Wireless Personal Communications (2023) 128:1839–1857
https://doi.org/10.1007/s11277-022-10022-7

IoT Routing Attacks Detection Using Machine Learning Algorithms

Sana Rabhi[1] · Tarek Abbes[2] · Faouzi Zarai[1]

Accepted: 29 August 2022 / Published online: 15 September 2022

Abstract
Internet of Things (IoT) is a concept that aims to make the real world more intelligent but susceptible to various attacks. In this paper, we focus on wireless sensor networks (WSNs), as a founding block in the IoT presenting the vulnerability of routing attacks against Routing Protocol for Low power and Lossy Network (RPL). Besides, we discuss some existing research proposals to detect intrusions, and we develop a technique for detecting three types of attacks against RPL. We simulate using Contiki-Cooja four network scenarios one normal and three malicious presenting different attacks, to be able to generate the training and the test sets that are used in the machine learning phase, in which we used WEKA, to decide according to the database whether the behavior is normal or malicious. For this phase, we use different classification algorithms, which enable us to obtain a high precision value that is superior to 96% in all cases.

Keywords WSN · RPL · Attacks · Machine learning · Classification

1 Introduction

The Internet of Things (IoT) is a concept that aims to make the real world more intelligent, by connecting objects without human intervention, and which is implicated in almost all the fields; Energy management, transportation systems, homes and buildings, industry and even healthcare. So it will be a world blanketed with billions of sensors that are taking information from real physical objects and uploading it to the Internet. This entails the exploitation of wireless sensor networks (WSNs), radio identification systems (RFID)...

✉ Sana Rabhi
sana.rabhi.etud@enetcom.usf.tn

Tarek Abbes
Tarek.abbes@enetcom.usf.tn

Faouzi Zarai
faouzifbz@gmail.com

[1] National School of Electronics and Telecommunication, 3000 Sfax, Tunisia

[2] Innov'Com Laboratory, SUP'COM, University of Sfax, Sfax, Tunisia

based on various protocols that enable the communication of these devices (e.g., Wi-Fi, Bluetooth, ZigBee,etc.).

In this paper we address the challenge to WSN as the founding block of IoT, which is an ad-hoc network with a large number of low-cost and battery-powered sensor nodes. The role of these sensors is to detect physical or environmental conditions such as heat, humidity, pressure, movement, etc. These networks generally include sink nodes, sensor nodes, and clients. Sensor nodes, which are randomly deployed, collect data and send it to the base station in this network which is the sink node. Over the transmission process, data may be handled by multiple nodes to get to the sink node after multi-hop routing, lastly attain the end user through the internet or satellite. According to specifications provided by IEEE 802.15.4 [1] WSN uses 6lowpan (IPv6 over Low- Power Wireless Personal Area Networks) and RPL (Routing Protocol for Low power and Lossy Network) on the network layer and CoAP (Constrained Application Protocol) or MQTT (Message Queuing Telemetry Transport) on the application layer.

RPL is a novel distance vector routing protocol standardized for constrained 6LoWPAN networks enabling nodes to communicate in a mesh topology [2].However, it is susceptible to diverse security issues and has some important privacy concerns, so appropriate mechanisms to secure communications will be fundamental. In this paper, we propose an anomaly based detection method to discover specific RPL attacks in a WSN network, relying on classification algorithms. Thereby we use WEKA as machine learning tool, to decide whether a behavior is normal or malicious according to datasets that are obtained by monitoring different network scenarios simulated using Contiki-Cooja.

Our proposed solution has the benefit of detecting three types of attacks using three different classification algorithms and ensemble learning. Besides unlike previous work it puts minimum charge on the wireless sensor network and enhances power consumption.We also ensure a lightweight detection by selecting only meaningful features to reveal matches.

The rest of the paper is organized as follows. In Section II we discuss the routing attacks against RPL. In Section III, we present the related work. The implementation of the proposed system is explained in Section IV and results are depicted in section V. Finally, we conclude the paper in section VI.

2 Routing Attacks Against RPL

In this section, we address the RPL protocol, its topology and its well-known attacks.

2.1 Routing Protocol for Low power and Lossy Network:RPL

Routing protocols are the basic building block for communication in any network. Since wireless sensor networks have strong resource limitations (energy, memory, computing power), the routing protocols for typical wired networks (OSPF, IS -IS) and for ad-hoc networks (AODV, OSLR) are not suitable for the characteristics of this type of network. That's why the IETF Routing over Low power and Lossy networks working group (ROLL) standardized a new routing protocol called IPv6 Routing Protocol for LLNs (RPL) [3].

RPL is developed specifically for the 6LoWPAN network, the main idea of this protocol is having instant knowledge of the state of the network due to DODAG (Destination Oriented Dynamic Acyclic) graphs. DODAG is a graph which organizes nodes into a hierarchical structure of a single destination that is the root node of the network, children and further descendants. This graph is created using an objective function that helps to optimize the metric used in the choice of routes. An RPL DODAG is created by the use of the following ICMPv6 control messages, as depicted in Fig. 1, DODAG Information Solicitation (DIS), DODAG Information Object (DIO) and Destination Advertisement Object (DAO).

A node can access a network by broadcasting DIS messages to request DIO messages from its neighbors. A DIO message is the most important message type in RPL, it is broadcasted initially by the root node, and it contains information needed by the other nodes to discover an RPL instance, so by receiving the DIO packets the nodes create their routing tables. These messages are broadcasted periodically with a rate that is set on by the trickle algorithm . If a new node joins the topology, all nodes send DIO packet again to reform DODAG and essentially, the more stable a DODAG is, the fewer DIO transmissions there are. DAO packets are sent to the parent node, asking permission for connection, the parent node therefore send back a DIO-ACK packet to accept this connection.

