cross-validation has been used in the conducted experiments. This is because it outperformed holdout method when ANN was applied in [21]. Moreover, different metrics are considered to evaluate the performance of the DMTs including:

- True Positive (**TP**) Rate: the rate of instances correctly classified as a given class. TP is calculated in Eq. (18):

$$TP/(TP + FN) \tag{18}$$

where, *FN* (False Negative) indicates the number of instances in a specific class, which is classified incorrectly as other class types.

- False Positives (**FP**) Rate: the rate of instances falsely classified as a given class. FP is calculated in Eq. (19):

$$FP/(TP+TN) \tag{19}$$

where *TN* (True Negative) represents all instances correctly classified as not the given class.

- Receiver Operating Characteristics (**ROC**) – also known as Area under curve (AUC) – measures the performance of a binary classification. It is the preferred measure to use in comparing classification algorithms.
- Time Complexity (**TC**): Time is taken to build the model.

4.1 Results

As mentioned before, WSN-DS is a newly published specialized dataset for WSN. The only found work utilizing this dataset is presented in [21]. The authors attempted only ANN to detect and classify DoS attacks that might exist in WSNs. The majority of the existed related works considered KDD dataset as shown previously in Table 1 and explained thoroughly in the related work section.

This research contributes in examining eight DMTs including ANN and tests their performances in detecting four different DoS attacks, in addition to the normal behavior, using specialized dataset for WSN. Detailed comparative analysis of this related work and the rest of DMTs in terms of detection accuracy (TP and FP), ROC and time complexity is presented in this section. The main outcome of this empirical study is to find the most appropriate DMT(s) to be integrated within the network's intrusion detection system while meeting the network requirements. Also, optimizing provided services by ensuring efficient detection of critical DoS attacks.

Table 5 shows the DMTs performance in terms of TP against packet dropping attacks. These are the types of DoS attacks that aim to drop the user's packets whether completely (Blackhole), selectively (Grayhole) or by causing packets' collision (TDMA). Whereas Table 6 shows the TP results in case of flooding attacks and in case of normal behaviors of the network protocols.

Table 5. DMTs performance in terms of True Positive (TP) against packet dropping attacks.

Packet Dropping attacks	Data Mining Technique (DMT)							
	ANN [21]	NB	BayesianNet	DecisionT	J48	Random Forest	SMO	KNN
Blackhole	0.659	0.993	0.83	0.976	0.993	0.997	0.955	0.992
Grayhole	0.905	0.589	0.911	0.9	0.982	0.99	0.501	0.981
TDMA	0.925	0.755	0.933	0.875	0.927	0.928	0.862	0.92

Table 6. DMTs performance in terms of True Positive against Flooding attacks in addition to the normal behavior.

Flooding attack / Normal case	Data mining Technique (DMT)							
	ANN [21]	NB	BayesianNet	DecisionT	J48	Random Forest	SMO	KNN
Flooding	0.997	1	0.998	0.898	0.975	0.989	0.941	0.922
Normal	0.998	0.971	0.973	0.998	0.999	0.999	0.994	0.997

In comparison to the results presented in [21] where ANN was only applied, many DMTs have achieved better detection rate for many attackers. For example, all DMTs have outperformed ANN in detecting Blackhole attacks. But, Random forest was the best among them. BayesianNet, J48, and Random Forest have also performed better in detecting Grayhole attacks. Better detection for TDMA attacks was observed in BayesianNet, J48, and Random forest.

In terms of Flooding attacks, NB, BayesianNet and Random Forest outperform the ANN. High accuracy with more than 99% in detecting normal instances was observed by the majority of the algorithms except for NB and BayesianNet where they reach 97% accuracy. Fig. 3 shows the TP results for all studied techniques including the related work and the normal behavior of the network.

