*PLOMS Journal of*
*Artificial Intelligence*
*(PLOMS AI)*

Article

# Intrusion Detection Based Learning in Wireless Sensor Networks

Rabie A. Ramadan [1,*] and Karen Medhat[1]

[1]Computer Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt; rabie@rabieramadan.org

Correspondence: rabie@rabieramadan.org;

Received: 01-08-2021; Accepted: 10-08-2021; Published: date: 12-08-2021

**Abstract:** Wireless Sensor Networks (WSNs) have different limitations, including storage and processing capabilities. Besides, sensors' communication ranges are limited. Besides, those limitations raise the issue of the network intrusion and sensors' capabilities in detecting intruders. This paper introduces two different algorithms to detect intrusions in wireless sensor-based systems, which are more vulnerable to attacks. The first proposed algorithm is supervised learning-based classification. On the other hand, the second algorithm is unsupervised learning-based clustering. Both algorithms try to detect intrusions using a set of detection rules that are structured in the form of decision trees. The algorithms are detailed and extensively examined on a well-known dataset. They also are tested against two different architectures, two and three levels networks. The three-level architecture represents the sensor node, sink node, and base station level, while the two-level architecture represents the levels of sensor and sink nodes. An enhancement for decision-tree-based classification algorithms is also proposed by changing the decision tree to a binary tree. Such change made a significant enhancement in the complexity of reaching a decision. The produced decision trees use a similar decision tree node structure as the one used in Classification and Regression Trees (CART). The performance of our proposed algorithms and techniques are measured and extensively examined.

**Keywords:** Wireless Sensor Networks; Intrusion; Regression Trees; Classification; Decision Tree

## I. Introduction

WSN [1] is one of the emerging technologies. It is used in many applications [2] and systems such as monitoring and tracking systems. The sensor nodes that are considered the main structure of the WSN can be deployed with different topologies such as star, tree, and mesh and can communicate using different methodologies. Wireless Sensor Networks are extensively used in many applications related to various fields, and in most cases, classified, and potentially important information should be secured from unauthorized access. WSNs are implemented with limited energy resources in very harsh environments. Besides, WSN devices cannot tolerate network failures triggered by intruders in the network. Therefore, protection measures must be taken into consideration for the prevention of intrusions on sensor nodes.

Efficient algorithms for intrusion detection are implemented in this work. Two intrusion detection algorithms are provided using a supervised learning mechanism, the other using an unsupervised learning mechanism. A set of detection rules is built in a binary decision tree in each of these algorithms. The learning algorithms are trained, and the decision trees are built before the network's operation, and then the decision trees are loaded to the sensor nodes to detect intrusions during the network's operation. The intrusion detection algorithms were used in two different network architectures, shown in figure 1 and figure 2.

The first architecture consists of the sensor node, sink node, and base station. In the second model, sensor and sink nodes level is considered. The supervised learning mechanism is used on the sensor node level, while on the sink node and base station levels, the decision tree is established by the unsupervised learning mechanism is used. The network architectures were set to monitor the differences between the numbers of the generated intrusion data packets for each architecture.

The introduced algorithms provided a high detection accuracy compared to decision-tree-based algorithms using less number of selected features (where the most relevant features were selected using Entropy and Pruning during the training phase) compared to previous work for feature selection. Usage of decision trees to detect the intrusions decreases the complexity of reaching a decision compared to previous work for intrusion detection where neural networks and other complex methodologies were used. Only 10% of the training dataset was used in the suggested learning algorithms. An enhancement is also implemented on the decision-tree classification algorithm that reduces the decision tree scale and makes it suitable for intrusion detection in WSNs.

The paper is organized as follows; Section 2 introduces the background and relevant research topics. Section 3 presents the main focus of the research proposed in this paper. Section 4 shows the WSN architecture that was set for the experiments and a brief introduction to the proposed algorithms. Section 5 is an overview of the nodes structure used to build the decision trees in the algorithms. Section 6 is an overview of the proposed supervised learning algorithm. Section 7 explains the details of the unsupervised learning algorithm proposed in this research. Section 8 introduces the proposed enhancement for the CART algorithm. Section 9 shows the results for the proposed algorithms on the KDD dataset. Section 10 shows the results for the proposed algorithms on the ADLs dataset. Section 11 shows the results of the proposed algorithms on the LWSNDR dataset. The paper is concluded in section 12. Finally, section 13 gives recommendations and future research directions.

## II. Background

WSNs face different challenges due to the different limitations of the resources in WSNs. WSNs challenged have been discussed in [3-8], affecting the overall network structure. This also affects the used protocols where specific restricted protocols and algorithms are forced to be used on the limited devices. The adjustments of already existing algorithms and protocols to be used in WSNs may also be considered. One of the main methods to secure the communications systems is to detect intrusions by building a reliable Intrusion Detection System (IDS). James Anderson in 1980 [9] introduced the concept of intrusion detection as "attempt or threat is the potential possibility of a deliberate unauthorized attempt to access information, manipulate information, or render a system unreliable or unusable." Anderson made an investigation about intrusions and intrusion detection, where he discussed the definition of fundamental terms of intrusions and intrusion detection, which are:

Risk: The exposure of information unpredictably.
Threat: The unauthorized access to the data or the network.
Attack: The execution of a plan to perform a threat.
Vulnerability: The flaw in the system or network that makes it vulnerable to attacks.
Penetration: The successful attack.

There are different methodologies to detect intrusions where the most commonly used are Signature-based intrusion detection and anomaly-based intrusion detection. In signature-based, the patterns of the intrusions are defined in a database. If the system is attacked, the received data pattern is compared to the saved ones, and an intrusion is reported whenever a match is found. This method is very effective when the attacks are known, but it is not efficient with unknown attacks. Their patterns are not recognized for the unknown attacks, and the attacks pass through the system as if they are normal activity.

For the anomaly-based, the normal behavior of the system is defined. If the system finds any activity that differs from the normal behavior, it generates an alarm. This method effectively detects new types of attacks to the system, but it may generate false alarms for normal system activity.

Monitoring of intrusions can be done at the host level, network level, or hybrid based [10, 11]. The host's activities are monitored for the host-based intrusion detection systems, and any suspicious activity is marked as an intrusion. For the network-based intrusion detection systems, the activities and the data packets sent on the network are monitored, and any suspicious activity or data packet is marked as an intrusion. For the hybrid-based approach, both the host-based and network-based methods are used in the same system. Artificial Intelligent techniques are considered the most common techniques used for intrusion detection. For instance, the Artificial Neural Network [12] is used to build a model to recognize patterns to identify any system's abnormal activity. In addition, state transitions tables [13] can be used to describe the sequence of activities that an intruder does.

The problem of intrusion detection and securing communications systems has attracted attention recently. Many researchers address intrusion detection in communication systems [14-18]. In addition to this, many researchers have addressed the security mechanisms that can be used generally in communications systems [19, 20] and those that can be explicitly used in mission-critical communication systems [21].

Ajenjo et al. [22] discussed the importance of monitoring traffic patterns in networks. The monitoring and the analysis of the traffic were applied to NATO's system. Similarly, Kumar et al. in [23] proposed an approach called AMGA2–NB. The approach contains three phases, and it uses a genetic algorithm to choose a set of solutions from a pool of proposed solutions to be used for intrusion detection. A set of individual solutions is generated in the first phase of the algorithm from the fitness function. The generated set of solutions is approximated in the second phase to generate an improved chromosome. The first and second phases' output acts as an input for the third phase, where the final ensemble's prediction is introduced.

Kruegel et al. [24] proposed an approach to enhance intrusion detection for the systems using anomaly-based algorithms. The authors highlighted some of the reasons behind reporting false alarms from the anomaly-based. One of the main problems for reporting false alarms is the simplicity of combining the model outputs on which the decision is based. The other problem is that the result is not supported with extra information to increase its confidence. The authors based their intrusion detection model on the Bayesian network to solve the mentioned problems. Bayesian network improved the process of combining the model outputs with taking a more accurate decision; in addition to this, it used additional information to strengthen the output credibility.

Krontiris et al. in [25] introduced a decentralized scheme for intrusion detection. Each device in the network has four modules:

- Local Packet Monitoring module which gathers the data to be sent to the Local Detection module.
- Local Detection Engine which collects the data sent to it by the Local Packet monitoring module. It analyzes the collected data and stores the specifications that describe the correct operation.
- Cooperative Detection Engine, if this engine detects an intrusion, sends the state information of the local detection module to the neighbors and receives information from the same module included in the neighboring devices and then applies a majority vote rule to indicate if there is an intrusion or not.
- Local Response module takes the appropriate actions to restore the normal network operation and isolate the intruded part when an intrusion is detected.