2.2 Attacks Against RPL

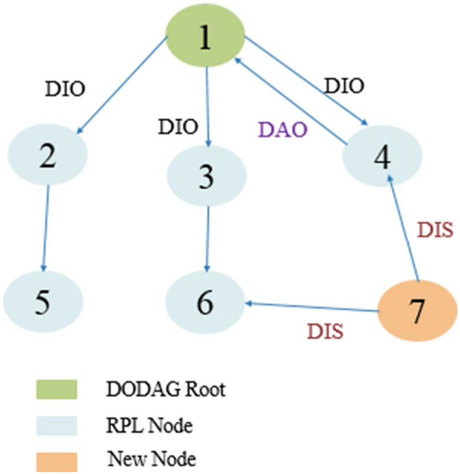
The taxonomy of RPL attacks as presented in [4], shows three categories of security attacks against RPL.

2.2.1 Attacks Against resources

This category covers attacks against resources, which leads to resource exhaustion by pushing the nodes to consume all their resources (energy, memory, and processor) in unnecessary actions.

- HELLO flooding : Flooding causes a congestion of the communication channels through retransmitting useless messages and high traffic. HELLO messages are exchanged between neighboring nodes in the network to declare their presence and availability. An attacker, using a powerful machine, sends a huge number of HELLO packets to different nodes, so that they will treat it as their neighbor, and they will trans-

Fig. 1 Control messages of DODAG



mit their data to this machine, thus besides the congestion of the communication channel, there is the loss of these packets. In RPL protocol a malicious node sends HELLO messages by DIS packets.

- Version number modification attack : The version number is an important field of each DIO message and it is incremented only by the root node. In this attack, the malicious node increases the version number, which causes an unnecessary graph rebuilding, so a DIO packet with invalid version number leads the root to update and reset its trickle timer to resend a new DIO.

2.2.2 Attacks Against topology

This category contains attacks targeting the topology of the network. The aim of these attacks is to disrupt the normal operation of the network, which could then cause the isolation of one or more legitimate nodes.

- Wormhole attack : These attacks can undercut or disable wireless sensor networks. In a wormhole attack, two malicious nodes establish a direct low-latency link between them, so they receive packets at one point in the network, sends them through the wormhole link and replays them at the other point [5].
- Black hole attack in a network would signify that one or more malevolent nodes would fully or partially drop data packets being routed through it, which leads to disruptions in the normal data flow in the network [1]. A malicious node falsifies the routing information, advertises itself as the best route towards the control node (called sink node), to force the passage of data by itself. Its only mission is then to transfer nothing, creating a kind of well or black hole in the network. The rise in the number of DIO messages exchanged between the nodes is a clear sign of this attack due to the rank change effectuated by the malicious node.

2.2.3 Attacks Against Network Traffic

This category presents attacks against network traffic. These attacks lead to information leakage by impersonating legitimate nodes or eavesdropping the traffic.

- Sybil attack: In this attack, the adversary may manipulate false identities on the same physical node to bother the performance of the network. By broadcasting messages with multiple identifications, a Sybil node can take control over large parts of a network.
- Spoofing : Spoofing is the identity attack, where the purpose of the hacker is to damage the data routing in the network that are controlled through the identity of nodes. With a legitimate ID, the attacker can take part in exchanging routing data by altering this data or distributing false information.

3 Related Work

In a network or a system, any type of illegitimate or unauthorized activity is an intrusion. The concept of intrusion detection was primary proposed by Anderson in 1980 [6] and is introduced to network system by Heberlein in the year of 1990 [7]. The intrusion detection system (IDS) is an active process that analyzes network activity and system by the gathering of tools, methods, and resources to identify and detect intruders or malevolent activities, and then declare an alarm to report that a malicious activity has occurred or is in progress. These systems (IDSs) afford specific information of the intruder that help in the detection such as identification of the attacker, location, time, intrusion type and layer where the intrusion occurs (physical, data link, network).

In [8], Gupta et al. suggested an architecture which uses computational intelligence algorithms to construct the normal behavior profile of each different device in the network. However, for this proposed system the authors did not consider networks with low capacity devices.

In [9], Kavitha et al. suggested a technique based on hierarchical cluster that detects anomalies in wireless networks. Authors assure that the proposed method is faster thanks to clustering algorithm when compared to other data mining methods. But the inconvenience of this method is that it is not adapted to larger dataset.

In [10], Yavuz et al. proposed a deep-learning-based machine learning method to detect some RPL routing attacks. The authors built deep neural network models trained with the IRAD datasets. They attained a high performance (over 99%), however, they didn't study the impact of the attacks on energy consumption.

In [11], the authors proposed a hyper grid KNN based anomaly detection in wireless sensor networks to detect cyber-attack. The detection region of conventional KNN is redefined as Hypercube and the parameters are estimated in a dynamic and adaptive way. The major benefit of this technique is that it is not required to adjust the parameters manually. But the authors assume that the training data do not contain any anomalous data although it is not always feasible to collect pure data from any WSN environment.

In [12], Napiah et al. proposed a Compression Header Analyzer Intrusion Detection System (CHA-IDS) to detectWormhole, Sinkhole, and HELLO flood attacks. That IDS analyzes 6LoWPAN compression header data to extract important features that are used for detecting combined and individual routing attacks. CHA-IDS uses best first and greedy stepwise with correlation-based feature selection to define only significant features needed for the detection, then those features are classified as normal and malicious traffic using different machine learning algorithms; Logistic Regression, Random Forest, Decision Trees, Naive Bayes, Support Vector Machine and Multilayer Perceptron. The major limitation of CHA-IDS is that it is not able to identify the attacker, besides it includes high energy and memory consumption.

[13] a signature-based IDS is proposed to detect version number and DIS attacks. In this IDS the detection and monitoring modules are placed on nodes, therefore the authors consider two types of additional nodes. The first are IDS routers, and the second are IDS detectors which monitor and send malicious traffic information to the IDS router that decides if the packet source is malicious or not, relying on the calculation of different metrics like; Received Signal Strength Indicator (RSSI), packet sending rate and packet drop rate. The major limitation of this technique is that it is not validated.

In [14] Shafique et al. proposed a specification based IDS to detect rank attacks in RPL networks. The proposed Sinkbased Intrusion Detection System (SBIDS) utilizes information in the DAO message such as node's previous rank (NPVR), node's current rank (NCR), node's parent rank (NPR), and parent switching threshold (PST) to distinguish normal and malicious nodes. This IDS reaches 100% accuracy in a static network, but it decreases when adding mobile nodes in the network, the other limitation is the high power consumption average of nodes.

