Computers and Electrical Engineering 98 (2022) 107716

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques☆

Alaeddine Mihoub [a], Ouissem Ben Fredj [b], Omar Cheikhrouhou [c,f,*], Abdelouahid Derhab [d], Moez Krichen [e]

[a] *Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University, P.O. Box: 6640, Buraidah: 51452, Saudi Arabia*
[b] *Higher Institute of Applied Sciences and Technology of Kairouan, University of Kairouan, Tunisia*
[c] *Higher Institute of Computer Science of Mahdia, University of Monastir, Mahdia 5111, Tunisia*
[d] *Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 11653, Saudi Arabia*
[e] *Faculty of CSIT, Al-Baha University, Saudi Arabia, & ReDCAD Laboratory, University of Sfax, Tunisia*
[f] *CES Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia*

ARTICLE INFO

Keywords:
Cybersecurity
DoS
DDoS
IoT
Machine Learning
Looking-Back method

ABSTRACT

IoT (Internet of Things) systems are still facing a great number of attacks due to their integration in several areas of life. The most-reported attacks against IoT systems are "Denial of Service" (DoS) and "Distributed Denial of Service" (DDoS) attacks. In this paper, we investigate DoS/DDoS attacks detection for IoT using machine learning techniques. We propose a new architecture composed of two components: DoS/DDoS detection and DoS/DDoS mitigation. The detection component provides fine-granularity detection, as it identifies the specific type of attack, and the packet type used in the attack. In this way, it is possible to apply the corresponding mitigation countermeasure on specific packet types. The proposed DoS/DDoS detection component is a multi-class classifier that adopts the "Looking-Back" concept, and is evaluated on the Bot-IoT dataset. Evaluation results show promising results as a Looking-Back-enabled Random Forest classifier achieves an accuracy of 99.81%.

1. Introduction

Internet of Things (IoT) is still emerging as a promising technology and it is more integrated with our daily life applications including smart transportation [1], smart healthcare [2], smart home [3], etc. The installed IoT devices worldwide are projected to reach 30.9 billion units by 2025 [4]. These IoT devices do not possess high computation power and do not have appropriate security mechanisms while they are connected to the Internet. This vast integration of IoT technology has made them an interesting target of attackers [5]. Two of the most challenging attacks in IoT are (1) Denial of Service (DoS) and (2) Distributed Denial of Service (DDoS). The problem is how to detect DoS/DDoS attacks, which are characterized by a high packet rate from one IP (for DoS) or multiple IPs (for DDoS), while normal traffic could also encompass a high traffic rate especially from environments equipped with IoT sensors. Cyber-attacks are considered as an obstacle to the spread of IoT devices. Therefore, the need for anomaly detection methods is of high

☆ This paper is for special section VSI-spc. Reviews were processed by Guest Editor Dr. Atilla Elci and recommended for publication.

* Corresponding author.
E-mail address: omar.cheikhrouhou@isetsf.rnu.tn (O. Cheikhrouhou).

https://doi.org/10.1016/j.compeleceng.2022.107716
Received 2 June 2021; Received in revised form 13 December 2021; Accepted 7 January 2022
Available online 18 January 2022

priority. Several Intrusion Detection Systems (IDSs) have been proposed in the literature [6]. They can be categorized in Signature-based IDS, Anomaly-based IDS, or Hybrid IDS. Signature-based IDS checks the traffic against predefined attack patterns and, therefore they are unable to detect zero-day attacks. However, Anomaly-based IDS is capable of identifying unseen attacks by identifying anomalies pertaining to high network latency, traffic on unusual ports, and high network volume, etc.