Silva et al. [26] introduced a decentralized intrusion detection algorithm. The authors defined a set of rules to be applied by the algorithm on collected features for intrusion detection. Each rule can detect a specific type of attack. The proposed algorithm contains three phases; the first phase is the data acquisition, the second phase is the rule application, and the last phase is intrusion detection. The analysis of the acquired data is done in the first phase. In the second phase, the introduced rules are applied to the data, and an intrusion is detected if any of the rules fail. In the last phase, an intrusion is reported if the number of detected intrusions reaches a certain threshold. The threshold expresses the number of expected attacks in the network. Similarly, Ravale et al. [27] introduced an approach to select the significant features according to the attack type. The authors used K-means clustering for building the clusters used to take the initial decision. Linear support vector machines were then used to make the most accurate decision regarding reporting an intrusion.

Decreasing the feature space used by the intrusion detection algorithms is taken into consideration by several researchers. For example, Karan et al. [28] proposed a technique for feature selection using a combination of feature selection algorithms like Information Gain, Gain Ratio, Correlation Attribute Evaluation. The authors tested the selected features' performance on different classification algorithms such as J48, Naïve Bayes, NB-Tree, Multi-Layer Perceptron, SVM, and SimpleCart. J48 is a classification algorithm that builds a decision tree using Entropy. J48 [29] can handle both discrete and continuous data. J48 can abide by missing attribute values by not including them in the Entropy calculation. Naïve Bayes is a classifier that applies the Bayes probability theorem to build a conditional probability model from which a classifier is built [30, 31]. NB-Tree [32] is a decision tree-based classifier that uses Naïve Bayes as each node. Multi-Layer Perceptron is a neural network classifier [33]. Support Vector Machines (SVM), also known as Support Vector Networks, are sets of supervised learning models that analyze and recognize patterns in the data [34]. SimpleCart is a decision tree-based algorithm and builds it based on Entropy [35]. Ruirui et al. [36] introduce an approach for intrusion detection using a negative selection algorithm where an immune layer is added to the network stack, and the intrusions are detected on the cluster heads' level. In some of the mentioned research papers, different aspects of intrusion detection systems were not covered. For example, the approach introduced by Kumar et al. in [23] contains different phases and steps to reach a decision that increases the complexity of making a decision. The approach introduced by Kruegel et al.[24] focused only on anomaly-based algorithms. Krontiris et al. [25] decentralized intrusion detection approach accuracy is not clearly stated to be a high detection rate, and the process for detecting the intrusion is very complex. The accuracy of the approach introduced by Silva et al. [26] is not mentioned to be high.

Looking at the intrusion detection development, some of the solutions tackled monitoring some network specifications and detecting any deviation from the normal behavior according to these specifications. Some solutions monitor the network's characteristics regarding the communications parameters and the parameters of data packets in the network to detect any abnormalities in these parameters—other solutions considered studying the network's behavior and detecting any differences or abnormalities from the normal behavior. Table 1 summarizes some of the research efforts in the field of intrusion detection in wireless sensor networks..

**Table 1** Intrusion detection Approaches

| Reference | Approach | Advantages and Disadvantages |
|---|---|---|
| [23] | AMGA2–NB | It enhanced the detection accuracy concerning other NB-based algorithms for classification. The computation overhead and the energy needed to apply these algorithms are not the most convenient regarding the limited resources of nodes in WSNs. |
| [24] | Bayesian network-based algorithm | Convenient detection accuracy by detecting the false alarms but the considerably large computation |
| [25] | Decentralized detection approach | The detection is decentralized at each device of the network. but the very complex approach in addition to this the detection accuracy is not mentioned to be very high |
| [26] | The decentralized rule-based detection approach | Each rule detects a certain type of attack, which may result in different attacks not to be considered; in addition to this, the detection accuracy is not mentioned to be high |
| [36] | NSA Based detection algorithm | Convenient detection accuracy but the centralized approach of high computational complexity |

## III. Primary Focus of the Paper

This paper aims to introduce intrusion detection mechanisms that can save the sensor nodes' energy and memory resources in the WSN compared to other existing methods. This work presents two intrusion detection algorithms, supervised and unsupervised learning algorithms. The algorithms are trained before the network's operation to get the detection rules structured in a decision tree. This is to save the energy resources of the sensor nodes in the WSNs. The decision tree nodes are in the form of <feature, value> pairs, where the feature represents an intrusion parameter that is monitored by the sensor node and a corresponding threshold value for this feature by which an intrusion can be detected.

The produced decision trees use a similar decision tree node structure as the one used in Classification and regression trees (CART) [37]. However, the decision trees produced from the proposed algorithms consider both categorical and numeric values for the features in a single classification decision tree instead of producing a classification tree for categorical features' values or a regression tree for numeric features' values as it's the case in CART.

Feature selection is applied where Entropy can be used, according to the application, to choose the most relevant <feature, value pairs> in addition to the pruning process at the training phase. The selected features proved their relevance to classification accuracy when applied to other classification algorithms. They gave higher accuracy than other feature selection techniques (information gain, gain ratio, and correlation attribute evaluation), especially for the algorithms based on neural networks and support vector machines. In addition, the decision trees of the proposed algorithms gave high detection accuracy compared to other decision-tree-based algorithms. The decision trees were loaded to sensor nodes and were tried during the network's operation to monitor the number of generated intrusion detection data packets.

The network architecture was set as introduced in [38], where the network architecture is a three-layer architecture. This three-layer architecture contains a sensor node layer having the decision tree produced from the supervised learning algorithm, a sink node layer, and a base station layer. The other architecture proposed is two-layer architecture for simplification where the base station layer is removed. The other two layers are kept where the senor and sink layers have the decision trees produced by supervised and unsupervised learning algorithms.

The intrusions detected at each layer are reported to upper layers to reach the base station layer in three-layer architecture and reach the two-layer architecture's sink node layer. The intrusions are reported to upper layers to be collected at a centralized sensor node (sink node in two-layer architecture and base station in three-layer architecture). The decision to report intrusions during the network's operation will be taken according to the application for which the WSN is built. Suppose the application cares about collecting the details of all the intrusions in one central point more than losing the energy of reporting intrusions' data. In that case, the process of reporting the intrusions will be applied. If not, then the solution will be applied on the sensor nodes' level without reporting the network's upper layer intrusions.

The proposed solution detects intrusions with high detection accuracy using binary decision trees. Besides, reaching a decision using a binary tree is less complex than other methods such as Naïve Bayes, Neural networks, and Support Vector Machines. The size of the tree is reduced compared to other decision tree-based algorithms. The decision trees are not over-fitted on the training data and were trained on only 10% of the data. The solution saves the sensor nodes' energy as the training phase and the decision tree building are done before the network's operation.

To summarize our contributions in this paper, the paper proposes the following:

- Supervised learning Intrusion Detection Algorithm (SLIDA).
- Unsupervised Learning Intrusion Detection Algorithm.
- Examining the WSNs' different architectures such as Base station-based and Sink node-based architectures.
- Efficient intrusion detection features are selected and compared to other proposed features.
- An enhanced version of the decision-tree based classification algorithm reducing the decision tree size.

## IV. Network Architecture and the Proposed Algorithms

Materials and Methods should be described with sufficient details to allow others to replicate and build on published results. Please note that publication of your manuscript implicates that you must make all materials, data, computer code, and protocols associated with the publication available to readers. Please disclose at the submission stage any restrictions on the availability of materials or information. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited.

Research manuscripts reporting large datasets that are deposited in a publicly available database should specify where the data have been deposited and provide the relevant accession numbers. If the accession numbers have not yet been obtained at the time of submission, please state that they will be provided during review. They must be provided prior to publication.

## V. Structure of the Decision Trees

The supervised and the unsupervised algorithms introduced in this paper produce a set of intrusion detection rules that are structured in the form of a binary tree. Each node in the decision tree expresses a particular feature and a specific value to this feature. When the algorithm is in operation, the measured feature's value (a feature's value from the data sample) is compared with the feature's value in the tree's node. The comparison of these values determines which branch of the binary tree will be taken next. The intrusion detection binary tree nodes are in the form of < feature, value > pair [39]. The features included in the < feature, value > pairs represent some or all of the features measured by the system. If the number of < feature, value > pairs is relatively large, Entropy (1) will be measured to select some of the features and select some of their values. The features that have a major effect on dividing the data samples into different classes will be selected.

$$\boldsymbol{Entropy(p)} = -\sum_{j=1}^{n}\left(\frac{|Pj|}{|P|}\,\boldsymbol{log}\,\frac{|Pj|}{|P|}\right) \quad (1)$$

Where P is the total number of samples, Pj is the number of samples for class j, and n is the number of classes. For instance, n will be equal to two in an intrusion detection system as there will be only two classes. One of the two classes is for the data samples that describe the system's normal activity, and the other class for the data samples that describe the abnormal activities (attacks) of the system.