In [15] the authors suggested a signature-based Network IDS to detect routing attacks in RPL-based IoT networks; Local Repair, Hello Flooding, SinkHole, Selective Forwarding, Sybil, BlackHole and Clone ID, this IDS called ELNIDS is based on combining different types of ML classifiers : Subspace Discriminant, Bagged Trees, Boosted Trees, and RUS-Boosted Tree. Each classifier is evaluated individually, then ensemble learning is applied to enhance the accuracy. Experimental results show that the best accuracy attained is 94.5% for the ensemble of Boosted Tree, and 77.8% for the Subspace Discriminant model.

In [16] Kumar et al. presented a unified intrusion detection system for IoT networks (UIDS) to detect DoS, exploit, generic and probe. The authors used the decision tree classifier on UNSW-NB15 dataset. Experimental results show that this signature-based IDS reaches better accuracy when compared with existing models: ENADS and DENDRON. However the UNSW-NB15 dataset is not specific to IoT , and the system is not able to detect unknown attacks.

In [17] Parra et al. proposed a cloud-based approach for IoT environment using deep learning to detect distributed attacks: Botnets, phishing and DDoS. The system includes two security models: a cloud-based temporal Long-Short Term Memory (LSTM) and a Distributed Convolutional Neural Network (DCNN) model. Experiments show that the best performance is attained when detecting Botnet attacks with an accuracy of 94.80%. For this approach more training time is needed.

In [18] Ulla et al. suggested an anomaly-based IDS for IoT networks to detect attacks such as DoS, DDoS, flooding attacks, OS Scan, Port Scan, Mirai, etc. the authors used a convolutional neural network model in 1D, 2D, and 3D to implement binary and multi-class classification. The proposed approach achieved high accuracy, but the deep learning approaches require more training time and computational costs.

In [19] the authors presented a supervised machine learning-based support vector machine (SVM) IDS, that detects attempts to inject unnecessary data into IoT networks. For this approach, Jan et al. used The CICID2017 dataset, where the rate of packet arrival was the only attribute considered to classify the packets as benign or intrusive.

Although these recent proposed approaches were able to attain high performance, there are many limitations that are needed to be addressed. Features in the dataset need to be more analyzed to examine the correlations between selected features. As we focus on WSN nodes, it is very important to minimize processing and communication costs. So, feature analysis needs to be optimized to reach this goal that was a priority in our approach in which we applied preprocessing and feature selection to our dataset to improve efficiency and reduce time and computational costs.

The proposed system overcomes the above limitations, as it puts minimum charge on the wireless sensor network as it needs only the packet traces of the network to detect and predict attacks which can be collected by specially designated nodes or network recording equipment, also our technique Enhances power consumption by notifying the network administrator at an early phase about a certain attack.

4 Proposed System

The large amount of network and sensing data generated by sensors in WSN makes machine learning methods very effective in detecting intruders. In this paper, we detected 3 types of routing attacks; blackhole, hello flooding and version number modification, using machine learning algorithms, where the main idea is defining a normal profile and comparing it the an observed one. To build our system we started with simulating four network scenarios using Contiki simulator which is a flexible and light operating system for sensor networks, it is open source, written in C and can be used in both commercial and noncommercial systems, Contiki has one of the major tools called cooja which is a software simulator designed for wireless sensor networks. The first scenario is called normal network, because it is free from any malicious activity. It includes 1 sink node and 24 sender nodes, these nodes were randomly placed on a grid of 200x200 meters, communicating using the protocol 6LoWPAN and the RPL as routing protocol. The other scenarios represent the malicious networks, in which one of the sender nodes is randomly selected from the 24 nodes to behave in a malicious way. After the simulation we moved to the preprocessing and feature selection to build our dataset.

4.1 Preprocessing and Feature Selection

In this section we describe the choice of the appropriate attributes to build the datasets needed for the machine learning phase.

4.1.1 Data Extraction

The data features we use in our machine learning models have a big influence on the results. Choosing appropriate attributes is a real fundamental challenge for good detection. To obtain data in relation with detecting the malicious nodes we continuously captured the network traffic considering observation windows of duration t, , in our case t = 5 seconds. Messages (such as DIS, DIO, and DAO) are observed using the "Radio messages" tool of COOJA, which enables us to generate PCAP files that are analyzed using Wireshark. By understanding the principal of the different attacks we took into account the following metrics:

- Number of DIS messages: The number of DIS messages exchanged between the nodes is calculated within a window size of 5 seconds.
- Number of DIO messages: The number of DIO messages exchanged between the nodes is calculated within a window size of 5 seconds.
- Number of DAO messages: The number of DAO messages exchanged between the nodes is calculated within a window size of 5 seconds.
- Version number modification: The version number describes the version of a DODAG graph, it is a field of DIO packet which is supposed to remain unchanged by the other nodes and only incremented by the root node. For our approach, to detect the version number modification attack we set the attribute version_modification = 0 if the version number is stable and version_modification = 1 if there is a modification.
- Rank value average: The rank of a node is a field of DIO packet which indicates the node's position within a DODAG with respect to the root. This value could be decreased by the malicious node to declare a better rank than neighbors, causing the modification of the DODAG, which is the principle of some attacks such as blackhole attack. In our approach we calculated the rank average in DIO message using window size of 5 seconds.

- Power consumption: In order to evaluate the impact of the different attacks on energy consumption, we considered the power consumption average of all the motes which is offered by the Power Tracker tool of COOJA.

A sample of the data captured from the simulation while monitoring the normal and the malicious behaviors (blackhole, Hello flooding, version modification) is presented respectively in the Tables 1, 2, 3 and 4.