In this paper, we propose an architecture that captures and mitigates DoS/DDoS attacks for the Internet of Things. In particular, the main contributions of this paper are the following:

- We design and implement an architecture that is composed of two main components: DoS/DDoS detection and DoS/DDoS mitigation, where the mitigation countermeasures are based on the detection decision.
- Different from previous works tested on Bot-IoT, which focused on distinguishing between normal traffic and attack traffic [7–19], we adopt a different detection approach, which takes into consideration the cyber security analyst perspective, and aims at identifying the subcategories of DoS/DDoS, and consequently identifying the attacks and their corresponding mitigation countermeasures at a fine-grained level.
- We identify six subcategories of attacks by combining two attack categories: DDoS and DoS, and three types of packets: TCP, UDP, and HTTP.
- We design the proposed detection classification approach by combining the Looking-Back concept with basic classifiers.
- Based on the detection results, we apply the following mitigation countermeasures:
  a. In case of DoS attack, we deny specific traffic: HTTP, TCP, or UDP from one IP address, and allow the rest of the traffic.
  b. In case of DDoS, we apply rate-limiting on specific traffic: HTTP, TCP, or UDP, and allow the rest of the traffic.
- We evaluate the performance of the proposed DoS/DDoS detection component on the Bot-IoT dataset, and compare it with basic classifiers. The evaluation results show promising results as Looking-Back-enabled Random Forest [20] records an accuracy of 99.81%.

The rest of the paper is organized as follows. Section 2 presents related work. The description of DoS/DDoS detection and mitigation architecture is given in Section 3. Section 4 describes the implementation of the proposed architecture. Evaluation results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related work

There are many works that aim to capture and mitigate both DoS and DDoS attacks in IoT systems. In the following, we present some of them.

Koroniotis et al. [7] applied three techniques; namely Support Vector Machine (SVM), Recurrent Neural Network (RNN), and Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) to the Bot-IoT dataset. The experiment results show that the SVM classifier has the highest accuracy and recall rates when using the 46 features or 10 best features.

Kumar et al. [8] used smart contracts provided by Ethereum and the proliferation of IoT devices in order to study the DDoS attacks in blockchain-IoT systems, especially in smart contracts. The authors used two AI techniques, namely random forest and XGBoost which give complete autonomy in capabilities of decision-making. They also proposed a distributed intrusion detection framework using fog computing tehchniques. The system is evaluated against the BoT-IoT dataset using the 10 best features and all the features of the dataset. They achieved a detection rate of around 99.99% using Random forest with 10 features.

Bhuvaneswari and Selvakumar [9] addressed anomaly detection for IoT in a fog environment with scalability. They used the VCDL (Vector Convolutional Deep Learning) methodology for learning. The VCDL models are based on Convolutional Neural Networks with a pair of levels: PL (Pooling Layer) and CL (Convolutional Layer). It encompasses a pair of modules: VCN (Vector Convolutional Network) and FCN (Fully Connected Network). The role of the first module is to extract the features. While the role of the second module consists in learning the extracted features for detecting the class of the IoT traffic. The hyper-parameters adopted to build the VCDL structure contain the numbers of PLs and CLs, HL (Hidden Layer) in the FCN, and nodes in the HLs. The presented system is assessed against the BoT-IoT dataset considering the ten best features and all the features of the dataset.

Alkadi et al. [10] developed DBF (Deep Blockchain Framework) that allows distributed intrusion detection using BiLSTM (Bidirectional Long Short-Term Memory) deep learning techniques that deals with sequential network data of the BoT-IoT dataset. The system also provides a blockchain with smart contracts of Ethereum in IoT networks to insure privacy to the distributed intrusion detection engines. The results of the DBF is evaluated with respect to other machine learning techniques; like: Mixture Localisation-based Outliers, Random Forests, Support Vector Machines and Naive Bayes [11], and has demonstrated the best false alarm rate.

Huong et al. [12] proposed an edge-cloud architecture that detects attacks at the edge layer, near the source of the attacks. This insures quick response, versatility, and reducing the cloud's workload. They also proposed a multi-attack detection mechanism named LocKedge that has reduced deployment complexity at edge zones and maintains high precision. To do so, they applied data preprocessing in order to take only numerical input. Then, they extracted the most important features using the PCA (Principal Components Analysis) technique. LocKedge implements the ADAM algorithm to optimize loss function. They compared their system with popular ML techniques such as Support Vector Machines, Random Forests, K-nearest neighbors and Decision Trees, using the BoT-IoT dataset. In general, LocKedge demonstrates good average detection rate higher than the other techniques in most classes. It is also worth noticing that the authors of [7] applied multiple binary models. In this case, each model is considered normal versus one type of attack which raises many issues. First, the execution time will be much longer as each model needs to be executed separately. Second, it