## VI. The Supervised Learning Algorithm

Add your results The intrusion detection tree of supervised learning is built using the Ginni Index (2). After the < feature, value > pairs are extracted from the training dataset, the Ginni Index is calculated at each level of the decision tree to choose a pair to be added at that level.

$$\boldsymbol{Gini\ Index} = 1 - \sum_{j=1}^{n}\left(\frac{|\boldsymbol{Pj}|}{|\boldsymbol{P}|}\right)^2 \quad (2)$$

The intrusion detection problem is mainly considered as a classification problem. For the illustration of the algorithm, a simple Weather dataset, shown in Table 2, is used. This dataset is one sample dataset for classification problems in the WEKA data mining tool [40]. Each record in the dataset represents a data sample. The dataset has four features:

Outlook.
Temperature.
Humidity.
Windy.

According to the features' values, the data sample is classified as one of the two existing classes. The two classes are the two values (YES, NO) in the PLAY column. Temperature and humidity are continuous features as they take numerical values, while Outlook and Windy are discrete features as they take non-numerical values.

**Table 2.** Weather Dataset

| OUTLOOK | TEMPERATURE | HUMIDITY | WINDY | PLAY |
|---|---|---|---|---|
| Sunny | 85 | 85 | FALSE | NO |
| Sunny | 80 | 90 | TRUE | NO |
| Overcast | 83 | 78 | FALSE | YES |
| Rain | 70 | 96 | FALSE | YES |
| Rain | 68 | 80 | FALSE | YES |
| Rain | 65 | 70 | TRUE | NO |
| Overcast | 64 | 65 | TRUE | YES |
| Sunny | 72 | 95 | FALSE | NO |
| Sunny | 69 | 70 | FALSE | YES |
| Rain | 75 | 80 | FALSE | YES |
| Sunny | 75 | 70 | TRUE | YES |
| Overcast | 72 | 90 | TRUE | YES |

| | | | | |
|---|---|---|---|---|
| Overcast | 81 | 75 | FALSE | YES |
| Rain | 71 | 80 | TRUE | NO |

Add your results here. The extracted < feature, value > pairs for this dataset are:

- < Outlook ,Sunny >
- < Outlook ,overcast >
- < Outlook ,rainy >
- < Temp , <= 75 >
- < Humidity , <= 75 >
- < Windy ,false >

For outlook, its values can take any of the three non-numerical values (Sunny, Overcast, and Rainy). Thus, all the values will be taken in separate < feature, value > pairs. For Temperature and Humidity, the numerical value that can divide the data samples into two classes (or clusters) evenly will be selected in the <feature, value> pair. The selected value was found to be 75 for Temperature and 75 for Humidity. Windy is a discrete feature of binary values; any of the two values will be selected in the <feature, value> pair. The Ginni Index value is calculated for each pair of the <feature, value> pairs. For the first pair, < Outlook, Sunny >, the dataset will be divided into two sets, as shown in Figure 3.

The data samples having outlook = Sunny are five data samples from the dataset and are included in the YES set. Two out of the five data samples have value play = YES, and three out of the five data samples have value play = NO. The data samples with other outlook values are nine data samples from the dataset and are included in the NO set. Seven out of the nine data samples have value play = YES, and two out of the nine data samples have value play= NO. Then the value of the Ginni Index will be calculated for the < Outlook, Sunny > pair as the following:

For the YES set: $G^{yes}$= 1 - ( ( $\frac{2}{5}$ )$^2$ * ( $\frac{3}{5}$ )$^2$ ) = 0.48

For the NO set: $G^{no}$= 1 - ( ( $\frac{2}{9}$ )$^2$ * ( $\frac{7}{9}$ )$^2$ ) = 0.346

The total value of the Ginni Index for the < Outlook, Sunny > Pair:

$G^{Total}$= $(\frac{5}{14})$ (0.48) + $(\frac{9}{14})$ (0.346)= 0.39365

The same calculations were applied to all the < feature, value > pairs to give the results shown in Table 3.

**Table 3.** Ginni Index values for the WEATHER's dataset < feature, value >pairs

| Feature Value Pairs | Total Value of Ginni Index (GTotal) |
|---|---|
| **Outlook = Sunny** | 0.39365 |
| **Outlook = overcast** | 0.5 |
| **Outlook = rainy** | 0.457 |
| **Temp < = 75** | 0.4428 |
| **Humidity** | 0.43157 |
| **Windy = false** | 0.4285 |

The largest value of the Ginni Index is the one for the < Outlook, overcast > pair. Thus, none of the < feature, value > pairs having the Outlook feature will be selected at the tree's current level. The smallest value of the Ginni Index is the one for < Windy, false > pair. Thus, < Windy, false > pair will be selected at the tree's current level, level zero (root node), as shown in figure 4.

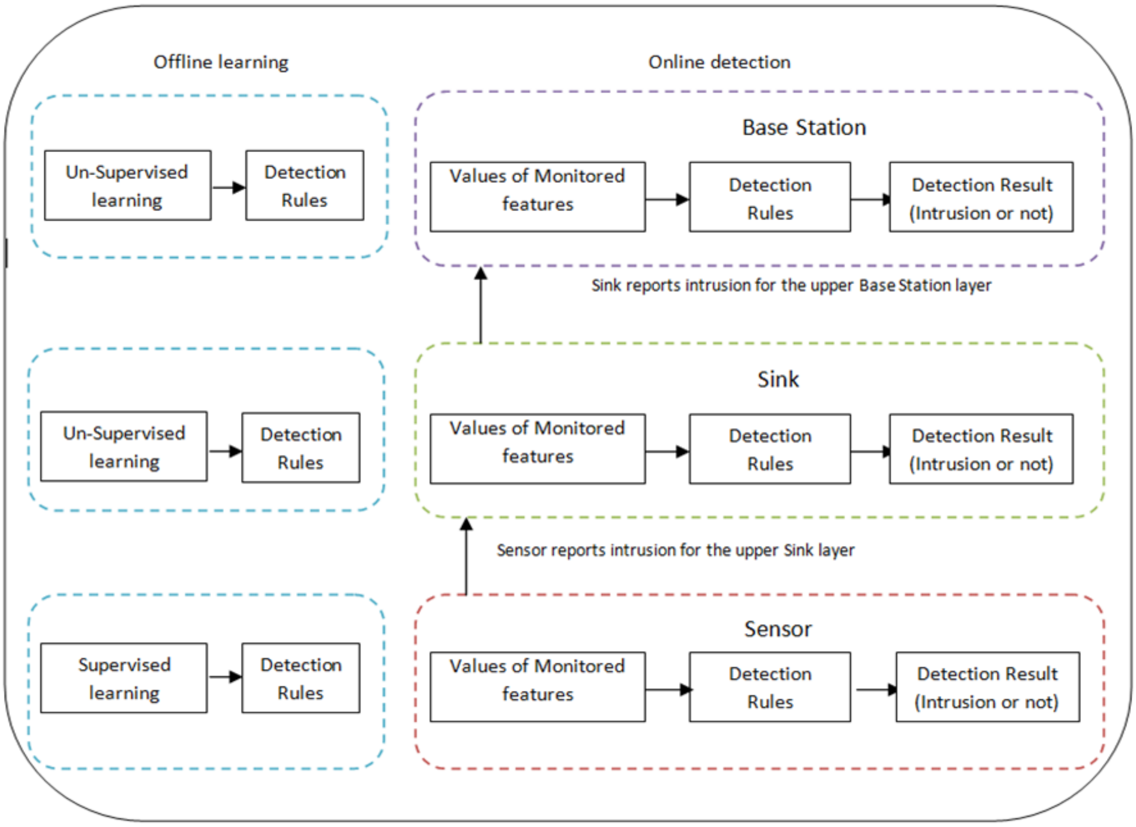
Offline learning
Online detection
Base Station
Un-Supervised learning
Detection Rules
Values of Monitored features
Detection Rules
Detection Result (Intrusion or not)
Sink reports intrusion for the upper Base Station layer
Sink
Un-Supervised learning
Detection Rules
Values of Monitored features
Detection Rules
Detection Result (Intrusion or not)
Sensor reports intrusion for the upper Sink layer
Sensor
Supervised learning
Detection Rules
Values of Monitored features
Detection Rules
Detection Result (Intrusion or not)

