Intrusion Detection Using Rule Based Approach in RPL Networks

Balachandra Muniyal (bala.chandra@manipal.edu)
Manipal Academy of Higher Education
Manjula C Belavagi
Manipal Academy of Higher Education

Research Article

Keywords: Intrusion Detection, Machine Learning, Rule Based Approach, RPL Network Security

Posted Date: April 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1525359/v1

Intrusion Detection Using Rule Based Approach in RPL Networks

Manjula C Belavagi[1] and Balachandra Muniyal[1*]

[1*]Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, 576104, Karnataka, India.

*Corresponding author(s). E-mail(s): bala.chandra@manipal.edu;
Contributing authors: manjula.cb@manipal.edu;

Abstract

To satisfy the growing need of wireless sensor networks in areas of defence, Internet of Things, health care, environmental monitoring, and so on., IETF has proposed a new Routing Protocol for Low power and Lossy networks which works with IPv6. The nodes of these network are placed in vulnerable environments and critical,sensitive information is transferred between them based on the application. Hence, the security of such a network is very important. Intrusion Detection System plays an important role in providing security to such types of networks, which is computationally costly owing to the limited resources of sensor nodes. By considering the capabilities of wireless sensor networks, an intrusion detection model is designed using Logistic Regression, Gaussian Naive Bayes, Artificial Neural Networks, Support Vector Machine and Random Forest and analyzed on IEEE-IoT-IDS, WSN-DS, and simulated data. Based on the analysis, suitable machine learning algorithm is selected for rule generation. Later, multiple attacks are identified using Rule Based Approach (RBA). For the efficient utilization of the sensor node′s energy, the rule based algorithm executes at the base station. Experimental results show that the proposed method gives good results in the identification of multiple intrusions.

Keywords: Intrusion Detection, Machine Learning, Rule Based Approach, RPL Network Security

1 Introduction

Tiny nodes with low power and sensing capability are the main components of the wireless sensor network [1]. This network technology has become an important research area due to its upcoming applications. These include military, Internet of Things[2], health [3], business and numerous other important applications [4]. These applications need interconnection of billions of such tiny nodes. Hence Internet Protocol(IP) version 6 (IPv6) address space is used instead of IPv4 with new routing protocol 'Routing Protocol for Low Power and Lossy Networks' (RPL)[5],[6].

RPL is designed for low power and lossy networks, namely, Wireless Sensor Networks (WSN). Nodes of this network have sensing capability and are used to observe the physical and environmental situations. These nodes form a topology by organizing themselves, which is Directed Acyclic Graph(DAG). It is partitioned into one or more Destination Oriented Directed Acyclic Graph (DODAG) having single root without any outgoing edges Routing within the RPL network depends on node metrics and link metrics [5], [7]. Node metrics depend on hop count, node state attribute and node energy, whereas link metrics depend on the latency, link quality level, throughput, and Expected Number of Transmissions (ETX). RPL supports three types of traffic, namely, Multi Point to Point(MP2P), Point to Point(P2P) and Point to Multi Point(P2MP).

Routing metrics, hop count and ETX are the main parameters for the topology construction. RPL network topology [8] construction depends on the objective functions (1) Objective Function 0(OF0) and (2) Minimum Rank Objective Function with Hysteresis (MRHOF) / ETX The DODAG construction also depends on three control messages:

1. Destination Advertisement Object (DAO) - Forwards routing information about destination towards the root (unicast)
2. DODAG Information Object (DIO) - Identifies the RPL instance (multicast).
3. DODAG Information Solicitation (DIS) - Used by the node after joining the network(multicast).

Neighboring nodes and DIO messages help a node to select a preferred parent. A child node communicates using DAO messages with the parent node. Energy used during the broadcasting DIO messages can be saved by sending the explicit message [9].

Number of control messages communicated are controlled by the Trickle-timer. This is taken care by the Trickle-algorithm. DIO-INTERVAL-MIN and DIO-DOUBLINGS are the main parameters used by this algorithm. DIO-INTERVAL-MIN is used as the interval of control packet transmission, and DIO-DOUBLINGS is used to place an upper limit, on the rate of this transmission.

Rank of a node represents the position of every node relative to other nodes of the network with respect to "root node" of the DODAG. Nodes which are

on the same level have the same rank value. A node which is near the sink has low rank value than the node which is far from the sink node.

Depending on the applications WSN, nodes are placed in non-secure environments. So, these nodes are vulnerable to security challenges such as routing attacks, Denial of Service (DoS) attacks and Sybil attack, etc. Hence there is a need for an Intrusion Detection System (IDS). A Security mechanism used to monitor the abnormal behavior of the WSN's is an IDS. Actions that violate confidentiality, availability and integrity of information and resources are called intrusion. Due to resource constraints of sensor nodes, Key-management techniques, security protocols and authentication techniques [10] are not applicable to this scenario. Currently limited research available on multiple intrusion detection on RPL based WSN. Hence, the main focus of the paper is to design and implement intrusion detection model using machine learning and rule based approach. This paper focuses on stage-wise intrusion detection using machine learning and rule based approach. Following are the key contributions of this paper:

- Simulate the network with malicious activity by considering the MRHOF function.
- Build the machine learning model on primary and secondary data for multiple attack detection.
- Generate the rules using suitable model to identify the multiple attacks (Rank, DoS and Selective forwarding).
- Analyze the energy savings with rule based approach.

The paper is organized as follows: In Section 2 the recent developments in intrusion detection system is discussed. Section 3 describes overall methodology followed to build intrusion detection model. Result analysis is carried out in Section 4. Finally, the conclusion of the work is presented in Section 5.

2 Literature Review

In this section, literature related to intrusion detection in WSN has been discussed.