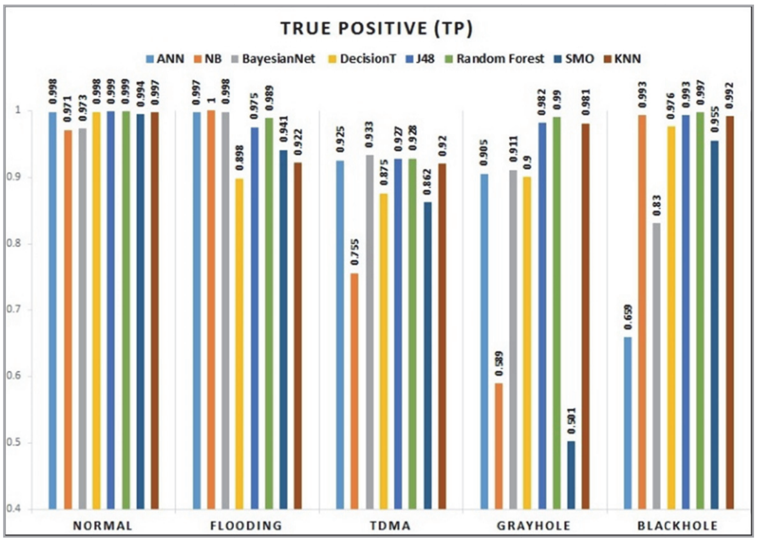

Fig. 3. TP of normal and DoS attacks in all DMTs.

Overall, Random forest was the best among others with close results to J48 and KNN as shown in Fig. 4. The percentage of enhancement in detecting some attacks has reached 34% in comparison to ANN and 49% compared to other DMTs like SMO.

Whereas, the worst in detecting Flooding, TDMA, Grayhole and Blackhole DoS attacks were: Decision Tree, NB, SMO, and ANN respectively.

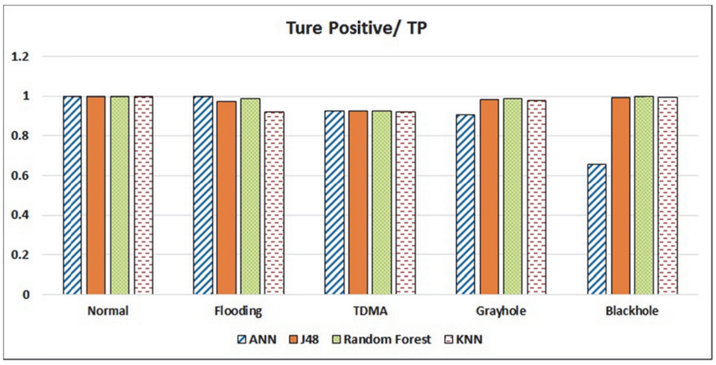

Fig. 4. TP Comparison among DMTs outperformed ANN.

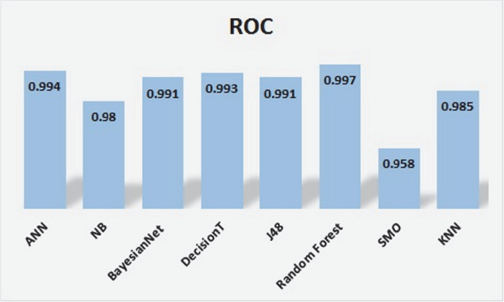

Fig. 5. ROC results of all DMTs.

Fig. 5 shows the evaluation for the tested DMTs in terms of ROC. The majority of techniques have achieved high, close results to each other, but Random Forest was the best among them and SMO was the lowest; the exact ROC numbers for all DMTs are shown in the figure.

The performance of all DMTs in terms of accuracy including both the average TP and ROC are depicted in Fig. 6. Close matching between TP and ROC can be observed by the majority of DMTs except in NB and BayseianNet, where ROC was higher, whereas in SMO it was lower than TP. But, in all cases, the difference did not exceed 2.6%.

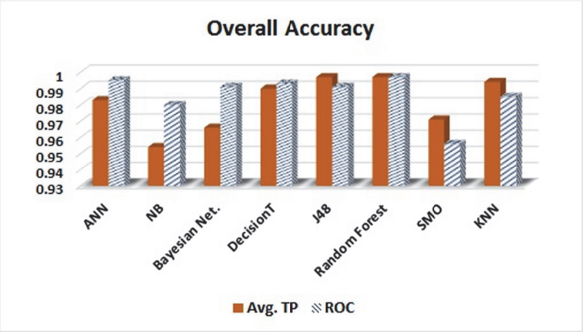

Fig. 6. The overall accuracy of DMTs.

In terms of time complexity, comparisons are presented in Fig. 7. There are large discrepancies among the techniques regarding the time required to build their models. KNN is superior to other techniques with low time complexity where ANN is the most time-consuming model. The time enhancement in case of KNN is 99.9% in comparison with ANN. The time complexity for each of the studied algorithms can be seen in the figure where also SMO and Random forest show more time requirements than other techniques.