**Figure 1.** Three-Layer WSN Architecture

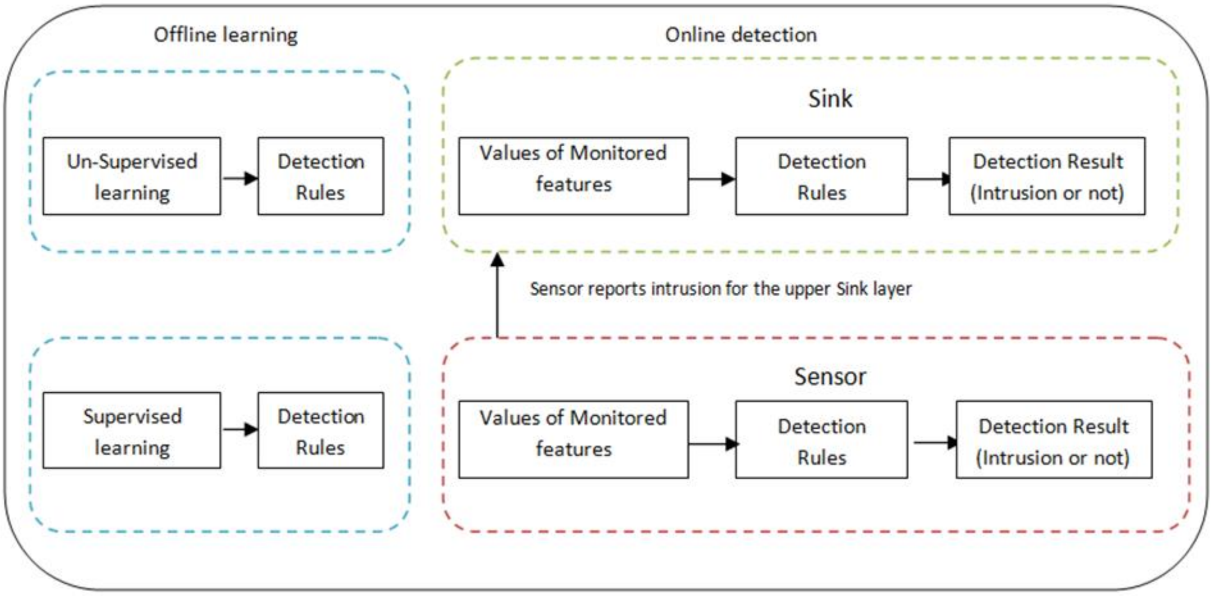
Offline learning
Online detection
Sink
Un-Supervised learning
Detection Rules
Values of Monitored features
Detection Rules
Detection Result (Intrusion or not)
Sensor reports intrusion for the upper Sink layer
Sensor
Supervised learning
Detection Rules
Values of Monitored features
Detection Rules
Detection Result (Intrusion or not)

**Figure 2.** Two-Layer WSN Architecture

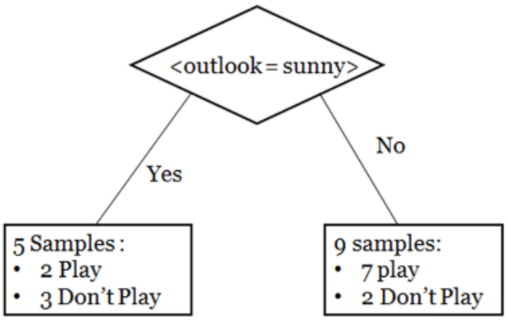

**Figure 3.** Division of data samples of the Weather dataset for <outlook, Sunny> pair

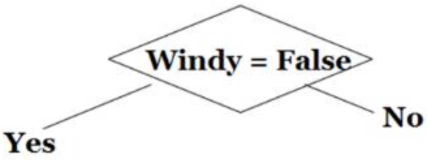

**Figure 4**. Level 0 of the Decision Tree for the supervised learning algorithm

The Ginni Index value is calculated for the data samples in the YES set and for the data samples in the No set for the < Windy, false> pair. The process is repeated till reaching the leaf nodes. Each data sample in the dataset is assigned to one class at the leaf nodes, whether play = YES or play = NO. The levels of the decision tree are shown in figures 5, 6, 7, 8.

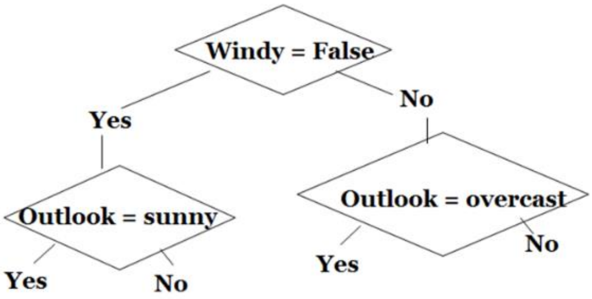

**Figure 5.** Level 1 of the Decision Tree for WEATHER dataset using the supervised learning algorithm

## VII. The Unsupervised Learning Algorithm

The intrusion detection tree of unsupervised learning is built using the Optimal Grouping (3). After the < feature, value > pairs are extracted from the training dataset. The value of optimal Grouping [38] is calculated at each decision tree level to choose the pair to be added at that level. The optimum grouping value is the summation of all Taxon (4) values for the sets of each < feature, value > pair.

$$\boldsymbol{g} = \sum_{i=1}^{L} \lambda^{i} \tag{3}$$

$$\lambda^{i} = \prod_{j=1}^{n} \frac{|\boldsymbol{V}_{j}^{i}|}{|\boldsymbol{D}_{j}|}, \tag{4}$$

Where $|\boldsymbol{V}_{j}^{i}|$ is the length of an interval (in case of the continuous features) or number of values of the appropriate subset $\mathbf{V}_{\mathbf{j}}^{\mathbf{i}}$ (in case of the discrete features); $|\boldsymbol{D}_{j}|$ is the length of an interval between the minimal and maximal values of continuous feature X for all objects from the initial dataset (for the continuous features) or the general number of discrete feature X values for all objects from the initial dataset (for the discrete features).