Attacks identification in 6LoWPAN is proposed by Le et al[11]. They used specification based technique, in which simulation of the network is carried out using Cooja simulator. They have observed the deviation in behaviour of each node to identify the malicious activities. They have concluded that the proposed method has good accuracy, whereas it causes overhead in case of large networks. The network traffic classification on real time data traffic is proposed by Jun et al [12]. Unsupervised machine learning approach is used to detect application based network traffic. Internet Protocol payload and some statistical properties are used as parameters. Content of the clusters are represented using bag of word model. They suggested that payload contents can be used to categorize similar traffic.

Packet fragmentation based intrusion detection in 6LowPAN networks was proposed by Hummen et al [13]. They have considered fragment duplication attack and buffer reservation attack. The cost of detection is less, whereas detection rate is moderate. Anomaly based method in wireless clusters architecture was proposed by Yassine et al [14]. They have experimented using Support Vector Machine (SVM) with the assumption that cluster head is a known node which sends the packets to sink node. The paper depicts detection rate as high and has low false positive rate. In RPL networks sinkhole defence mechanisms are evaluated based on rank verification and parent fail over techniques [16]. Results show that the combination of above mentioned methods can be used to improve the performance.

Mahalakshmi et al. [17] have proposed genetic algorithm based IDS to identify DoS attacks in WSN. They used Modified RSA (MRSA) algorithm for the generation of pair of keys among the sensor nodes. Before the transmission of packets, the optimal path for the communication is identified by Adhoc On-demand Distance Vector routing (AODV). They also performed fitness calculations to identify the reliability of relay nodes. Behavior of the attacker nodes is identified by cross and mutation techniques. During the determination of the attacker nodes, there is no communication between the base station and the remaining network. The paper not only focuses on DoS attacks on simulated data but also on data cleaning and selection of features. However,the number of features considered is not discussed.

Wang et al. [18] and Rebello et al. [19] have discussed the importance of the intrusion detection model in WSN and how to take care of empowered intruders. They have also done a comparative study on different attacks on WSN and the suitable technique to handle such attacks. The importance of pre-processing of network traffic data for intrusion detection and steps that need to be followed for the pre-processing of the data is presented in [20] [21] [22]. The pre-processing of network traffic data to identify the anomalies is presented by Jonathan et al. [23] and Chen et al. [24]. Identification of network traffic features based on packet headers, protocol information, and the payload is discussed. They also mentioned cleaning, transformation, and relabeling of network data.

Singh et al. [25] proposed an IDS which detects the intrusions automatically. The model is based on cluster architecture and advanced LEACH protocol to reduce the energy consumption of the nodes. Fuzzy rules and neural networks with multi-layer perceptron are used for anomaly and misuse based detection. Sreenivas et al. [26] proposed ETX and Rank based IDS modules in 6LowPAN networks. They make use of geographic locations to identify the malicious nodes. Both the models show good detection rate. However, the network size considered is very small.

IDS using data mining classification techniques such as Decision Tree, Naive Bayes, Random Forest, Neural Networks and, Support Vector Machine is

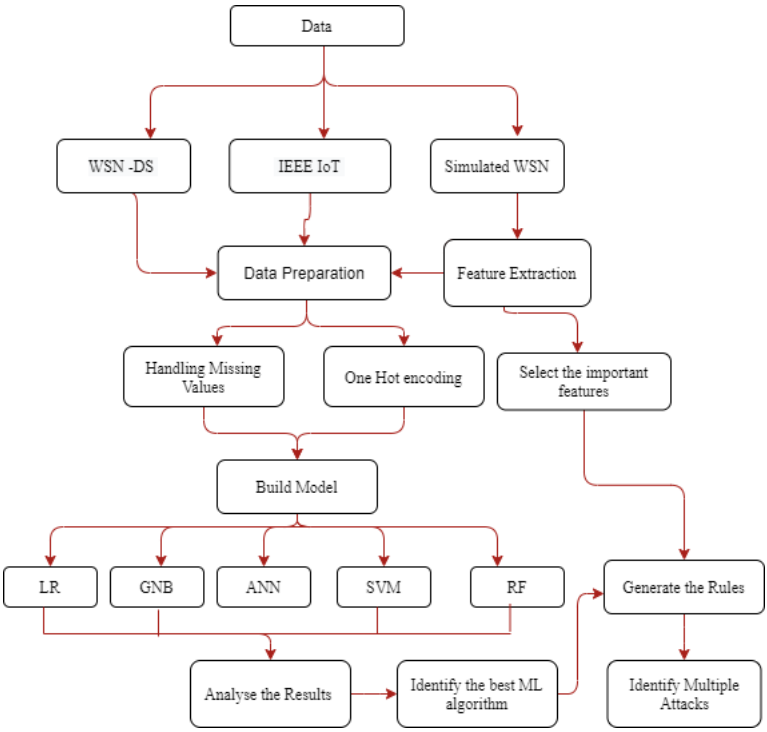

Fig. 1 Methodology for Rule Based IDS

proposed in [27] [28]. The authors tested the data mining algorithms on KDD-CUP99 dataset. In [27] authors proposed an algorithm to handle insufficient labeled data. Different validation techniques are used in [28].

3 Methodology

Methodology for building ML-based prediction models for intrusion detection is shown in Figure 1. The data pre-processing is an important step in intrusion detection before building the ML models. The data-set is made suitable for ML models by taking care of missing values and categorical values. Feature selection is also necessary as part of data preparation if the data set has a large number of attributes. Hence, feature selection is carried out only for the simulated data. Once the data is ready, different ML models are built and a suitable ML model is identified by analyzing the results. The selected model is used for rule generation to identify multiple attacks.