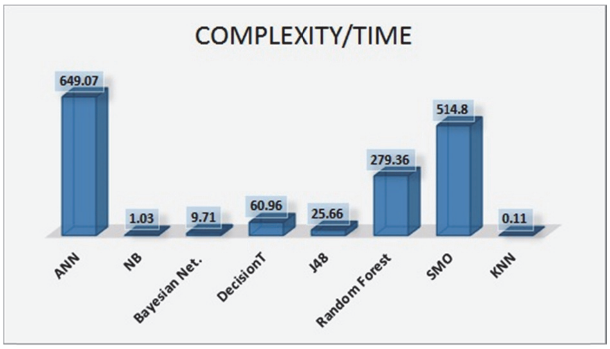

Fig. 7. Time complexity of DMTs.

Table 7 summarizes the calculation of the weighted average of detection rate measured by TP for all types of DoS attacks in addition to the normal case. It can be concluded that the performance was relatively high in all DMTs.

On the other hand, by looking at each attack in specific, the accuracy varies among them. Table 8, shows the maximum and the minimum TP achieved by the corresponding DMTs. Random Forest was the best in detecting Grayhole, Blackhole attacks and the normal instances. NB and BayesianNet were the best in detecting Flooding and TDMA attacks respectively. Whereas, SMO, for example, was the worse technique in detecting Grayhole attacks.

Table 7. Average TP of all types of DoS attacks in each DMT.

DMT Name	ANN	NB	Bayesian Net.	DecisionT	J48	RandF	SMO	KNN
Avg. (TP)	0.983	0.954	0.966	0.99	0.997	0.997	0.971	0.994

Table 8. Maximum/Minimum TP achieved by DMTs.

	Class type				
TP	Normal	Flooding	TDMA	Grayhole	Blackhole
Max Value	0.999	1	0.933	0.99	0.997
DMT Name	J48, RandomForest	NB	BayesianNet	RandomForest	RandomForest
Min Value	0.971	0.898	0.755	0.501	0.659
DMT Name	NB	DecisionT	NB	SMO	ANN

Moreover, the experiments' results reveal that FP mainly occurs in the normal class type for almost all DMTs, whereas it is less occurred when detecting TDMA attacks.

Random forest was the best among other techniques with less FP overall. Table 9 illustrates the FP results in each class type for each technique including ANN (the technique considered by the related work).

Table 9. FP in the normal and attacks' scenarios for all data mining techniques.

Avg. FP	Data mining technique (DMT)							
	ANN [21]	NB	BayesianNet	DecisionT	J48	Random Forest	SMO	KNN
Normal	0.015	0.012	0.012	0.071	0.020	0.017	0.087	0.028
TDMA	0.000	0.000	0.020	0.000	0.000	0.000	0.001	0.001
Blackhole	0.004	0.018	0.001	0.001	0.000	0.000	0.015	0.000
Grayhole	0.011	0.018	0.010	0.001	0.001	0.000	0.005	0.001
Flooding	0.001	0.010	0.002	0.001	0.000	0.000	0.001	0.001

In reference to the above results, we can conclude that choosing proper DMT should prioritize the studied performance metrics whether accuracy or complexity, WSN requirements, and resources in addition to attack types. More discussion is presented in section 4.3.

4.2 Features Selection

It is important to understand, which features are the most influential features in determining the attack type. This will examine how well each attribute can individually differentiate attack types. Hall and Holmes [33] categorized feature selection algorithms to (1) algorithms that evaluate individual attributes and (2) algorithms that evaluate a subset of attributes. The first category of feature selection algorithms identifies which attribute is able to serve as a discriminatory attribute for indicating the attack type. The second category selects a subset of features that are best to identify the class label.

In this paper, we tried several popular feature selection algorithms, namely Information Gain, Gain Ratio, Correlation, ReliefF, and Principal Component. None of these algorithms gave us very good results. Therefore, we have consulted LEACH protocol experts who did a deep analysis of the attack models stimulated during the execution of WSN services to construct WSN-DS dataset. Their deep analysis revealed some of the important features to indicate the attack type. Originally, 19 features were considered in detecting the four types of DoS attacks in addition to the normal case. However, after a thorough study of these features and a well understanding of LEACH protocol and its messages, a set of features that might contribute to the classification of attacks were selected by the experts. Their list agrees to some extent with the Gain Ratio algorithm results but with some differences. We have decided to select 9 features out of 19 which are more influential in detecting the attacks described previously in section 3.1. The rest of features could be more valuable to detect other types of attacks such as Sybil attacks, DoS attacks originated from the member sensors themselves not the CHs, and many others that could be investigated in a separate study. These selected features are ADV_S, Is_CH, SCH_S, DATA_S, SCH_R, Data_Sent_To_BS, ADV_R, DATA_R and Attack Type.