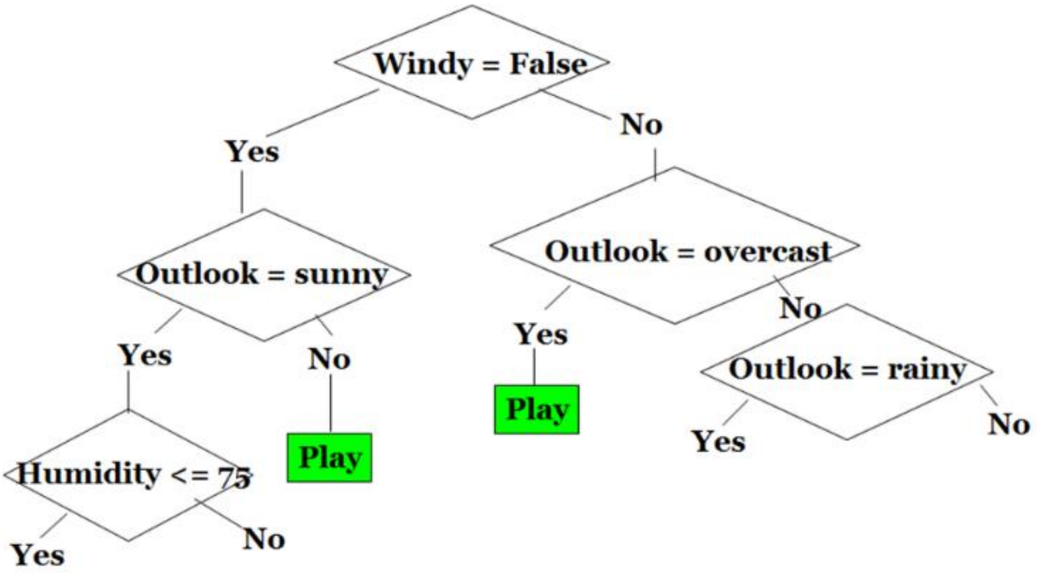

**Figure 6.** Level 2 of the Decision Tree for WEATHER dataset using the supervised learning algorithm

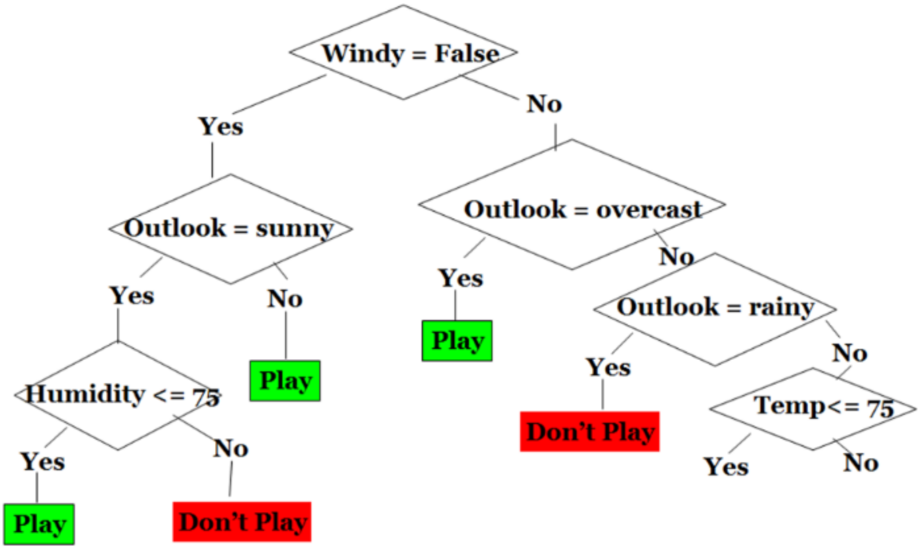

**Figure 7.** Level 3 of the Decision Tree for WEATHER dataset using the supervised learning algorithm

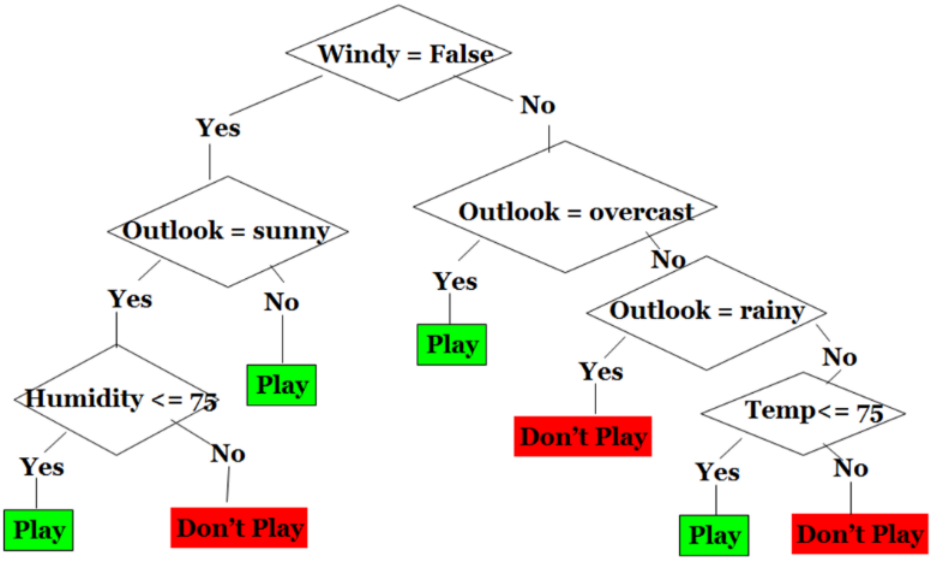

**Figure 8.** Level 4 of the Decision Tree for WEATHER dataset using the supervised learning algorithm (The **final decision tree)**

The illustration of the algorithm will be applied to the simple Weather dataset shown in Table 1. The < feature, value > pairs are as introduced before to be:

< Outlook, Sunny >
< Outlook, overcast >
< Outlook, rainy >
< Temp, <= 75 >
< Humidity, <= 75 >
< Windy, false >

For the first pair, < Outlook, Sunny >, the dataset will be divided into two sets as shown in Figure 1. The data samples having outlook = sunny are five data samples from the dataset (shown in Table 4) and are included in the YES set.

**Table 4.** Data Samples having Outlook = Sunny

| OUTLOOK | TEMPERATURE | HUMIDITY | WINDY | PLAY |
|---|---|---|---|---|
| Sunny | 85 | 85 | FALSE | NO |
| Sunny | 80 | 90 | TRUE | NO |
| Sunny | 72 | 95 | FALSE | NO |
| Sunny | 69 | 70 | FALSE | YES |
| Sunny | 75 | 70 | TRUE | YES |

The value of Taxon $\boldsymbol{\lambda}^{i} = \prod_{j=1}^{n} \frac{|V_j^i|}{|D_j|}$ will be calculated as below for the YES set:

$\boldsymbol{\lambda}^{\text{yes}} = \frac{1}{3} * \frac{(85-69)}{85-64} * \frac{(95-70)}{96-65} * \frac{2}{2} = 0.2048$

As can be seen, the outlook feature has only one value in the YES set (sunny) out of the three values that it can take. For the temperature, the maximum value is 85, and the minimum value is 69 in the data samples included in the YES set. The numerator expresses the difference between these two values (85-69). The denominator expresses the difference between the maximum and the minimum values of the temperature for all the data samples in the dataset. For the humidity, the maximum value is 95, and the minimum value is 70 in the data samples included in the YES set. The numerator expresses the difference between these two values (95-70). The denominator expresses the difference between the maximum and the minimum values of the humidity for all the data samples in the dataset. For Windy, the data samples in the YES set have the two values of this feature. The same calculations were applied for the data samples in the NO set to get the result below:

$\boldsymbol{\lambda}^{\mathbf{No}}$ = 0.5059

The value of optimal Grouping was then calculated from the summation of the Taxon values for both the YES and NO sets:

$$\mathbf{g} = \sum_{\mathbf{i=1}}^{\mathbf{L}} \boldsymbol{\lambda}^{\mathbf{i}} = \boldsymbol{\lambda}^{\mathbf{yes}} + \boldsymbol{\lambda}^{\mathbf{No}} = 0.2048 + 0.5059 = 0.7107$$

The same calculations were applied to all the < feature, value > pairs to give the results shown in Table 5.

**Table 5.** Optimal Grouping values for the WEATHER's dataset < feature, value >pairs

| Feature Value Pairs | Value of Optimum Grouping (g) |
|---|---|
| **Outlook = Sunny** | 0.7107 |
| **Outlook = overcast** | 0.7757 |
| **Outlook = rainy** | 0.7783 |
| **Temp < = 75** | **0.6006** |
| **Humidity** | 0.6789 |
| **Windy = false** | 0.65898 |

The smallest value of the Optimal Grouping is the one for the < Temp, 75 > pair. Thus, it will be selected at the current level (level 0) of the decision tree, as shown in Figure 9.

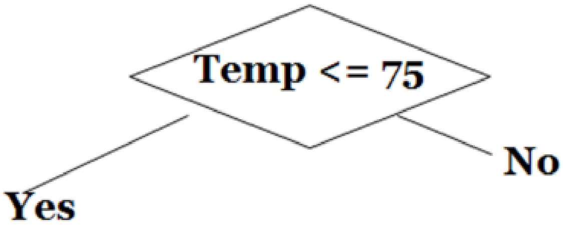

**Figure 9.** Level 0 of the Decision Tree for WEATHER dataset using the unsupervised learning algorithm

The levels of the decision tree are shown in figures 10, 11, 12, 13.