3.1 DataSets Used

The data is considered from two different sources: primary and secondary. The IEEE-IoT-IDS and WSN-DS are considered as secondary datasets and simulated WSN is considered as a primary dataset for the research. To understand

the behavior of the traffic in the WSN, the secondary datasets are studied and analyzed. Based on the analysis, the WSN data is simulated for various other attacks using Cooja simulator on Contiki operating system. The main reason behind choosing these two types of datasets is to focus more on understanding different malicious behavior and attacks. The primary data is generated using simulation of WSN scenario which is used to build the machine learning models. The secondary datasets are considered for experimenting the ML models to avoid biasing in the dataset with respect to simulated attacks. The IEEE-IoT-IDS and WSN-DS datasets are based on RPL networks, and deals with multiple attacks which are simulated in the primary dataset.

- IEEE-IoT-IDS Dataset:
  This dataset is developed by Avast AIC laboratory [29]. It has attack wise pcap files and benign pcap files taken on different dates. It includes 23 features and has different attack types. The features of the dataset include the time of capture, Identification number and IP address of the node in each pcap file, port number, payload, flags, protocols(TCP, UDP, MAC), service, duration, amount of data sent and received, label, frame length, dns details (count,flags,id) and detailed description of the label. Table 3.1 shows the details of the dataset.

Table 1 Description of IEEE-IoT-IDS Data

Normal/Attack	Number of Records
Normal	30858735
C&C	21995
C&C-FileDownload	53
C&C-HeartBeat	33673
C&C-HeartBeat-Attack	834
C&C-HeartBeat-FileDownload	11
C&C-Mirai	2
C&C-PartOfAHorizontalPortScan	888
C&C-Torii	30
DDoS	19538713
FileDownload	18
Okiru	47381241
Okiru-Attack	13609470
PartOfAHorizontalPortScan	213852924
PartOfAHorizontalPortScan-Attack	5
Attack	9398

- WSN-DS Dataset:
  This dataset is developed by Iman et al [30]. Authors have simulated four types of DoS attacks namely, gray-hole, black-hole, flooding and scheduling attacks. This dataset has 19 features and is based on LEACH protocol. It has features related to identification of the nodes (nodeID, cluster head), energy consumption (current energy, energy consumed), messages sent and

received by the cluster head (advertise with a broad cast), data messages, node's rank, join request message and attack types. The description of the dataset is shown in Table 2.

Table 2 Description of WSN-DS Dataset

Attack-Type	Number of Records
Normal	340066
TDMA	6638
Grayscale	14596
Flooding	3313
Blackhole	10049

- Simulated Data:
  Simulation is carried out by considering network architecture having three types of nodes namely sink (BS), cluster heads, and sensor nodes. The cluster head communicates with the sink and the sensor nodes. It acts as an intermediate node between these two. The following assumptions are made for the proposed network architecture.
  - Type 1 - Energy constrained sensor nodes that perform the simple task of data gathering (sensing) and transmitting the data.
  - Type 2 - Control unit named as IDS agent which has a higher transmission range, performs monitoring of nodes in addition to data gathering.
  - Type 3 - Base station to generate rules and data analysis.

  IDS Agent is installed on CH to monitor all the sensor nodes and communications of the network. It also interacts with the BS.
  Proposed IDS deployment architecture in WSN is shown in Figure 2. It shows communications between IDS Agent and Base Station. IDS agent monitors the group of sensor nodes using set of specification rules. Initially these rules are generated by Random Forest Classifier based on the historical data at the base station and are updated periodically and send to the IDS agent. At the base station, actual prediction is carried out using a predictive algorithm to identify whether the node is malicious or not. The prediction model is also trained periodically to incorporate new behavior of malicious nodes.
  WSN traffic data with normal and malicious activity is simulated using a Cooja simulator by considering MRHOF as an objective function. The simulation parameters are shown in Table 3. Three attacks namely Rank, Selective forwarding and DoS attacks are simulated. The best suitable parameters are identified after performing experimentation [31]. The ratio of the transmission is set to 100%. This indicates that losses are introduced at the receiver end but not at the transmitter end. The "Ratio of Packet Reception" is set in percentages during the successive iterations of the simulation, and represents how lossy is the radio medium. Parameters Minimum

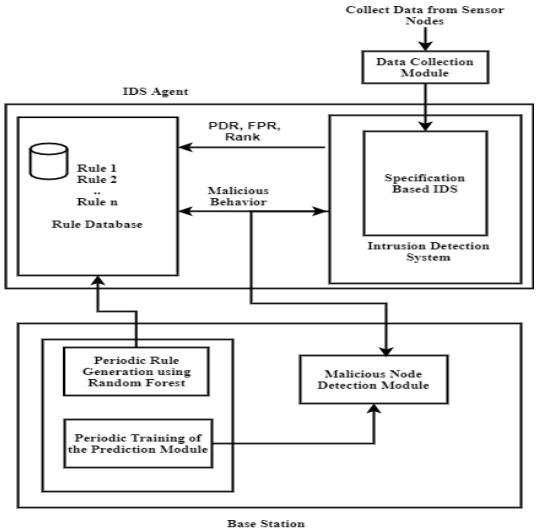

Fig. 2 Deployment Architecture of IDS

DIO Interval and DIO-Doublings are set to default values of Contiki. The range of transmission is set to 50m and the interference range is set to 55m and the objective function considered is MRHOF. The mode of operation of the RPL network is set to "NO-DOWNWARD-ROUTE" because multi-point to point network traffic is required for our experiment, stating data collection from different nodes of the network. Simulation is carried out for 50 nodes, by varying the "Ratio of packet reception" in the range of 30% - 100% for every 15 minutes.