It is obvious that due to malicious activity introduced by node11 in all the malicious networks, there is instability in the network topology. For the network containing a blackhole attack, there is a clear increase in the number of DIO messages during every t= 5

Table 1 Trace of the normal scenario

DIS_nbr	DIO_nbr	DAO_nbr	POWER_ CONSP %	Rank_avg	Version_ modif
517	30	0	1.82	128	0
0	242	1769	4.29	461.12	0
39	839	2375	6.99	765.51	0
0	1738	651	7.97	824.36	0
0	579	656	7.22	965.27	0
0	1129	1311	6.94	902.31	0
0	712	386	7.8	1142.89	0

Table 2 Trace of Blackhole attack scenario

DIS_nbr	DIO_nbr	DAO_nbr	POWER_ CONSP %	Rank_avg	Version_ modif
517	103	0	1.72	128	0
0	1253	2343	3.78	402.31	0
39	1633	2576	7.63	632.15	0
0	1508	2269	8.15	745.54	0
0	877	1173	10.42	702.35	0
0	1356	1730	11.45	845.12	0
0	1926	1203	11.4	933.27	0

Table 3 Trace of HELLO Flooding attack scenario

DIS_nbr	DIO_nbr	DAO_nbr	POWER_ CONSP %	Rank_avg	Version_ modif
1278	30	0	8.19	128	0
952	137	973	9.82	423.12	0
948	193	1146	11.25	794.64	0
691	821	936	11.89	789.84	0
866	898	1240	11.79	951.23	0
550	439	2435	12.19	917.45	0
521	892	320	12.11	1003.62	0

Table 4 Trace of Version modofication attack scenario

DIS_nbr	DIO_nbr	DAO_nbr	POWER_ CONSP %	Rank_avg	Version_ modif
517	30	0	1.78	128	0
0	1146	1008	3.64	485.98	0
39	2094	2513	8.01	754.24	0
0	2690	966	9.08	893.82	0
0	1021	1203	8.61	962.17	1
0	810	854	9.1	973.12	1
0	760	872	9.78	1174.48	1

seconds, when compared to normal network scenario, we note also that there is a decrease in the rank average which explains the impact of the attack. As explained in the previous section, a node performing hello flooding attack broadcasts a huge number of DIS packets which is evident in the dataset. In the last dataset, showing the results of the version number modification attack, the attribute version_modification = 1, which means that there is modification in the version number.

For the power transmission consumption average, it is clear that the malicious activity increased the energy consumption during the whole time of the simulation in all attack scenarios. This effect is depicted by the Fig. below(2, 3, 4 and 5), where the power consumption of each node on the active mode (ON) in the transmission radio (Tx) and the reception radio (Rx) is presented in the different scenarios.

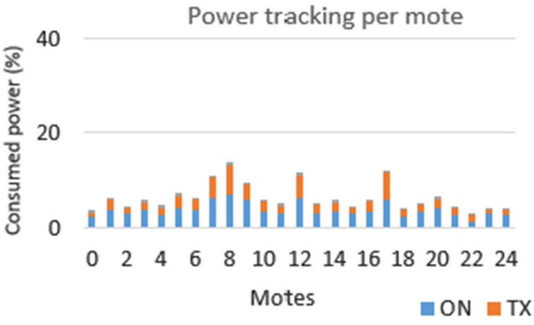


Fig. 2 Power tracking per mote in the normal scenario

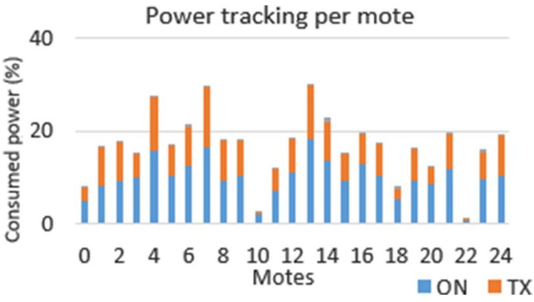


Fig. 3 Power tracking per mote in the Blachhole scenario

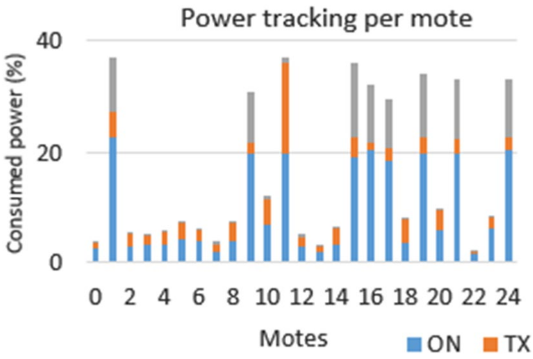


Fig. 4 Power tracking per mote in the Hello flooding scenario

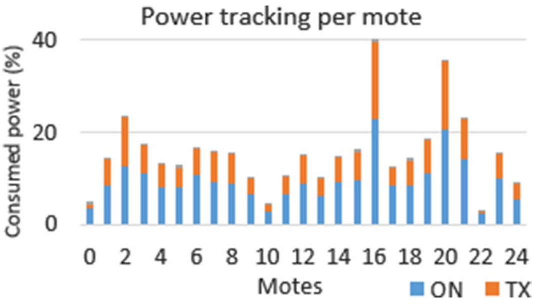


Fig. 5 Power tracking per mote in the version modification scenario

4.1.2 Feature selection

The goal of machine learning is to study and train algorithms so that they can learn and make predictions on a large amount of data. Before running any machine learning algorithm, feature selection is usually applied to the dataset which is the first and the most important step of designing any model because the attributes that are used to train the machine learning model have a big influence on the performance.

Feature Selection is the process that enables us to manually or automatically select the optimal subset of all features and eliminate the weakly relevant and irrelevant ones which can negatively impact the model performance. For the machine learning phase, including the feature selection process, we used Weka (Waikato Environment for Knowledge Analysis), which is a collection of machine learning algorithms for data mining, developed at the University of Waikato, New Zealand and it's a free software available under the General Public License (GNU) [20, 21].

The Weka tool is written in Java and it enables:

- The pre-processing and Analysis of the features in a database.
- The Definition of the class attributes which separates the instances into the suitable classes.
- The application of classification, regression and clustering algorithms.
- The implementation of the most artificial intelligence algorithms, including decision trees and neural networks.
- The estimation of the selected algorithm's performance.