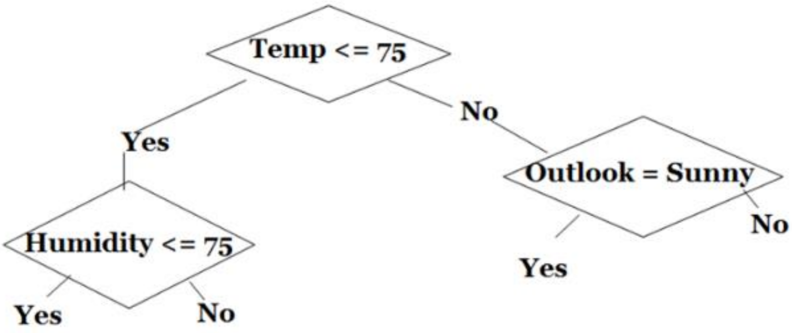

**Figure 10.** Level 1 of the Decision Tree for WEATHER dataset using the unsupervised learning algorithm

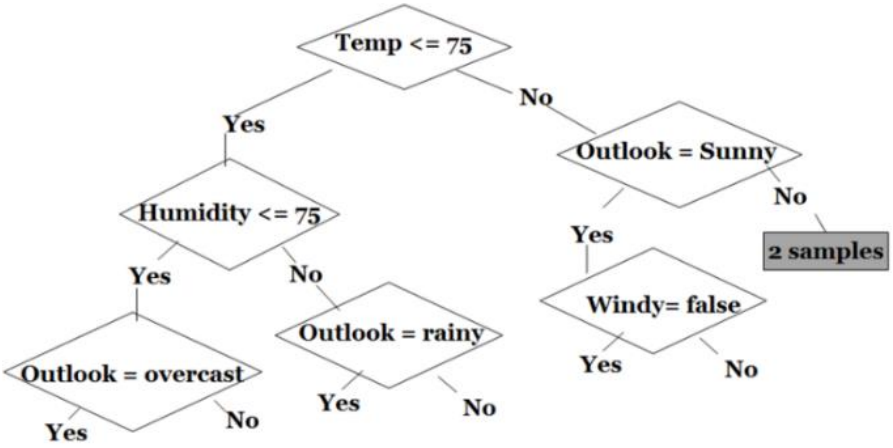

**Figure 11.** Level 2 of the Decision Tree for WEATHER dataset using the unsupervised learning algorithm

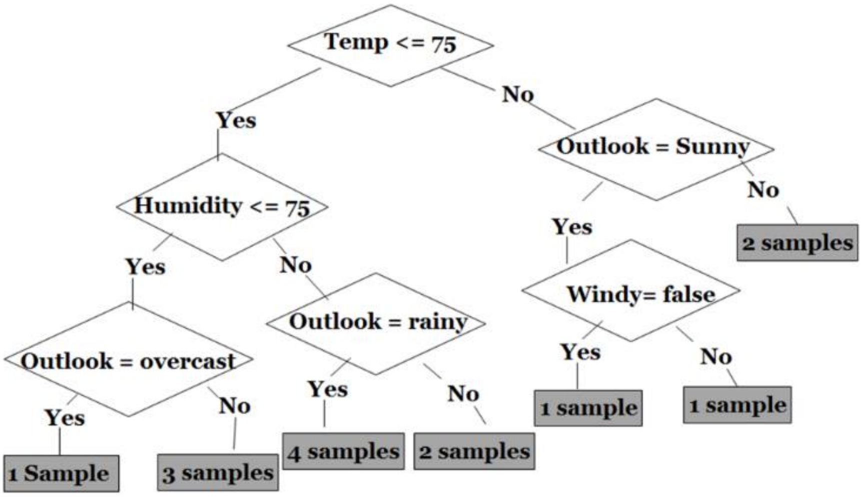

**Figure 12.** Level 3 of the Decision Tree for WEATHER dataset using the unsupervised learning algorithm

The decision tree shown in Figure 10 has seven clusters of one, three, four, two, one, one, two data samples from left to right, respectively. The dataset has 14 data samples, as shown in Table 1. Nine of them are classified as "play", and five are classified as "don't play". Thus, the "play" class appears more in the dataset than "don't play" class. According to the dataset analysis, the clusters having more data samples will be labeled with the class label that appears more in the dataset "play". Thus, the four clusters with three, four, two, and two data samples will be labeled as" play," and the other clusters will be labeled as "don't play" to get the latest decision tree shown in Figure 13.

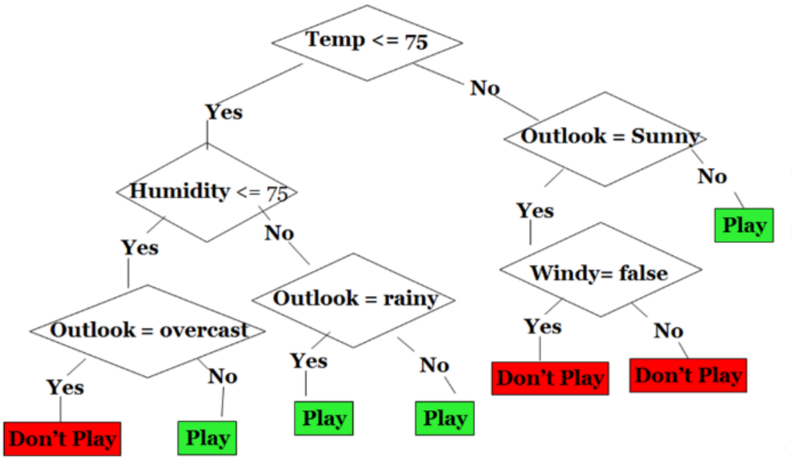

**Figure 13.** Final Decision Tree for WEATHER dataset using the unsupervised learning algorithm

## VIII. The CART Algorithm's Enhancement

The CART decision tree algorithm's enhancement is to prune its branches that have no data samples from the training dataset for the large datasets and convert the algorithm's decision tree to a binary tree. The enhancement was applied to the KDD dataset [41], and the size of the decision tree was greatly decreased. The decision tree's size was reduced from 392 nodes to 145 nodes and from 331 leaves to 146 leaves. Part of the decision tree for the KDD dataset is shown in figure 14, and the enhancement for this part of the decision tree is shown in figure 15.

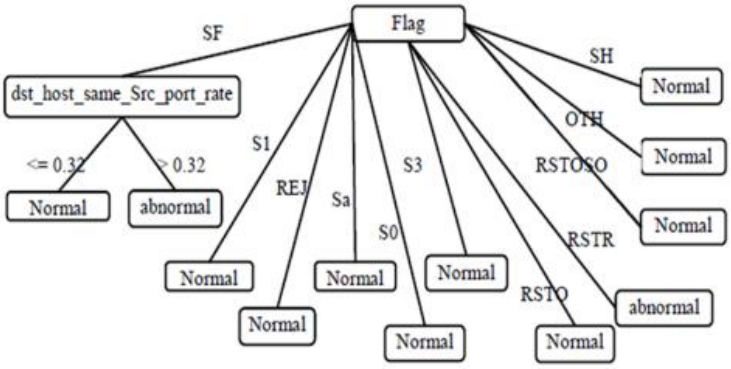

**Figure 14.** Part of Cart Decision tree for KDD dataset

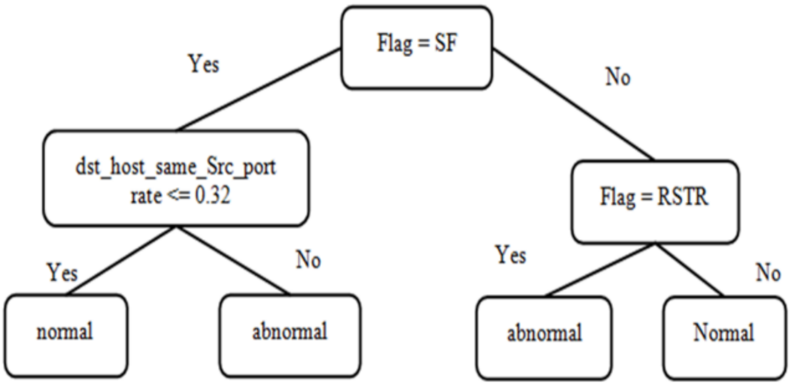

**Figure 15.** Enhancement for part of J48 Decision tree for KDD dataset shown in figure 12

## IX. Detection Accuracy of the Algorithms on KDD Dataset

For the application of the intrusion detection algorithms, 494,021 data samples of the KDD Cup '99 dataset were selected. Only 10% of the data was used as a training set of about 49,402 data samples out of 494,021 total data samples. The detection accuracy was calculated on the selected dataset, and a high detection accuracy was reached, given that only 10% of the data was used for training. The detection accuracy results are introduced in Figure 16. The feature space was reduced by selecting the most significant features. The dataset consists of 41 features; the unsupervised learning algorithm selected only 15 features, the supervised learning algorithm selected 14 features, and 22 features were selected after applying the enhancements for the J48 decision tree.