The new obtained dataset has also been examined by applying same DMTs to study the impact of this features selection. Table 10 summarizes the results of the evaluation metrics for all DMTs before and after features selection.

Table 10. Overall average DMTs performance in case of full dataset or subset dataset (after features selection).

DMT(s)	Avg. True Positive (TP)		Avg. ROC		Time Complexity	
	Full dataset	Subset dataset	Full dataset	Subset dataset	Full dataset	Subset dataset
ANN [21]	0.983	0.978	0.994	0.992	649.07	310.2
NB	0.954	0.957	0.98	0.982	1.03	0.55
BayesianNet	0.966	0.959	0.991	0.988	9.71	2.1
DecisionT	0.99	0.979	0.993	0.993	60.96	16.15
J48	0.997	0.982	0.991	0.992	25.66	10.71
Random Forest	0.997	0.982	0.997	0.997	279.36	191.66
SMO	0.971	0.962	0.958	0.947	514.8	1695.49
KNN	0.994	0.982	0.985	0.995	0.11	0.06

Interesting results can be concluded and observed in Figs. 8 and 9, which show high attainment of detection accuracy and a major reduction in time complexity after the features selection.

Fig. 8 proves efficient feature selection, which did not affect the accuracy of correctly classified instances. On the other hand, it improves significantly the time required to build the models in all techniques except SMO.

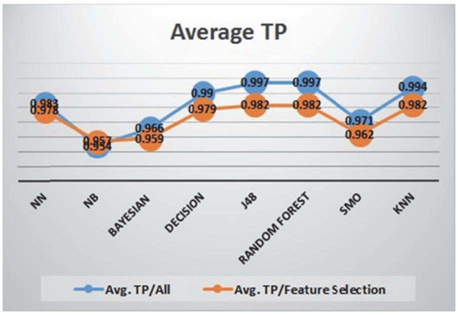

Fig. 8. Accuracy comparison before and after feature selection.

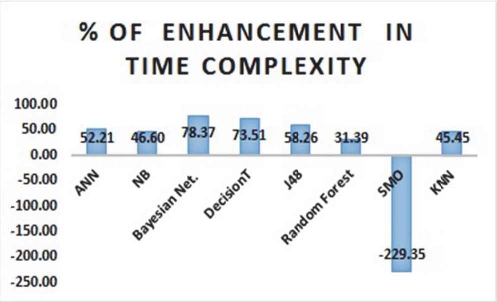

Fig. 9. Percentage of enhancement in time complexity after feature selection.

Fig. 9 shows the reduction in time complexity by up to 78.37 %. Only SMO has been affected negatively by reducing the number of features. This is due to the long time SMO takes to converge, which seems to be an issue of the algorithm being trapped with a local minimum and not finding the global one. Such behavior has been already discussed in other studies [34, 35].

ROC was the most stable metric as it reported minor changes after the features selection process in all studied techniques as clearly exposed in Table 10.

4.3 Security Analysis

Fast, accurate detection of security attacks would introduce more efficient IDSs, which accordingly provide more secure WSN services. In this paper, since LEACH is the protocol under study, the proposed IDS should analyze the data coming from the sensors at the end of each round and be able to detect the security attackers. Then, notify the sink and the corresponding sensor nodes. This should be done before the commencement of the next round where new CHs will be assigned; as the protocol states. Also, to make sure malicious nodes will be hindered from being elected as CHs to protect the network from lunching DoS attacks.

In reference to the implementation of LEACH protocol, the round time is calculated in Eq. (20). The initial energy of the sensor nodes was set to 10J, which means each round will last for 50 seconds. After that, LEACH will start a new round with a new set-up phase as explained before.

$$RoundTime(s) = 5{*}initialEnergy(joule) \tag{20}$$

Therefore, it is recommended to send the monitored data before the end of each round in order to apply the required analysis and send the attacks' alerts before the beginning of the next round. In other words, within the same 50 seconds. Table 11 provides an analysis (using Eq. 21) for the studied DMTs and when they are able to report the existence of DoS attacks.

$$AttackDetection\ Reqularity = \lceil ModelTime/RoundTime \rceil \tag{21}$$

Table 11. Alerting regularity (in terms of rounds) for all DMTs.