It has protocol specific features such as ICMPv6, UDP, IPv6, topology related features, data communicated details, flags, and other time related features. The detailed description of the data is shown in Table 3.1. Total records of the dataset are 390230. Among these, 362521 records represent the normal activities and the remaining were the malicious data records.

3.2 Data Pre-processing

The data-sets considered and the simulated data are raw, hence these need pre-processing to convert them into the form which is ready for applying ML models. Data pre-processing on IEEE-IoT-IDS dataset, WSN-DS and simulated data is carried out. The pre-processing is carried out by handling missing values and one hot encoding. The technique used is replacing the missing values with zero as the features are protocol specific. A new column is added for each value of categorical attribute using one hot encoding technique.

Data exploration helps to identify the relationship between the variables, to understand the data before building analytical models over that data. This analysis can be performed individual attribute wise or by combining two or

Table 3 Simulation Parameters

Parameters	Value
Objective Function used	MRHOF
Ratio of Transmission(%)	100
Ratio of Packet Reception(%)	30 - 100
Range of Transmission (m)	50
Range of Interference (m)	55
Simulation Duration	For each ratio of packet reception -15 minutes
Number of client nodes	50
Minimum DIO Interval(ms)	12
DIO Doublings (ms)	8
Mode of Operation - RPL	NO-DOWNWARD-ROUTE

Table 4 Description of Simulated Data

Type of Traffic	Number of Records
Normal	362521
Rank Attack	9325
Selective Forwarding Attack	9325
DoS	9059

more attributes. Exploratory analysis is carried out by identifying the correlation between the attributes, pair wise relationship between the attributes, number of control messages communicated and

3.3 Feature Selection

Simulated data has more than 150 features. From these features, few extra features are also generated such as Packet Drop Rate (PDRR), Duplicate Packet Rate (DPR) and Packet Forward Rate(PFR), average Source Packet size(avg-src-pkt-sz), average source packets per second(avg-src-pkt-ps), average destination packets per second(avg-des-pk-ps), average source bytes per second (avg-src-bytes-ps), percentage of DIO, DAO and DIS messages communicated, cluster head and data sent/received from cluster head.

Features must be highly correlated with the label and should have a low correlation with the other attributes. If two independent attributes are highly correlated, one which is highly correlated with the label can be retained and other can be removed. The correlation factor is computed using Information Gain. It depends on entropy and is computed as follows:

Entropy: It is a measure of uncertainty or disorder of a feature. It is computed using Equation 1.

$$H(A) = -\sum P(A_i) * log_2(P(A_i)) \tag{1}$$

Table 5 Selected Features

Sl. No.	Feature Name	Sl. No.	Feature Name
1	icmpv6.type	14	wpan.seq-no
2	icmpv6.code	15	ipv6.src
3	icmpv6.checksum.status	16	ipv6.addr
4	icmpv6.rpl.dio.instance	17	udp.srcport
5	icmpv6.rpl.dio.version	18	udp.dstport
6	icmpv6.rpl.dio.rank	19	udp.length
7	icmpv6.rpl.dio.flag	20	wpan.frame-type
8	icmpv6.rpl.dio.dtsn	21	ipv6.opt.rpl.flag
9	icmpv6.rpl.dio.dagid	22	icmpv6.rpl.dao.instance
10	icmpv6.rpl.opt.length	23	icmpv6.rpl.dao.sequence
11	icmpv6.rpl.opt.metric.type	24	icmpv6.rpl.dao.dodagid
12	frame.len	24	icmpv6.rpl.opt.target.flag
13	wpan.seq-no	.	.

Information Gain: It is knowledge gain obtained for feature A if feature B is already known. Information gain is computed using Equation 2.

$$IG(A/B) = H(A) - H(A/B) \tag{2}$$

Based on Information Gain, selected features are shown in Table 5. Selected features are based on different protocols such as IPv6, UDP and ICMPv6. Features related to ICMPv6 are related to control traffic. UDP related fields are the source and destination port, length and data. Some of the fields are frame related such as type of frame and sequence number. The other generated features selected are PDR, DPR, PFR, avg-src-pkt-sz, avg-src-pkt-ps, avg-des-pk-ps, avg-src-bytes-ps, percentage of DIO, DAO and DIS messages communicated. These features are used by the ML models to identify the intrusions. The main aim is to identify the suitable ML model for a generation of rules to identify multiple intrusions. Machine learning models are built using Logistic Regression (LR), Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Artificial Neural Networks (ANN) and Random Forest (RF) classifiers to test the standard recent intrusion detection data-sets [30] [29] and simulated data. The data is pre-processed by handling missing values and categorical values. Data is divided into training(80%) and testing (40%) data. A suitable model is identified by analyzing the evaluation metrics of the ML models. The ML models used are explained below:

- Logistic regression: The hypothesis function in Logistic Regression relates exogenous variables which are p selected features from the datasets (IEEE-IoT-IDS, WSN-DS, and simulated data) and outcome variable which is

attack or normal. The model returns an estimated probability score of predicting the any of the attack type or normal. This probability score is given as input to the Sigmoid activation function. If the input probability value is zero or positive, then the prediction will be a value greater than or equal to 0.5, which is approximated to 1 for attack class. A negative input probability value returns value less than 0.5, which is approximated to 0 for normal.