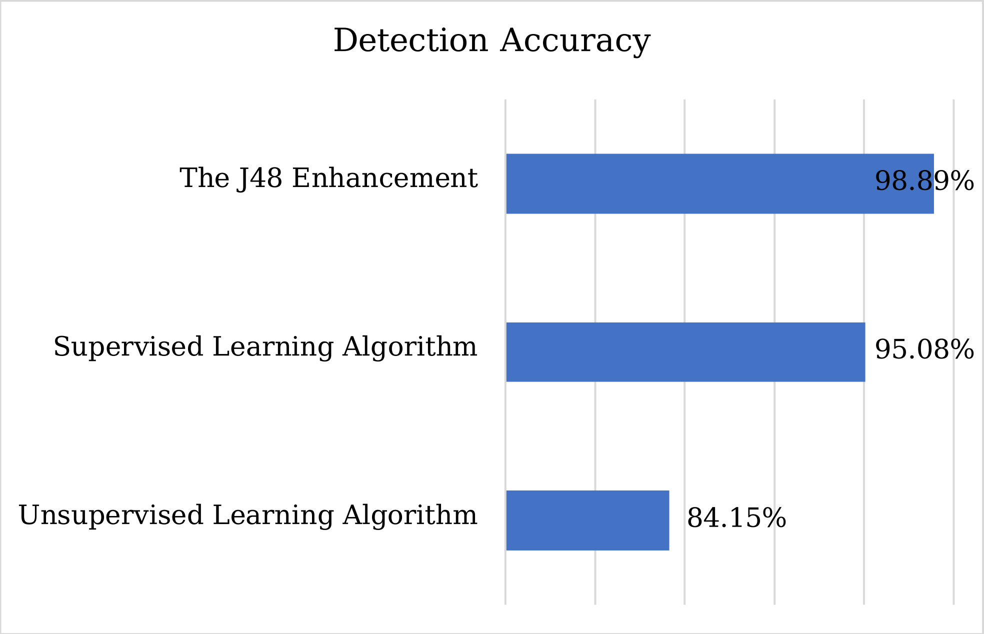

**Figure 16.** Detection accuracy of the proposed algorithms

## X. Detection Accuracy of the Algorithms on Activities of Daily Living (ADLs) Recognition dataset

In this dataset, the daily activities of two users are monitored via sensors. The dataset contains the data of 35 days of the user's daily activities and has nine labels. A user's dataset is used in the experiments using 50% of the data for training and 50% for testing [42]. On applying the algorithms to the ADL dataset, convenient classification accuracy was reached compared to the CART algorithm, given that 50% of the data was used for training and the dataset contains 9 labels. The classification accuracy results were compared to the CART algorithm and are introduced in Figure 17. The results show that the sensor node algorithm gave higher classification accuracy than the cart algorithm. The accuracy of the algorithms, given that the dataset is not binary labeled (as it contains 9 labels) indicates that the algorithms can be used on data of more than two labels and still be able to give excellent accuracy compared to other decision tree-based algorithms.

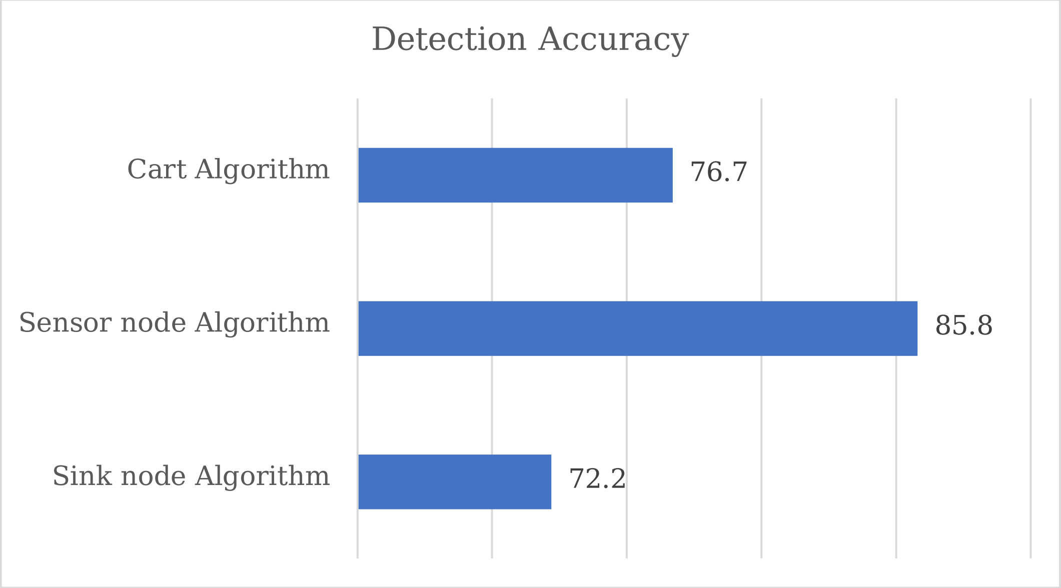

**Figure 17.** Detection accuracy of the algorithms of the sink node and the sensor node on ADLs dataset compared to cart algorithm.

## XI. Detection Accuracy of the Algorithms on 3 Labeled Wireless Sensor Network Data Repository (LWSNDR)

The data consists of humidity and temperature measurements collected for 6 hours at intervals of 5 seconds. Single-hop data is collected on 9th May 2010, and the multi-hop data is collected on 10th July 2010. Label '0' denotes normal data, and label '1' denotes an introduced event [43]. 50% of the data of one sensor was used as training data in the experiments; testing was done on the other 50% (for sensor 1), and testing was also done on the whole data of another sensor (sensor 2). The classification accuracy results were compared to the CART algorithm and are introduced in Figures 18 and 19.

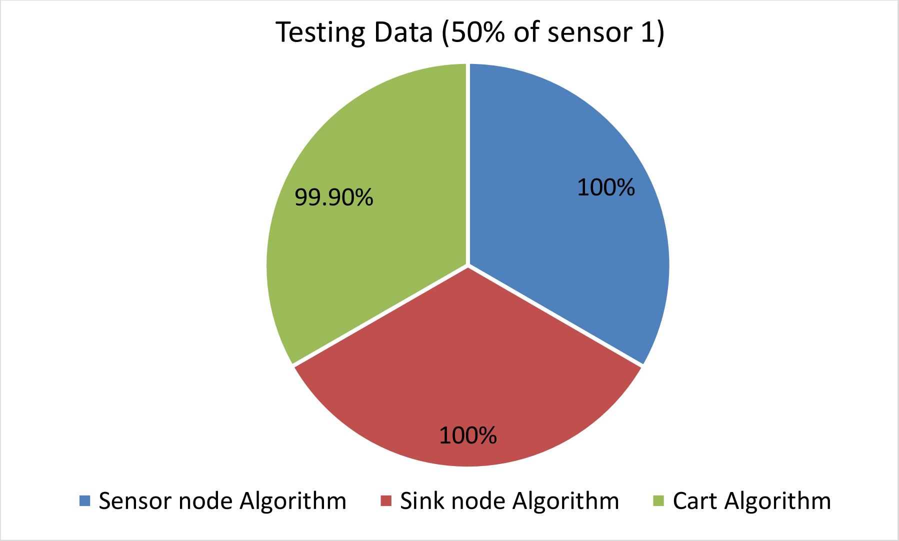

**Figure 18.** Detection accuracy of the algorithms of the sink node and the sensor node on lwsndr dataset

compared to cart algorithm with 50% of sensor 1

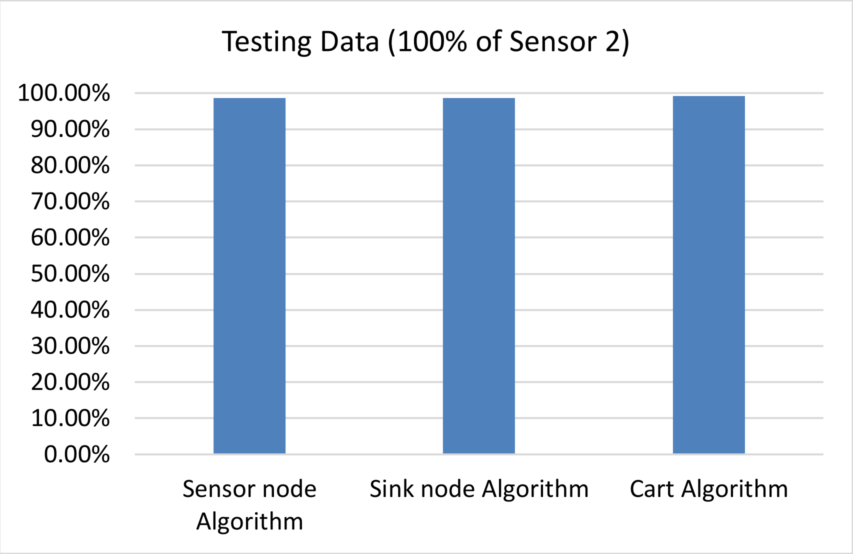

**Figure 19.** Detection accuracy of the algorithms of the sink node and the sensor node on lwsndr dataset compared to cart algorithm with 100% of sensor 2

## XII. Conclusion

Two intrusion detection algorithms were proposed in this work, one uses a supervised learning mechanism, and the other uses an unsupervised learning mechanism. The output of each of the algorithms is a set of detection rules structured in the form of a binary decision tree. The learning algorithms are trained, and the decision trees are built before the network's operation. Then the decision trees are loaded to the sensor nodes to detect intrusions during the network's operation. The intrusion detection algorithms were used in two different network architectures. The first architecture represents the level of the sensor node, sink node, and base station. Sensor and sink nodes are the second architecture level. This paper presented two intrusion detection algorithms, namely supervised intrusion detection and unsupervised intrusion detection. The network architectures were set to monitor the differences between the numbers of the generated intrusion data packets for each architecture. The introduced algorithms provided a high detection accuracy compared to the decision-tree-based cart algorithm using fewer selected features (where the most relevant <feature, value> pairs were selected using Entropy and pruning of the decision tree in the training phase), compared to previous work for feature selection. The introduced algorithms' selected features provided competitive results on different algorithms and gave higher detection accuracy for neural networks and support vector machine-based algorithms. The proposed learning algorithms used only 10% of the data for training for KDD dataset, 50% of the data for training for the ADL dataset and 25% of the data for training for LWSNDR, and recognizable detection/classification accuracy was achieved. Simplification for CART classification algorithm is also introduced, which decreases the algorithm's decision tree size and makes it suitable for intrusion detection in WSNs.