DMT	Reporting/Alerting Time (Round Number)	
	All features	Selected Features
ANN	13	7
NB	1	1
Bayesian Net	1	1
DecisionT	2	1
J48	1	1
Random Forest	6	4
SMO	11	34
KNN	1	1

Considering all features; NB, Bayesian Net, J48, and KNN are able to detect the attacks at the end of each round and prevent them from entering the next round. DecisionT, Random Forest, SMO, and ANN will react to the intrusion detection every 2, 6, 11 and 13 rounds respectively. Thus, the attackers in case of Random forest, for example, will keep functioning for 6 continuous rounds before being dismissed from the network. But, after applying features selection, the majority will be able to take an action at the end of each round, which definitely will enhance the security and limit the impact of attackers especially when the detection rate is also high.

Therefore, taking proper decisions will contribute to building efficient IDS for WSN. Such decisions include the selection of suitable DMTs to be injected in the IDSs. The selection should not consider only the accuracy of the detection but also the complexity of DMT models, the network characteristics and requirements, and the available resources such as the initial sensors' energy.

5. CONCLUSIONS AND FUTURE WORK

This paper introduced an efficient IDS for detecting DoS attacks in WSN. Firstly, the paper reviewed some of the existing IDSs which applied data mining techniques in their solutions and provided a detailed comparison among them. Then an architecture for integrating DMTs in WSN's IDS has been proposed. Eight DMTs have been examined by applying them to a new, specialized dataset for WSN (named as WSN-DS) and evaluating them in terms of accuracy and complexity. Four types of DoS attacks were addressed in this study: Flooding, TDMA, Grayhole, and Blackhole. The capability of DMTs in detecting these DoS attacks has been studied. While the results show high accuracy rates, different levels of complexities are also observed. The complete WSN-DS was used to run this empirical study. To accommodate the requirements of WSNs and their limited resources, the complexity of building DMTs needed to be reduced. Therefore, this paper also implemented a feature selection algorithm, which reduces 53% of the features without affecting the accuracy rates, but reducing the complexity to reach 78.37% in some DMTs. The impact of selecting suitable DMT to be part of the data analysis component in the IDS has been discussed from security perspectives. Accurate and early detection of DoS attacks will limit and contain the damage caused by them. Consequently, ensuring persistent, efficient services provided by WSNs.

For the future work, measuring the impact of delaying the exclusion of attacks due to model complexity in terms of energy consumption and data correctness and delivery could be investigated. WSN-DS could also be tested to explore other types of attacks, such as Sybil attacks. Building IDS that integrates other computational intelligence components could be attempted and compared with the outcomes of this research.

REFERENCES

1. K. Chelli, "Security issues in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the World Congress on Engineering*, Vol I, 2015, pp. 1-6.
2. P. B. Hari and S. N. Singh, "Security issues in wireless sensor networks: Current research and challenges," in *Proceedings of International Conference on Advances in Computing*, *Communication and Automation*, 2016, pp. 1-6.
3. 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, 2015, pp. 131-139.
4. D. K. Sadhasivan and K. Balasubramanian, "Fusion of multiagent functionalities for effective intrusion detection system," *Security and Communication Networks*, Vol. 2016, Article ID 6216078, 15 pages.
5. M. Bijone, "A survey on secure network: Intrusion detection & prevention approaches," *American Journal of Information Systems*, Vol. 4, 2016, pp. 69-88.
6. O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *Proceedings of the 6th International Conference on Modeling*, *Simulation, and Applied Optimization*, 2015, pp. 1-6.
7. R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, Vol. 42, 2014, pp. 1-23.