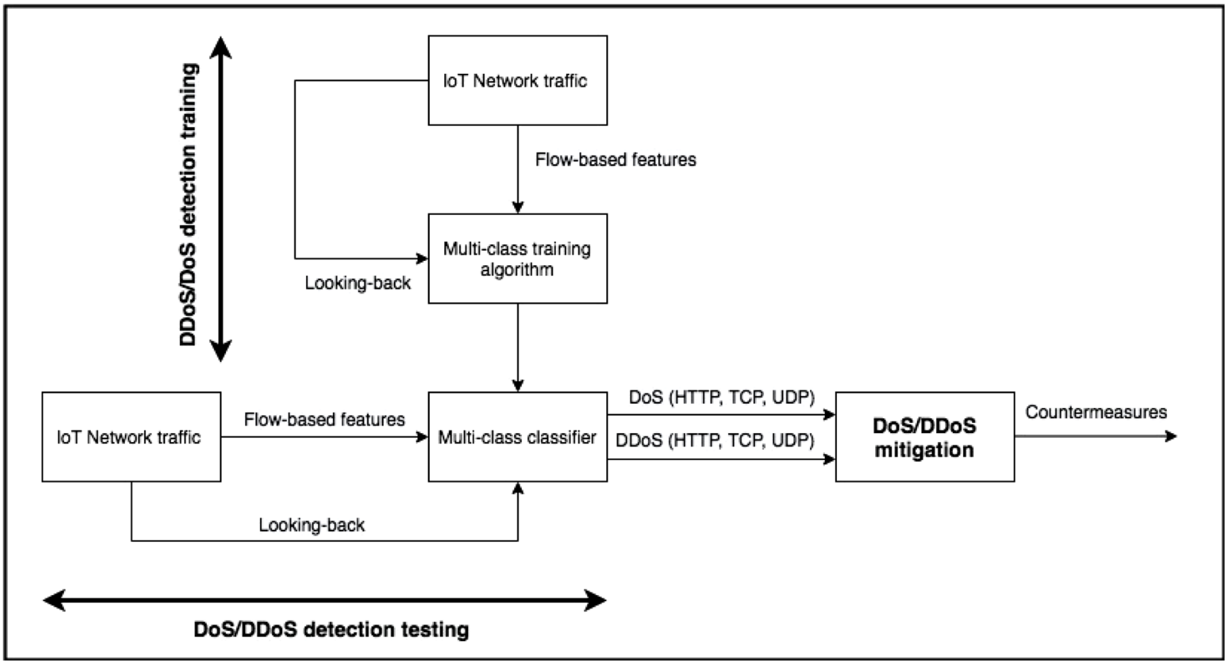


Fig. 1. DoS/DDoS detection and mitigation architecture.

is not clear how to aggregate the multiple predictions coming from the different models especially when they detect different types of attacks.

Galeano-Brajones et al. [13] focused on 5 G, and they took advantage of SDN (Software-Defined Networks) and NFV (Network Function Virtualization) as they are the main components of 5G SDN paradigm decouples control planes whichare based on softwares from data planes which are based on hardwares. NFV is a network approach decouples network functions from the underlying hardware by replacing expensive, dedicated, and proprietary network hardware with software-based network devices. The authors proposed an entropy-based technique which works in a proper manner for stateful SDN architectures in order to detect attacks. More reduced values of the entropy mean higher incidence for a source IP appearance.

Khraisat et al. [14] presented HIDS (Hybrid IDS) which combines SIDS (Signature-based IDS) and AIDS (Anomaly-based IDS). The SIDS uses the C5 decision tree classifier and it is used as a first step of attacks detection. Pattern matching was adopted in order to handle unknown traffic. If the request is not raised as an attack, it will be forwarded to the AIDS. For training AIDS, a Single-Class SVM is utilized, which learns the attributes of normal packets only. It does not train from the other classes. Thus, the AIDS supposes that any outliers which are outside the normal behavior, are identified as a zero-day attacks. In the third stage, ensemble methods are utilized for enhancing prediction precision. By applying the information gain feature selection method, 13 features out of 43 features from the BoT-IoT dataset are selected. The proposed HIDS reached 99.97% as an accuracy.