- Gaussian Naive Bayes: The Gaussian Naive Bayes (GNB) algorithm is a supervised learning method. It uses the probabilities of each attribute belonging to each class to make a prediction. The algorithm works based on the strong assumption that the probability of each attribute belonging to a given class value is independent of all other attributes. The probability of a class value given a value of an attribute is called conditional probability. The likelihood probability of a data instance belonging to a specific class can be computed by multiplying the conditional probabilities together for each attribute for a given class value. Prediction can be made by calculating the probabilities of the instances belonging to each class and selecting the class value with the highest probability [32]. GNB uses categorical as well as numeric data and assumes that the attributes are normally distributed.
  WSN traffic data is given as input to the GNB classifier. Prior probabilities of each feature are obtained using training data. Using these features, the probability of evidence is computed. Later, likelihood probability for each class, namely normal, rank, SF and DoS is computed. Based on these values, labels of the test samples are predicted.
- Support Vector Machine: Support vectors are data points closer to the plane and influence the orientation along with position of the hyperplane [33]. Hyperplane for the attack prediction is built such that it separates the samples, which are labelled as an attack or normal. The algorithm is designed to compute support vectors using linear kernel function, which clearly transforms the feature space and segregate the two classes of the training sample. Hyperplane represented by weights W and bias b, and Support Vectors SV are the model parameters. To compute local support vectors initial weights and bias are randomly assigned. For these weights and bias value, initial hyperplane and boundary planes are computed. Local support vectors are the actual data points present on boundary planes. Based on the predictions considering the hyperplane, weights are updated. Then using the model parameters the predictions are computed. The predictions are any attack and normal.
  SVM's linear kernel function is used for intrusion detection. Model parameters considered are weights W and bias b, and Support Vectors SV for the hyperplane. Initial weights and bias are randomly assigned. For these values, initial hyperplane and boundary planes are computed. Local support vectors are the actual data points present on boundary planes. Weights are updated based on these predictions with respect to the hyperplane.
- Random Forest: Random Forest is a collection of decision trees. Each decision tree predicts the outcome as Rank, Selective forwarding or DoS

attack.These decisions are combined using majority voting by counting the decisions. The random forest is based on the standard machine learning technique called decision tree which, in ensemble terms, corresponds to the weak learner. In the case of a decision tree, input data is given at the top node and the data navigate down the sub-trees [34]. After reading the WSN data with 34 features 'k' features are randomly selected. Best split point 'd'is computed among these features, which divides nodes into child nodes. This process is repeated to create 'n' number of trees. Final prediction depends on the decision of every node. The predicted output having the highest number of votes is considered as the final prediction

- Artificial Neural Networks: Neurons are the most essential processing units of the brain which are billions in number. They process information in parallel to generate an output. Each input is associated with synaptic weights. The cluster of neurons that function together to process the information is neural networks. Artificial Neural Networks (ANN) are processing models which have artificial neurons in multiple layers. The three main layers of ANN are the input layer, hidden layer and output layer. Each layer consists of one or more nodes (neurons). All layers are connected and information flows from the input to the output layer. Some ANN architectures support feedback flow, which is used to improve the results by correcting the errors [35]. ANNs are efficiently used to identify network intrusions [36]. The number of nodes in the hidden layer is computed with the equation 3. Input layer with 23 nodes, hidden layer with 20 nodes and an output layer with p nodes, where p is the number of attacks considered for the implementation for the dataset IEEE-IoT-IDS. Similarly for the WSN-DS dataset input layer with 19 nodes, hidden nodes with 15 nodes and an output layer with p nodes. For the simulated data, Input layer with 34 nodes, hidden layer with 25 nodes and a output layer with p nodes, p no of attacks is considered.

$$no - of - nodes = \frac{(Inputnodes + Outputnodes) * 2}{3} \tag{3}$$

The data is iteratively fed to the input nodes, computations are carried out through hidden layers and the result is obtained in the output layer node. This output is compared with the actual labels to compute potential error. This computed error is fed back to the model in each iteration. Based on the diminishing error and desired output, the weights are adjusted and final model is obtained. The Sigmoid function is used to compute the intermediate values on the weighted sum of all the inputs given to the neuron. The intermediate values provided by the last hidden layer are used to compute the predictions at the output layer. Using these predictions and actual output, error at the output layer is computed. This error is back-propagated to adjust weights to reduce the error. The classification process continues with the updated weights. This process is repeated till the error reaches the threshold value. Finally, returns the predicted output of the test data.

4 Result Analysis

This section gives insight into the results of the supervised ML algorithms namely LR, GNB, ANN, SVM and RF on IEEE-IoT-IDS, WSN-DS, and simulated data. Multiple attacks are also identified using the generated rules.

The performance of these algorithms is analyzed based on different metrics. These metrics are based on confusion matrix which is shown in Table 4. Performance metrics used for the analysis of ML algorithms are shown in

Table 6 Confusion Matrix

Actual \ Predicted	Malicious	Non-Malicious
Malicious	TP	FN
Non-Malicious	FP	TN

Equations 4, 5, 6, 7, and 8. Terminologies used in these equations are as follows: *TP* is True Positive, *FN* is False Negative, *TN* is True Negative, *FP* is False Positive, and *DR* is Detection Rate. *n-classes* indicates number of classes, and value of j varies from 1 to *n-classes*. The ML models are evaluated based on Precision_{macro}, Recall_{macro} F1_{macro}, weighted (precision), weighted (recall), weighted (F1), and accuracy. For multi class classification macro metrics and weighted average are suitable. In case of macro metrics the values are computed independently for each class and then the average of these is taken, whereas weighted metrics compute the average by considering each class size.