8. S. S. Pandit and D. 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, 2015, pp. 2674-2681.
9. S. Duhan and P. Khandnor, "Intrusion detection system in wireless sensor networks: A comprehensive review," in *Proceedings of International Conference on Electrical, Electronics, and Optimization Techniques*, 2016, pp. 2707-2713.
10. Abhaya, K. Kumar, R. Jha, and S. Afroz, "Data mining techniques for intrusion detection: A review," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 3, 2014, pp. 6938-6942.
11. C. So-In, N. Mongkonchai, and P. Aimtongkham, "An evaluation of data mining classification models for network intrusion detection," in *Proceedings of the 4th International Conference on Digital Information and Communication Technology and its Applications*, 2014, pp. 90-94.
12. L. Coppolino, S. D'Antonio, A. Garofalo, and L. Romano, "Applying data mining techniques to intrusion detection, in wireless sensor networks," in *Proceedings of the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2013, pp. 247-257.
13. P. Amudha and H. A. Rauf, "Performance analysis of data mining approaches in intrusion detection," in *Proceedings of International Conference on Process Automation, Control and Computing*, 2011, pp. 1-6.
14. 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.
15. S. R. Chakole, V. Balpande, and V. Giripunge, "A powerful tool for intrusion detection & clustering techniques and methodology," *International Journal of Computing and Technology*, Vol. 1, 2014, pp. 587-592.
16. Y. E. 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*, Vol. 6, 2015, pp. 164-172.
17. Knowledge Discovery and Data Mining (KDD datasets), https://kdd.ics.uci.edu, last access August 29, 2017.
18. 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, 2013, pp. 1621-1624.
19. P. Amudha, S. Karthik, and S. Sivakumari, "Classification techniques for intrusion detection – An overview," *International Journal of Computer Applications*, Vol. 76, 2013, pp. 33-40.
20. C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, Vol. 36, 2009, pp. 11994-12000.
21. I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *Journal of Sensors*, Vol. 2016, Article ID 4731953.
22. 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, 2000, pp. 1-10.

23. 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*, 2014, Vol. 3, pp. 5883-5890.
24. D. Kumar, "Performance analysis of energy efficient clustering protocols for maximizing lifetime of wireless sensor networks," *IET Wireless Sensor Systems*, Vol. 4, 2014, pp. 9-16.
25. 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, 2014, pp. 21-27.
26. 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, 2013, pp. 623-645.
27. S. Taneja, "An energy efficient approach using load distribution through LEACH-TLCH protocol," *Journal of Network Communications and Emerging Technologies*, Vol. 5, 2015, pp. 20-23.
28. S. Peng, T. Wang, and C. P. Low, "Energy neutral clustering for energy harvesting wireless sensors networks," *Ad Hoc Networks*, Vol. 28, 2015, pp. 1-19.
29. Y. M. Miao, "Cluster-head election algorithm for wireless sensor networks based on LEACH protocol," *Applied Mechanics and Materials*, Vol. 738-739, 2015, pp. 19-22.
30. P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?" in *Proceedings of International Conference on Image Analysis and Processing*, 2005, pp. 50-57.
31. M. Alenezi, S. Banitaan, and Q. Obeidat, "Fault-proneness of open source systems: An empirical analysis," in *Proceedings of International Arab Conference on Information Technology*, 2014, pp. 1-5.
32. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, MA, 2016.
33. M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, 2003, pp. 1437-1447.
34. A. G. K. Janecek, W. N. Gansterer, M. A. Demel, and G. F. Ecker, "On the relationship between feature selection and classification accuracy," *Proceedings of Machine Learning Research*, Vol. 4, 2008, pp. 90-105.
35. N. H. AlNuaimi, M. M. Masud, and F. Mohammed, "Examining the effect of feature selection on improving patient deterioration prediction," *International Journal of Data Mining & Knowledge Management Process*, Vol. 5, 2015, pp. 13-33.

Iman Almomani is an Associate Professor and Associate Chair of the Department of Computer Science at Prince Sultan University, KSA. Iman is also the Associate Director of research and initiatives center. Before joining Prince Sultan University, Iman worked as an Associate Professor and Head of the Computer Science Department at the University of Jordan, in Jordan. Iman received her Ph.D. degree from De Montfort University, UK in Network Security in 2007. Her research interests include wireless networks and security, mainly wireless mobile ad hoc NETworks (WMANETs), wireless sensor networks (WSNs), multimedia networking (VoIP) and security issues in wireless networks. Iman is also a senior member of IEEE and IEEE WIE.

Mamdouh Alenezi is currently the Chief Information and Technology Officer (CITO) at Prince Sultan University. Dr. Alenezi received his MS and Ph.D. degrees from DePaul University and North Dakota State University in 2011 and 2014, respectively. His research interests include software engineering and data mining.