Ge et al. [15] developed a FFNN (Feed-Forward Neural Network) model for classification including DOS, DDOS, reconnaissance, and information theft assaults against IoT nodes. The work focuses on the Message Queuing Telemetry Transport (MQTT) [21], which is an IoT connectivity protocol and is responsible for publish-subscribe facilities between IoT devices and centralized brokers or base stations. The model consists of three dense hidden layers with 512 neurons for the first 2 layers and 2/4 neurons with softmax function in the third layer depending on the number of classes. In the multiclass classification, the detection accuracy is 99.41% for DoS/DDoS assaults while the classification of the normal traffic reaches a precision of 98%.

In this paper, we explore classic machine learning techniques such as decision tree classifier, Random Forest, SVM, etc. as well as newer deep learning models, all coupled with the Looking-Back approach. Different from previous works, which focus on distinguishing between normal traffic and multiple categories of attack traffic, we aim in this work at identifying the subcategories of DoS/DDoS attacks, and consequently identifying the attacks and their corresponding mitigation countermeasures at a fine-grained level. More precisely, we identify six subcategories of attacks by combining two attack categories: DDoS and DoS, and three types of packets: TCP, UDP, and HTTP.

Moreover, in this paper, we aim to study the impact of attack history on the detection accuracy. For this purpose, we introduce the concept of Looking-Back that takes into consideration the previous attack type in the detection decision of the current attack type.

3. DoS/DDoS detection and mitigation architecture

We present in this section the Looking-Back approach, and describe the proposed architecture which is composed of two components: DoS/DDoS detection and DoS/DDoS mitigation, as shown in Fig. 1.

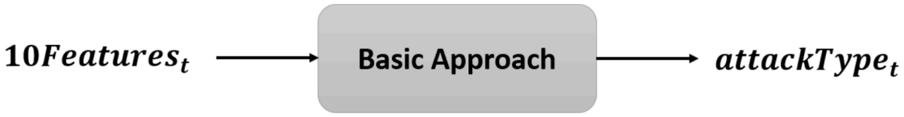


Fig. 2. The Basic Approach: the model in this approach takes as inputs the 10 selected features of the instant t. The target output is the attack type of time t.

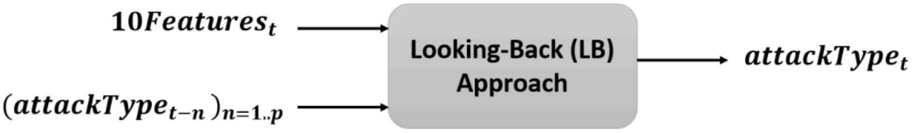


Fig. 3. The Looking-Back (LB) Approach.: the model in this approach takes as inputs (1) the 10 selected features of the instant t and (2) the list of previous attacks types. The target output is the attack type of time t.

3.1. The looking-back approach

The aim of this work is to predict the type of cyber-attacks, and more precisely, the subcategories of DoS and DDoS attacks. To this end, we develop several approaches ranging from conventional machine learning methods such as KNN, Decision trees, and Random Forest to more sophisticated deep learning methods such as MLP, RNN, and LSTM. All these models are explored by considering two configurations with regards to input features. The first configuration is named the "Basic Approach" and the second one is the "Looking-Back Approach" ("LB Approach"). We remind that 10 best features from the "BoT-IoT" dataset were selected according to [7]. The "Basic Approach" uses those 10 inputs at time t (named here10Features$_t$) to detect the type of attack at that time t (named hereattackType$_t$). In addition of the 10Features$_t$ inputs, the "Looking-Back Approach" uses the list of previous attacks till the time t-p, where p is the Looking-Back time step. This list can be described as follows:

$$\text{List of the previous attacks} = \left(\text{attackType}_{\text{t}-\text{n}}\right)_{\text{n}=1.\text{p}}$$

For instance, if $p = 3$, the model takes as inputs 10Features$_t$, attackType$_{t-1}$, attackType$_{t-2}$ and attackType$_{t-3}$. Please note that this approach is not restricted to DoS/DDoS attacks but may be applied to any type of sequential attacks since it relies on the idea of potential temporal patterns in the cyber-attacks.