$$Precision_{macro} = \frac{1}{n - classes} \sum_{j=1}^{nclasses} \frac{TP_j}{TP_j + FP_j} \tag{4}$$

$$Recall_{macro} = \frac{1}{n - classes} \sum_{j=1}^{nclasses} \frac{TP_j}{TP_j + FN_j} \tag{5}$$

$$F1_{macro} = \frac{2 * Precision_{macro} * Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$DR = \frac{TP}{TP + FN} \tag{8}$$

$$Precision_{Weighted} = \frac{\sum_{j=1}^{nclasses} Precision_j * S_j}{\sum_{j=1}^{nclasses} S_j} \tag{9}$$

$$Recall_{Weighted} = \frac{\sum_{j=1}^{nclasses} Recall_j * S_j}{\sum_{j=1}^{nclasses} S_j} \tag{10}$$

$$F1_{Weighted} = \frac{\sum_{j=1}^{nclasses} F1_j * S_j}{\sum_{j=1}^{nclasses} S_j} \quad (11)$$

Multi-class result of LR, SVM, GNB, ANN and RF for simulated WSN data, IEEE-IoT-IDS and WSN-DS data is shown in Table 7 , 8 respectively. From the tables, it can be identified that the Random Forest shows the best average and weighted accuracy, Precision_{macro}, Recall_{macro} and precision(weighted), recall(weighted), and F1(weighted). Gaussian Naive Bayes has the least performance, whereas the accuracy of ANN and SVM is better than the LR.

Table 7 Comparison of ML Models - Simulated WSN Data

Metrics used	LR	SVM	GNB	ANN	RF
$\text{F1 Score}_{Weighted}$	0.73	0.69	0.72	0.82	0.90
F1 Score_{Macro}	0.63	0.45	0.48	0.71	0.90
$\text{Precision}_{Weighted}$	0.78	0.75	0.75	0.84	0.90
Precision_{Macro}	0.63	0.42	0.49	0.63	0.92
$\text{Recall}_{Weighted}$	0.77	0.69	0.66	0.77	0.88
Recall_{Macro}	0.66	0.52	0.51	0.64	0.92
Accuracy	0.64	0.73	0.49	0.75	0.93

4.1 Rules Generation for Multiple Attacks

The comparative analysis done in previous section gives the Random Forest as the best model for intrusion detection. Hence, model parameters of the random forest are used for generating the rules. These rules are deployed on the IDS-agents for monitoring and detecting multiple attacks. The RF based rule generating model is getting executed on the base station. The rules are getting updated periodically. Steps to generate rules using Random Forest are shown in Algorithm 1.

In the Algorithm, the labeled WSN Intrusion Data is divided into training and test data as D-Training and D-Test respectively. D-Training is used for building Random Forest, which constructs *dtN* number of decision trees. The rules the decision tree DT_i are saved in set R_i. All the rules are merged and saved in set R. Once the Rule-set is constructed, the next step is to identify the optimal rules for intrusion detection. Optimal rules are identified using the test set TrDs-test. For that, three steps are used, first for each sample in the test set the prediction is obtained using the ensemble approach of the random forest using majority voting. For the correct prediction CrrPEM count is incremented. For each sample in the test set prediction is obtained using Rule_i in the second step. Subsequently, it is checked if it is correct prediction then CRl count is incremented. In the third step, Rule_i is checked for optimality considering the condition if correct predictions count using Rule_i is at least better than 50% of the predictions using ensemble approach, then

Table 8 Tables which are too long to fit, should be written using the "sidewaystable" environment as shown here

	IEEE IoT Dataset [1]					WSN-DS[2]				
Metrics used	LR	SVM	GNB	ANN	RF	LR	SVM	GNB	ANN	RF
F1 Score Weighted	0.71	0.69	0.69	0.78	0.85	0.74	0.69	0.71	0.80	0.87
F1 Score Macro	0.58	0.41	0.43	0.66	0.87	0.60	0.43	0.46	0.69	0.89
Precision Weighted	0.71	0.68	0.74	0.78	0.89	0.80	0.71	0.76	0.81	0.90
Precision Macro	0.62	0.41	0.45	0.62	0.88	0.64	0.44	0.47	0.66	0.89
Recall Weighted	0.73	0.65	0.63	0.77	0.84	0.78	0.67	0.67	0.79	0.86
Recall Macro	0.60	0.43	0.48	0.65	0.88	0.65	0.51	0.53	0.66	0.89
Accuracy	0.62	0.68	0.46	0.72	0.91	0.65	0.72	0.49	0.76	0.91

Algorithm 1 Random Forest Rule Generation (RFRG)

procedure GENERATE-RULE($Train - set - TrDs$)
Input: Training set TrDs = (X_1, y_1) (X_2, y_2), = (X_3, y_3)..... (X_n, y_n) Result: Rule set RLs = Rl_1, Rl_2, ... Rl_P
    DtreeN=Number of decision trees to construct in random forest
    for i=1 to DtreeN **do**
        Bstrap = BootStrapSampling(training set TrDs) ▷ Bstrap is subset from TrDs without replacement
        TrD_i = Decision tree using Bstrap
        Rl_i = All the rules generated by TrD_i
        Rall = R U Rl_i ▷ Rall is the set containing all the rules generated
    end for
    OptRules =ϕ ▷ OptRules : Set with optimized rules
    for each sample k in the test set **do**
        PV^k = Prediction using majority voting ▷ Prediction using ensemble approach
        IF PV^k== y_i ▷ Correct Prediction using Ensemble majority voting
        CrrPEM++; ▷ Count of correct prediction using Ensemble majority voting
        for each Rule i in Rall **do**
            $PR^k{}_i$ = Prediction using Rule i ▷ Prediction for sample k using Rule i
            if $PR^j{}_i$==y_i **then** ▷ Correct Prediction using Rule i
                CRl++ ▷ Count of correct prediction using Rule i
            end if
        end for
        if CRl > 0.5*CrrPEM **then**
            OptRules = OptRules U Rl_i ▷ OptRules: Optimized Rule Set
        end if
    end for
    return
end procedure

it is added to the Optimal Rule-set (OptRules). Algorithm 1 is executed at the base station periodically so that rules are generated and updated dynamically. Then the rules are executed to identify the malicious node behavior at the base station. The generated rules are used by the IDS agent during the monitoring state to identify the intrusions as malicious communication.