The future work may involve algorithms for higher detection accuracy on the two-layer network architecture. Some modifications on the algorithms for higher detection accuracy or an enhanced feature selection criterion may also be considered. Data aggregation for reporting the intrusions detected may be studied to decrease energy consumption and decrease intrusion data packets sent across the network. The actions to be taken to detect an intrusion may be analyzed based on the application of the WSN. Techniques for decreasing the complexity and decreasing the energy consumption may be studied. The detection of the failed nodes may also be considered in the network applying the proposed model.

## XIII. References

[1]. M. Carlos-Mancilla, E. López-Mellado, and M. Siller. (2016). Wireless Sensor Networks Formation: Approaches and Techniques.

[2]. S. R. J. Ramson and D. J. Moni, "Applications of wireless sensor networks — A survey," presented at the 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), Coimbatore, India, 2017.

[3]. V. Rocha, "Sensing the World - Challenges on WSNs," presented at the IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics, Romania, 2008.

[4]. E. P. K. Gilbert, B. Kaliaperumal, and E. B. Rajsingh, "Research Issues in Wireless Sensor Network Applications: A Survey," International Journal of Information and Electronics Engineering, vol. 2, 2012.

[5]. N. Srivastava, "Challenges of Next-Generation Wireless Sensor Networks and its impact on Society," Journal of Telecommunications, vol. 1, 2010.

[6]. J. A. Stankovic, "Research challenges for wireless sensor networks," in ACM SIGBED Review - Special issue on embedded sensor networks and wireless computing vol. 1, ed: ACM New York, NY, USA, 2004.

[7]. H. Chawla, "Some issues and challenges of Wireless Sensor Networks," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, 2014

[8]. Gowrishankar.S, T.G.Basavaraju, M. D.H, and S. K. Sarkar, "Issues in Wireless Sensor Networks," in World Congress on Engineering, London, U.K., 2008.

[9]. J. P. Anderson, Computer Security Threat Monitoring and Surveillance. James P Anderson Company, Fort Washington, Pennsylvania, April 1980.

[10]. U. A. Sandhu, S. Haider, S. Naseer, and O. U. Ateeb, "A Survey of Intrusion Detection & Prevention Techniques " presented at the 2011 International Conference on Information Communication and Management, Singapore, 2011.

[11]. H. A. M. Uppal, M. Javed, and M. J. Arshad, "An Overview of Intrusion Detection System (IDS) along with its Commonly Used Techniques and Classifications," International Journal of Computer Science and Telecommunications, vol. 5, pp. 20 - 24, February 2014 2014.

[12]. M. Al-Subaie and M. Zulkernine, "Efficacy of Hidden Markov Models Over Neural Networks in Anomaly Intrusion Detection," presented at the Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International, Chicago, IL.

[13]. K. Ilgun, R. A. Kemmerer, Fellow, IEEE, and P. A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. XX, 1995.

[14]. Grzech, "Intelligent Distributed Intrusion Detection Systems of Computer Communication Systems," presented at the Asian Conference on, Intelligent Information and Database Systems, Quang Binh University, Dong Hoi City, Vietnam, 2009.

[15]. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," IEEE Wireless Communications, vol. 14, pp. 56–63, 2007.

[16]. 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, pp. 1636-1652, 2012.

[17]. J. S. Parihar, J. S. Rathore, and K. Burse, "Agent Based Intrusion Detection System to Find Layers Attacks," presented at the 2014 Fourth International Conference on Communication Systems and Network Technologies Bhopal, India April 2014

[18]. Patel, Q. Qassim, and C. Wills, "A survey of intrusion detection and prevention systems," Information Management & Computer Security, vol. 18, pp. 277 - 290, 2010.

[19]. L. Chen and G. Gong, Communication System Security, Chapman and Hall/CRC, 2012.

[20]. F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, B. Wiedersheim, et al. (2008) Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges. Communications Magazine, IEEE 110 - 118.

[21]. T. Turek, S.-A. Zerawa, and T. Anees, "Towards safety and security critical communication systems based on SOA paradigm," presented at the 2011 IEEE 20th International Symposium on Industrial Electronics (ISIE), Gdansk University of Technology, Gdansk, Poland, 2011.

[22]. D. Ajenjo and H. Wietgrefe, "Minimal-Intrusion Traffic Monitoring And Analysis In Mission-Critical Communication Networks " Journal of Systemics, Cybernetics and Informatics vol. 1, 2003.

[23]. G. Kumar and K. Kumar, "Design of an Evolutionary Approach for Intrusion Detection," The Scientific World Journal, vol. 2013, pp. 1 -14, 2013.

[24]. C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian Event Classification for Intrusion Detection," in 19th Annual Computer Security Applications Conference, Las Vegas, NV, USA, 2003.

[25]. Krontiris, T. Dimitriou, and F. C. Freiling, "Towards Intrusion Detection in Wireless Sensor Networks," presented at the 13th European Wireless Conference, Paris, France, 2007.

[26]. P. R. d. Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," presented at the Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks,ACM., 2005.

[27]. U. Ravale, N. Marathe, and P. Padiya, "Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K-Means and RBF Kernel Function," presented at the International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), Mumbai, India, 2015.

[28]. A. Karan Bajaj, "Dimension Reduction in Intrusion Detection Features Using Discriminative Machine Learning Approach " IJCSI International Journal of Computer Science Issues, vol. 10, pp. 324 - 328, 2013.

[29]. N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision Tree Analysis on J48 Algorithm for Data Mining," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, pp. 1114-1119, June 2013.

[30]. P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," Machine Learning vol. 29, pp. 103-130 1997-11 1997.

[31]. C. D. Manning, P. Raghavan, and H. Schütze, "Naive Bayes text classification," in Introduction to Information Retrieval, C. D. Manning, P. Raghavan, and H. Schütze, Eds., ed: Cambridge University Press. 2008, 2009.

[32]. M. J. Fonseca and J. A. Jorge, "NB-Tree: An Indexing Structure for Content-Based Retrieval in Large Databases," INESC-ID2003b.

[33]. S. JK and V. J., "Training Multilayer Perceptron Classifiers Based on a Modified Support Vector Method," IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 10, July 1999 1999.

[34]. C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning vol. 20, 1995-09 1995.

[35]. S. Kalmegh, "Analysis of WEKA Data Mining Algorithm REPTree, SimpleCart and RandomTree for Classification of Indian News " IJISET - International Journal of Innovative Science, Engineering & Technology, vol. 2, February 2015 2015.

[36]. R. Zhang and X. Xiao, "Intrusion Detection in Wireless Sensor Networks with an Improved NSA Based on Space Division," Journal of Sensors, vol. 2019, p. 20, 2019.

[37]. K. Medhat, R. A. Ramadan, and I. Talkhan, "Distributed Intrusion Detection System for Wireless Sensor Networks " presented at the 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, Cambridge, UK, 2015.

[38]. J. Wu, S. Liu, Z. Zhou, and M. Zhan, "Toward Intelligent Intrusion Prediction forWireless Sensor Networks Using Three-Layer Brain-Like Learning," International Journal of Distributed Sensor Networks, vol. 2012, pp. 1- 14, 2012.

[39]. P. Bellot and M. El-Bèze, "Clustering by means of Unsupervised Decision Trees or Hierarchical and K-means-like Algorithm," in RIAO'2000 Conference Proceedings - Collège de France, Paris, France, 2000, pp. 344-363.

[40]. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations, vol. 11, 2009.

[41]. E. B. Mahbod Tavallaee, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in CISDA'09 Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, 2009, pp. 53-58.

[42]. F. J. Ordóñez, P. d. Toledo, and A. Sanchis, "Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors," Sensors, vol. 13, 2013.

[43]. S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled Data Collection for Anomaly Detection in Wireless Sensor Networks," in Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2010), Brisbane, Australia, 2010.