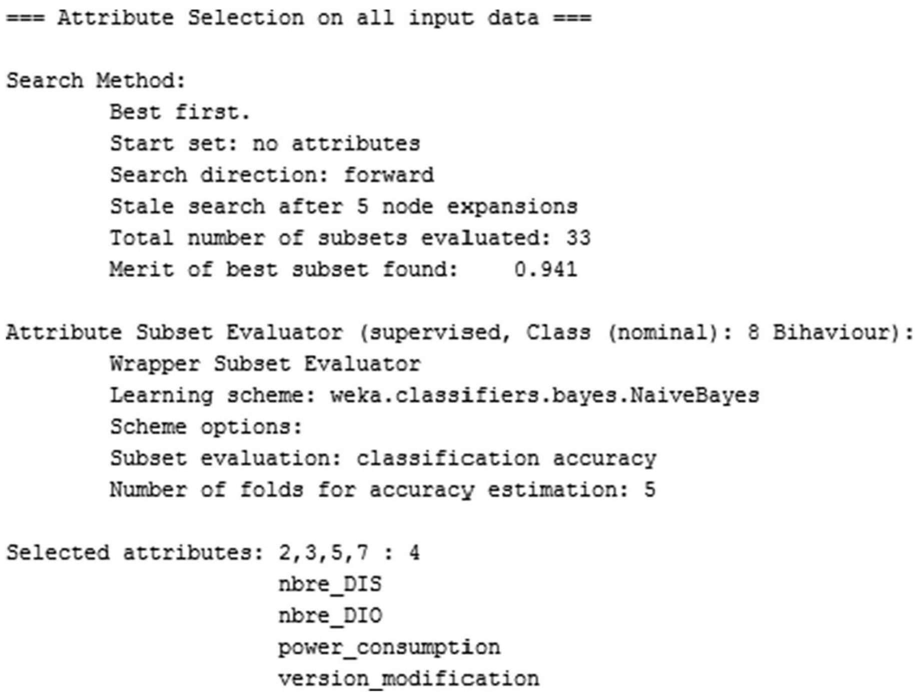
To effectuate feature selection usingWeka, three main elements are required, the first is dataset, the second is search method and the last is evaluation method. Both search method and evaluation method need to be initiated and defined in a container class AttributeSelection. The attribute evaluator is the technique which enables evaluating each feature in the dataset in the context of the output variable, and Search Method evaluates each attribute and lists the results in a chosen form (for example in a rank order or the best first...).

In our model, we used WrapperSubsetEval technique as attribute evaluator, that is a popular technique which uses a chosen learning algorithm to evaluate attribute sets, it searches through the attribute space and uses the classifier to find the best attribute set. The result of attribute selection using WrapperSubsetEval is shown in Fig. 6.

As shown in the figure, the selected attributes are number of DIS messages, number of DIO, power consumption and version modification, so it’s obvious that the attributes DAO number and Rank avg are eliminated. This can be explicated by the fact that, as shown in the datasets of the attack scenarios the increase of the number of DAO packets is very low, also the decrease of the rank average when compared to the normal dataset, which makes this attributes weakly relevant.

4.2 Learning

There are two main classification techniques, supervised and unsupervised.



```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 33
        Merit of best subset found:    0.941

Attribute Subset Evaluator (supervised, Class (nominal): 8 Bihaviour):
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.bayes.NaiveBayes
        Scheme options:
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 2,3,5,7 : 4
                     nbre_DIS
                     nbre_DIO
                     power_consumption
                     version_modification
```

Fig. 6 Feature selection using Weka

4.2.1 Supervised learning

The purpose of supervised learning is to find a function or model that can predict the label or class of a sample as accurately as possible, from a labeled training set. This type of learning approach is used to resolve various issues for WSNs like event detection, objects targeting and localization, medium access control, intrusion detection and security, data integrity and QoS [22]. Some of supervised machine learning algorithms are presented below.

- Decision Trees : The decision tree classification includes repeating input of data using tree of learning to predict output labels. A decision tree is composed of three main elements: a decision node representing test or condition on data item, a branch which corresponds to the one of the test attribute outcomes and a leaf which define the class to which the object belongs.
- Support Vector Machines : Support Vector Machines offer alternatives for neural networks, which are favored options for solving non convex unconstrained optimization problems, this algorithm is based on the concept of decision plans defining decision boundaries. A decision plan is a plan that separates a set of objects that belong to different classes. In WSN, SVM have been used for detecting the malicious behavior of sensor nodes.
- Naïve Bayes : Naïve Bayes can be considered as an improved version of Bayes Theorem. Learning techniques based on Bayesian statistics require lesser training samples than the others. Bayesian classifier encodes probabilistic relationships between variables of interest. This means that the probability of one attribute does not influence the probability of the other.

4.2.2 Unsupervised learning

The purpose of unsupervised learning is to discover classes within samples by grouping them by similarity without any prior knowledge. This type of learning algorithm is used in WSN node clustering or data aggregation at sink code scenarios. In this category there are two major types of algorithms, K-means clustering and principal component analysis.

- K-Means Clustering : This learning algorithm classifies data into different clusters. and works according to three major in steps starting with a random selection of k nodes as first centroids for different clusters, then the use of a distance function to instructions every node with the nearest centroid finally iteratively re-compute the centroids using a predefined threshold value and stop the iterations if the convergence condition is met. The Kmeans clustering algorithm is favored in WSN sensor node clustering because of its simplicity.
- Principal Component Analysis: This learning algorithm is popular into data compression field and it is utilized for dimensionality reduction. It is a multivariate method and its purpose is to extract important information from data in terms of principal components, which is nothing however a set of new orthogonal variables [17]. Further, this method can solve the big data problem into small data by permit-

ting selection of only significant primary components and eliminating other lower order insignificant components from the model.

The supervised algorithms achieve excellent results for known intrusions, they are better than unsupervised algorithms. In our system we are detecting already known attack so we are using supervised learning.