The algorithm to identify the attacks namely SF, Rank and DoS is shown in Algorithm 2. If PDRR is above the mentioned threshold δ_{PDRR} then it is considered as SF attack. If DPR and PFR are not within the specified threshold, it is considered a DoS attack. If the rank of the nodes are not according to the DODAG structure, then it results in a Rank attack. Results of multiple attacks such as rank attacks, selective forwarding and DoS attack and the normal (non-malicious) traffic using Rule-based technique is shown in Table

Algorithm 2 Detection Rules for Multiple Attacks

procedure ATTACKS-IDENTIFICATION($NodeID$)
  if ($\text{PDRR}_{NodeID} > \delta_{PDRR}$) **then**
    Send Message (Selective Forwarding, NodeID) to Sink
  end if
  if ($\text{DPR}_{NodeID} > \delta_{DPR}$) and ($\text{PFR}_{NodeID} > \delta_{PFR}$) **then**
    Send Message (DoS attack, NodeID) to Sink
  end if
  if Mismatch in Node-rank **then**
    Send Message (Rank-attack, NodeID) to Sink
  end if
  Return NodeID
end procedure

Table 9 Result of Multiple Attack Detection Using Rule Based Technique

Attack Types	Detection Rate (%)	Accuracy (%)
Normal	90.2	90.4
Selective Forwarding	88.4	87.1
DoS	84.7	85.2
Rank	89.3	88.4

Table 10 Result - With and Without Rule based Approach

Simulation(s) Time(s)	Total Nodes	malicious Nodes	Ids agents	RE[1] With [2]	RE[1] Without [2]
600	50	3	5	1.97	2.1
900	50	3	5	1.86	1.94
1200	50	3	5	1.74	1.88

[1]Residual Energy

[2]RBA (mJ)

9. From the table it can be observed that, all the attacks are identified with good DR and accuracy. However, Rank attack is identified with the best DR and accuracy. DR and accuracy of SF attack is better than the DoS attack. Table 10 shows the comparison between the IDS with the rule based and without using the rule based approach. From the results, it can be identified that the rule based approach shows improvement in energy consumption.

5 Conclusion

Due to recent growth in applications of wireless sensor networks, there is a possibility of more attacks which disturb the normal behavior of the network. The existing intrusion detection mechanisms are not sufficient to identify multiple intrusions dynamically and also mechanisms consume more than required energy which in-turn degrades the performance of the network. The proposed work emphasizes on the complete framework consisting the phases from data capturing process to optimization of energy consumption. The framework starts with the data generation process for WSN using Cooja simulator on Contiki operating system.

Appropriate features are selected as part of pre-processing task. In order to identify multiple intrusions, a dynamically updated model for rule generation is identified by building machine learning models on IEEE-IoT-IDS, WSN-IDS, and simulated data. GNB, LR, ANN, SVM, and RF supervised machine learning models are built on these data and the performance of these algorithms is analyzed. From the analysis, it is identified that the Random Forest model is more suitable to generate the rules. The rule generation process is deployed on base station to avoid energy consumption by IDS-agent nodes and updates the rules dynamically. IDS-agent present in cluster head interacts with the base station and informs the base station if it observes any malicious activity. In the future, the IDS can be enhanced by considering a broad range of attacks. The performance of the IDS can be further evaluated using different techniques of intrusion detection.

Declarations

- Funding
  - Open access funding provided by Manipal Academy of Higher Education, Manipal.
  - No funding was received for conducting this study.
- Conflict of Interest
  - Authors have no conflicts of interest to disclose.
- Ethics approval
  - Not Applicable
- Consent to participate
  - Yes
- Consent for publication
  - Yes
- Availability of data and materials

  - Will be provided on request
- Code availability
  - Will be provided on request
- Authors Contribution
  - First author has carried out the simulation of the RPL network, and implementation.
  - First author has written the paper.
  - Second author has given an idea and reviewed the paper

References

[1] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless Sensor Networks: A Survey on Recent Developments and Potential Synergies", The Journal of Supercomputing, vol. 68, no. 1, pp. 1–48, Apr 2014. [Online]. Available: https://doi.org/10.1007/s11227-013-1021-9

[2] M. Diaz, C. Martín, B. Rubioy, "State-of-the-art, Challenges, and Open Issues in the Integration of Internet of Things and Cloud Computing", J. Netw. Comput. Appl.,, vol. 67, no. 1, pp. 99–117, 2016.

[3] S. K. P. S Prasad Nayak; S Das, S.C. Rai, "SIMAS: Smart IoT Model for Acute Stroke Avoidance", International Journal of Sensor Networks (IJSNET), vol. 30, no. 2, pp. 83–92, 2019.

[4] D. Li, Z. Cai, L. Deng, X. Yao, and H. H. Wang, "Information Security Model of Block Chain Based on Intrusion Sensing in the IoT Environment", Cluster computing, vol. 22, no. 1, pp. 451–468, 2019.

[5] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey and T. Winter , "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks". [Online]. Available: https://rfc-editor.org/rfc/rfc6550.txt

[6] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler et al., "Transmission of IPv6 Packets Over IEEE 802.15. 4 Networks", Internet proposed standard RFC, vol. 4944, p. 130, 2007.

[7] T. Kushalnagar, G. Montenegro, C. Schumache, "IPv6 over Low-power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", 2007.

[8] Kim, Hyung-Sin and Ko, Jeonggil and Culler, David and Paek, Jeongyeup, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Ne works (RPL): A Survey", IEEE Communications Surveys & Tutorials, vol. 19, pp. 2502–2525, 09 2017.