5 Results and performance

In our case we are effectuating a comparison between some classification algorithms usingWeka. Data used for the learning phase were captured and recorded by the simulator and they are divided into 2 sets. The first is the training set which is used for learning the model. The second is the test set that is used to validate and evaluate the model's performance. The evaluation of the quality of classification is done with different measures.

- Confusion matrix is an evaluation technique applied to all the types of classification problems. It displays the four values ; true positive (TP), true negative(TN), false positive(FP), and false negative(FN) in a way that the relationship between them is easily understood as shown in Table 5.
- The precision shows how many intrusions predicted by an IDS are real intrusions. A convenient IDS should aim for high accuracy, which means that false alarms are minimized.

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

- The recall is a metric that shows the percentage of predicted intrusions in relation to all intrusions present. A convenient IDS should have a high recall value.

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

- The F-Measure uses a combination of precision and recall.

$$F\_Measure = 2\frac{precision.recall}{precision + recall)} \tag{3}$$

Tables 6, 7 and 8 below show the result of learning with Weka using the respective classification algorithms: SVM, Naïve Bayes and decision tree.

Results show that, in our case we obtained high performance with all the used classification algorithms, we can see that we have a very low false positive rate and a precision

Table 5 Caption text

	Predicted positive	Predicted negative
True positive	TP	FN
True negative	FP	TN

Table 6 Results and performance of SVM classification algorithmo

Class/metric	True positive rate	False positive rate	Recall	F-Mesure	Precision
Normal	1	0.05	1	0.952	0.909
Blackhole	0.9	0.014	0.9	0.923	0.947
Hello flooding	1	0	1	1	1
Version modification	0.9	0	0.9	0.947	1
Weighted average	0.956	0.02	0.956	0.955	0.958

Table 7 Results and performance of Naïve Bayes classification algorithm

Class/metric	True positive rate	False positive rate	Recall	F-Mesure	Precision
Normal	0.967	0.017	0.967	0.967	0.967
Blackhole	0.975	0.036	0.975	0.929	0.886
Hello flooding	1	0	1	1	1
Version modification	0.9	0	0.9	0.947	1
Weighted average	0.961	0.013	0.961	0.961	0.964

Table 8 Results and performance of decision tree classification algorithm

Class/metric	True positive rate	False positive rate	Recall	F-Mesure	Precision
Normal	1	0.05	1	0.952	0.909
Blackhole	0.925	0.007	0.925	0.949	0.974
Hello flooding	1	0	1	1	1
Version modification	0.9	0	0.9	0.947	1
Weighted average	0.961	0.018	0.961	0.961	0.964

value superior to 95% in all cases. The best performance is observed with Naïve Bayes classifier, in which the false positive rate is about 0.017, 0.036,0 and 0 for the classes normal, blackhole, hello flooding and version modification respectively, and the average of the precision value for all the classes is equals to 96%.

- ROC curves and AUC calculations: To study the accuracy of a system, outcomes like ROC curves and AUC are important to examine. The aim of ROC (Receiver Operating Characteristic) Curves is to analyze the performance of a classifier, by creating a graph of the True Positives versus False Positives for each classification threshold. A ROC curve that is closer to the upper left corner represents a powerful classifier, that perfectly separates the classes, while a curve that falls near the line y =x represents an inefficient classifier. The Area Under the Curve value (AUC), is a method used to quantify the classifier performance and is given onWeka with the ROC curve. Most classifiers have AUCs that fall somewhere between 0.5 and 1, a perfect classifier has an AUC value equals to 1, while an AUC value equals to 0.5 describes a classifier with no power. The Fig. 7, 8, 9 bellow show ROC curves for each attack using the different

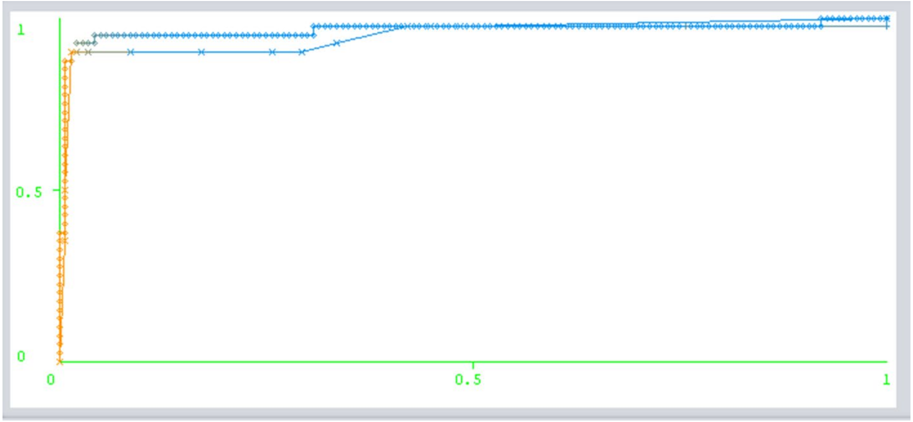


Fig. 7 ROC curves of the Blackhole class

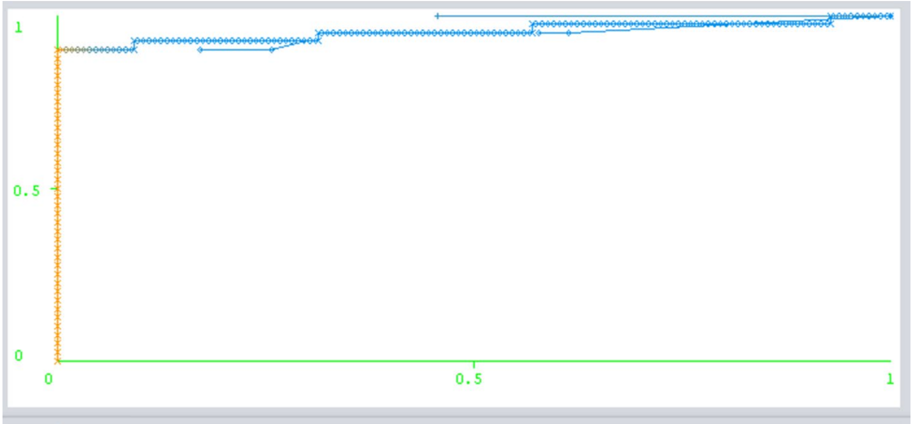


Fig. 8 ROC curves of the Version modification class

classification algorithms. ROC curves show that the accuracy of Naive Bayes also overcomes the accuracy of decision tree and SVM classifiers for the blackhole and version modification attacks, and it's almost the same for the hello flooding attack. The performance of Naïve Bayes is also proved with the AUC value which is superior to 0.98 for all the attacks and it's over 0.97 for the SVM and decision tree classifiers.

- Ensemble learning method: Ensemble algorithms are a powerful class of machine learning algorithm where decisions of multiple algorithms are combined in some way to improve the performance of the model. The principal concept of ensemble learning is integrating several single approaches to improve the performance of the final classifier, therefore an

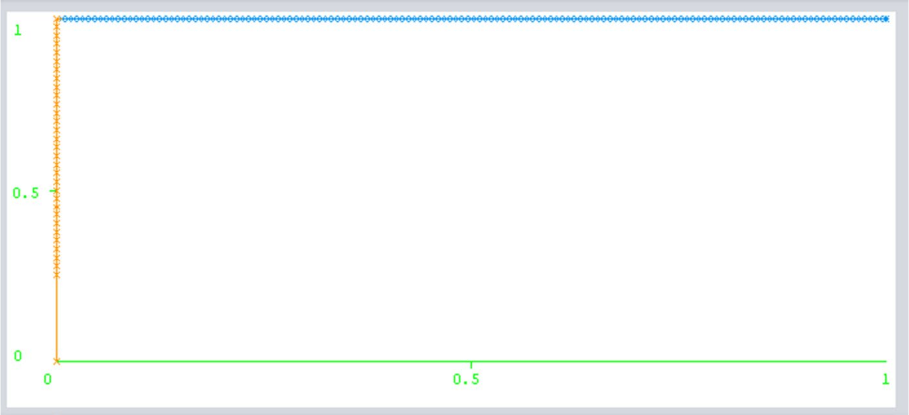
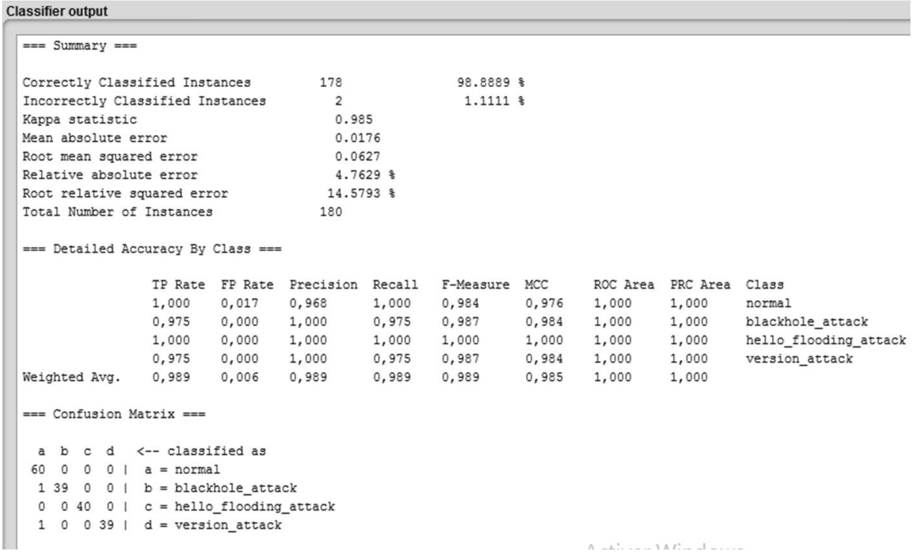


Fig. 9 ROC curves of the Hello flooding class



```
=== Summary ===

Correctly Classified Instances         178               98.8889 %
Incorrectly Classified Instances         2                1.1111 %
Kappa statistic                          0.985
Mean absolute error                      0.0176
Root mean squared error                  0.0627
Relative absolute error                  4.7629 %
Root relative squared error             14.5793 %
Total Number of Instances              180

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,017    0,968      1,000    0,984      0,976    1,000     1,000     normal
                 0,975    0,000    1,000      0,975    0,987      0,984    1,000     1,000     blackhole_attack
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     hello_flooding_attack
                 0,975    0,000    1,000      0,975    0,987      0,984    1,000     1,000     version_attack
Weighted Avg.    0,989    0,006    0,989      0,989    0,989      0,985    1,000     1,000

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 60  0  0  0 |  a = normal
  1 39  0  0 |  b = blackhole_attack
  0  0 40  0 |  c = hello_flooding_attack
  1  0  0 39 |  d = version_attack
```

Fig. 10 Results of learning with Ensemble learning method

ensemble classifier can have better accuracy than the individual base classifiers. Voting is the simplest ensemble algorithm and is frequently highly effective which can be used for regression or classification. The results show that, by using this ensemble method we obtained higher performance than single classifiers, we can see that we have a very low false positive rate and a precision value superior to 98% as shown in Fig. 10 bellow, which make this algorithm very accurate in our case.

6 Conclusion

RPL in developed specifically for the 6LoWPAN protocol used within WSN networks to be suitable to resource limitations of WSN devices. However, this protocol is vulnerable to various threats that cannot be ignored and requires powerful security countermeasures.

Our aim was to detect three types of attacks against RPL protocol in WSN networks, ensuring a best performance by selecting only relevant features to create single models of classification, then a combined model to improve the accuracy. During this work, we followed several steps in order to reach our goal, We first simulated using Contiki Cooja simulator four network scenarios, one without attacks, called normal, and the others with one malicious sensor node in each scenario performing one of the attacks :blackhole, Hello flooding and version number modification. Then we build our training sets, which are needed for the learning phase. Our detection method relies on finding the optimal attribute set to reveal matches, so we used the Weka feature selection tool.

For the learning phase, we used different classification models and an ensemble learning model, which allowed us to obtain a high precision value that is superior to 96% in all cases .