[9] Imed Romdhani, Ahmed Yassin Al-Dubai, Mamoun Qasem and Baraq Ghaleb, "Cooja Simulator Manual", Edinburgh Napier University, Tech. Rep., 2016.

[10] M. Carlos-Mancilla, E. López-Mellado, and M. Siller, "Wireless Sensor Networks Formation: Approaches and Techniques", Journal of Sensors, vol. 2016, 2016.

[11] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology", Information, vol. 7, no. 2, 2016.

[12] Jun Zhang and Yang Xiang and Wanlei Zhou and Yu Wang, "Unsupervised Traffic Classification using Flow Statistical Properties and IP Packet Payload", Journal of Computer and System Sciences, vol. 79, no. 5, pp. 573 – 585, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000012001729

[13] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN Fragmentation Attacks and Mitigation Mechanisms", in Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2013, pp. 55–66. [Online]. Available: http://doi.acm.org/10.1145/2462096.2462107 [

[14] Y. Maleh, A. Ezzati, Y. Qasmaoui, and M. Mbida, "A Global Hybrid Intrusion Detection System for Wireless Sensor Networks", Procedia Computer Science, vol. 52, pp. 1047 – 1052, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915009084

[15] Pediaditakis, Dimosthenis and Tselishchev, Yuri and Boulis, Athanassios, "Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator", in Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, ser. SIMUTools, 2010, pp. 53:1–53:6. [Online]. Available: https://doi.org/10.4108/ICST.SIMUTOOLS2010.8727

[16] K. Weekly and K. Pister, "Evaluating Sinkhole Defense Techniques in RPL Networks", in Proceedings of the 2012 20th IEEE International Con ference on Network Protocols (ICNP), ser. ICNP '12. IEEE Computer Society, 2012, pp. 1–6.

[17] M. Gunasekaran and P. Subathra, "Ga-dosld: Genetic algorithm based denial-of-sleep attack detection in wsn," Security and Communication Networks, vol. 1, no. 1, pp. 1–10, 2017, doi:10.1155/2017/9863032.

[18] W. Wang, H. Huang, Q. Li, F. He, and C. Sha, "Generalized intrusion detection mechanism for empowered intruders in wireless sensor networks", IEEE Access, vol. PP, pp. 1–1, 02 2020, doi:10.1109/ACCESS.2020.2970973.

[19] R. Rebello, V. Pai, and K. Pai, "A review: Intrusion detection systems in remote sensor network," 11 2019, pp. 313–317, doi:10.1109/ICSSIT46314.2019.8987789.

[20] Y. Miao, Z. Ruan, L. Pan, J. Zhang, and Y. Xiang, "Comprehensive analysis of network traffic data", Concurrency and Computation: Practice and Experience, vol. 30, no. 5, p. e4181, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4181

[21] B. Alothman, "Raw network traffic data preprocessing and preparation for automatic analysis", in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2019, pp. 1–5.

[22] L. Vokorokos, A. Pekár, and N. Ádám, "Data preprocessing for efficient evaluation of network traffic parameters", in 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), 2012, pp. 363–367.

[23] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review", Computers and Security, vol. 30, no. 6, pp. 353–375, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404811000691

[24] Y. Chen, X. Ma, and X. Wu, "Ddos detection algorithm based on preprocessing network traffic predicted method and chaos theory", IEEE Communications Letters, vol. 17, no. 5, pp. 1052–1054, 2013.

[25] J. S. Rupinder Singh and R. Singh, "Fuzzy based advanced hybrid intrusion detection system to detect malicious nodes in wireless sensor networks", Wireless Communications and Mobile Computings, vol. 1, no. 1, pp. 1–15, April 2017, doi:10.1155/2017/3548607.

[26] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion Detection in the RPL-connected 6LoWPAN Networks", in Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, ser. IoTPTS '17, 2017, pp. 31–38. [Online]. Available: http://doi.acm.org/10.1145/3055245.3055252

[27] G. Nadiammai and M. Hemalatha, "Effective approach toward intrusion detection system using data mining techniques", Egyptian Informatics Journal, vol. 15, no. 1, pp. 37 – 50, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1110866513000418

[28] A. T. Yousef El Mourabit, .Anouar Bouirden and N. E. Moussaidr, "Intrusion detection techniques in wireless sensor network using data mining algorithms: Comparative evaluation based on attacks detection", vol. Vol. 6, No. 9. International Journal of Advanced Computer Science and Applications, 2015, pp. 164–172.

[29] Stratosphere laboratory. a labeled dataset with malicious and benign iot network traffict.

[30] 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, 2016, doi:10.1155/2016/4731953.

[31] M. C. Belavagi and B. Muniyal, "Multiple Intrusion Detection in RPL based Networks", International Journal of Electrical and Computer Engineering(IJECE), vol. 1, no. 10, pp. 467–476, Feb 2020.

[32] H. Sahli, "An introduction to machine learning", TORUS 1–Toward an Open Resource Using Services: Cloud Computing for Environmental Data, pp. 61–74, 2020.

[33] S. Angra and S. Ahuja, "Machine learning and its applications: A review", in 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), 2017, pp. 57–60, doi:10.1109/ICBDACI.2017.8070809.

[34] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection", IEEE Access, vol. 6, no. 1, pp. 33 789–33 795, 2018, doi:10.1109/ACCESS.2018.2841987.

[35] B. Yegnanarayana, Artificial neural networks. PHI Learning Pvt. Ltd., 2009.

[36] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks", Journal of Big Data, vol. 7, pp. 1–41, 2020, doi:10.1186/s40537-020-00305-w