As future work, we plan to simulate different scenarios with various rate of malicious and normal nodes along with a higher number of nodes. We also intend to raise our IDS prediction performance to detect additional routing attacks, so investigating more routing metrics as delay, hop count, throughput and bandwidth.

Author Contributions Authors contributed equally to this work.

Funding No funding was received to assist with the preparation of this manuscript.

Data Availability Our data set is obtained from simulated scenarios, by capturing the network traffic considering observation windows of duration t = 5 seconds. “Radio messages” tool of COOJA enables us to generate PCAP files that are analyzed using Wireshark. The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest/Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Code Availability Not applicable.

References

1. Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials., 17*(3), 1294–1312.
2. Kfoury, E., Saab, J., Younes, P., & Achkar, R. (2019). A self organizing map intrusion detection system for RPL protocol attacks. *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)., 11*(1), 30–43.
3. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., & Levis, P., et al. (2012). RPL: IPv6 routing protocol for low-power and lossy networks;

4. Wallgren, L., Raza, S., & Voigt, T. (2013). Routing attacks and countermeasures in the RPL-based internet of things. *International Journal of Distributed Sensor Networks., 9*(8), 794326.
5. Pongle, P., Chavan, G. A., & survey: Attacks on RPL and 6LoWPAN in IoT. In,. (2015). International conference on pervasive computing (ICPC). *IEEE, 2015*, 1–6.
6. Anderson, J. P. (1980). *Computer security threat monitoring and surveillance*. James P Anderson Company: Technical Report.
7. Heberlein, LT., Dias, GV., Levitt, KN., Mukherjee, B., Wood, J., & Wolber, D. (1989). A network security monitor. Lawrence Livermore National Lab., CA (USA); California Univ., Davis, CA (USA ...;
8. Gupta, A., Pandey, OJ., Shukla, M., Dadhich, A., Mathur, S., & Ingle, A. (2013). Computational intelligence based intrusion detection systems for wireless communication and pervasive computing networks. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research. IEEE; p. 1–7.
9. Kavitha, P., & Usha, M. (2014). Cluster based anomaly detection in wireless LAN. *International Journal of Computer Trends and Technology (IJCTT)., 12*(5), 227–230.
10. Yavuz, F. Y., Devrim, Ü., & Ensar, G. (2018). Deep learning for detection of routing attacks in the internet of things. *International Journal of Computational Intelligence Systems., 12*(1), 39.
11. Yuan, Y., Li, S., Zhang, X., & Sun, J. (2018). A comparative analysis of svm, naive bayes and gbdt for data faults detection in wsns. In: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE; pp. 394–399.
12. Napiah, M. N., Idris, M. Y. I. B., Ramli, R., & Ahmedy, I. (2018). Compression header analyzer intrusion detection system (CHA-IDS) for 6LoWPAN communication protocol. *IEEE Access., 6*, 16623–16638.
13. Ioulianou, P., Vasilakis, V., Moscholios, I., & Logothetis, M. (2018) A signature-based intrusion detection system for the internet of things. Information and Communication Technology Form. .
14. Shafique, U., Khan, A., Rehman, A., Bashir, F., & Alam, M. (2018). Detection of rank attack in routing protocol for Low Power and Lossy Networks. *Annals of Telecommunications., 73*(7), 429–438.
15. Verma, A., Ranga, V., & ELNIDS: Ensemble learning based network intrusion detection system for RPL based Internet of Things. In,. (2019). 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU). *IEEE, 2019*, 1–6.
16. Kumar, V., Das, A. K., & Sinha, D. (2021). UIDS: a unified intrusion detection system for IoT environment. *Evolutionary intelligence., 14*(1), 47–59.
17. Parra, G. D. L. T., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. *Journal of Network and Computer Applications., 163*, 102662.
18. Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access., 9*, 103906–103926.
19. Jan, S. U., Ahmed, S., Shakhov, V., & Koo, I. (2019). Toward a lightweight intrusion detection system for the internet of things. *IEEE Access., 7*, 42450–42471.
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter., 11*(1), 10–18.
21. Kulkarni, S. R., Lugosi, G., & Venkatesh, S. S. (1998). Learning pattern classification-a survey. *IEEE Transactions on Information Theory., 44*(6), 2178–2206.
22. Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics., 21*(3), 660–674.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sana Rabhi was born in Tunisia in 1994, she received her license and master's degrees in telecommunications from the national school of electronics and telecommunications (ENET'COM). Currently, she is a PhD candidate in the national school of electronics and telecommunications of Sfax. Her research interest is detecting anomalies in wireless networks.

Tarek Abbes obtained the Telecommunication Engineering degree from the High School of Communication, Tunis, in 2000. He got the DEA and the Ph.D. Diplomas in computer science from the University of Henri Poincaré, Nancy, France, in 2001 and 2004, respectively. He received the habilitation to conduct research from the National School of Engineers of Sfax in 2020. Dr. Tarek Abbes is an associate professor at the National School of Electronics and Telecoms of Sfax (ENET'COM). From 2017 to 2020, he acted as the Head of the Telecommunications Department at ENET'COM. He is conducting his research activities in the areas of security and networking, and IoT.

Faouzi Zarai received the Engineering Diploma, Master Diploma, and PhD in Information and Communication Technologies from the Engineering School of Communications (Sup'Com, Tunisia) in 2002, 2003, and 2007; respectively. He is also recipient of the habilitation degree in 2011. From 2002 to 2005 he has worked for the National Digital Certification Agency (NDCA, Tunisia). Since 2011, he serves on the editorial boards of the International Journal of Communication Systems. He published one book and 5 chapters and co-authored more than 80 papers that have been published in international journals and conferences. Currently, Dr. Zarai is serving as professor for the National School of Electronic and Telecommunications Sfax (ENET'COM). From 2008 to 2014, he has the Head of the Department of telecommunications at ENET'COM. Since 2016, he is Director of the research unit of News Technologies and Telecommunications Systems (NTS'COM). He is conducting research activities in the areas of security and Quality of services in news generations wireless networks LTE-Advanced PRO: authentication, IP Taceback, Seamless Mobility, Radio Resource Management.