In this way, the first basic model has one type of inputs represented by the 10Features$_t$, while the second LB model has 2 types of inputs (10Features$_t$ and the list of previous attack types), as shown in Fig. 2 and Fig. 3.

3.2. DoS/DDoS detection

DoS/DDoS detection component operates on 2 phases, as illustrated in Fig. 1: training and testing. In the training phase, the following operations are performed:

- The flow-based features are extracted from IoT network traffic using some protocols like sflow [22].
- A Looking-Back-enabled multi-class classifier is trained. This classifier has the following properties:
  a. Any learning algorithm can be used to train the classifier whether it is conventional machine learning or a deep learning algorithm.
  b. The training algorithm considers the classes corresponding to the subcategories of DoS and DDoS attacks. In particular, we consider six classes, namely: DoS-UDP, DoS-TCP, DoS-HTTP, DDoS-UDP, DDoS-TCP,DDoS-HTTP, where UDP, TCP, and HTTP, are the packet types employed in the attack.
  c. The multi-class training algorithm leverages the Looking-Back concept, which uses the list of previously detected attacks till time t-p, where t and p denote the detection time and the Looking-Back step. This list and the flow-based features are fed together to the multi-class training algorithm to generate a Looking-Back-enabled multi-class classifier.

During the testing stage, the following operations are made:

- The flow-based features are extracted from network traffic using some protocols like sflow [22].
- The extracted flow-based features and the list of the previously detected attacks till time t-p are fed to the multi-class classifier, where t and p denote the detection time and the Looking-Back step.
  The prediction result of the classifier, i.e., attack type, is sent to the mitigation component.

Table 1
The summary of the reduced dataset.

	Training		Testing		Generation Tools
	Number of packets	% packets	Number of packets	% packets	
DDoS_HTTP	786	0.02	203	0.02	golden-eye
DDoS_TCP	782,228	26.65	195,152	26.59	hping3
DDoS_UDP	758,301	25.834	189,954	25.88	hping3
DoS_HTTP	1184	0.04	301	0.04	golden-eye
DoS_TCP	492,615	16.78	123,185	16.78	hping3
DoS_UDP	826,349	28.15	206,626	28.16	hping3
Total	2861,463	97.49	715,421	97.50	

Table 2
Features extracted from the IoT-Bot dataset.

Feature	Description
Srate	Source-to-destination packets per second
Drate	Destination-to-source packets per second
Max	Maximum duration of aggregated records
state_number	Numerical representation of feature state
Mean	Average duration of aggregated records
Min	Minimum duration of aggregated records
Stddev	Standard deviation of aggregated records
Seq	Argus sequence number
N IN Conn P SrcIP	Number of inbound connections per source IP
N IN Conn P DstIP	Number of inbound connections per destination IP

3.3. DoS/DDoS mitigation

Based on the result of the DoS/DDoS detection component, the following mitigation countermeasures are taken:

- If the predicted attack is DoS-TCP, DoS-HTTP, or DoS-UDP, then, TCP, HTTP, or UDP packets respectively are denied from one IP address, and the rest of packets that are generated from the same IP address are allowed.
- If the predicted attack is DDoS-HTTP, DDoS-TCP, or DDoS-UDP, then rate-limiting is applied on HTTP, TCP, or UDP packets respectively, and the rest of packets are allowed.

4. Implementation

4.1. Dataset

The proposed IoT-Bot [7] testbed was designed at the Research Cyber Range Laboratory of the University of New South Wales Canberra. The architecture is an IoT network environment composed of 3 main elements: 1. network platforms, 2. simulated IoT services, and 3. feature extraction platform. The network platforms are either normal or attacking VMs. Several VMs form a cluster and they are linked to the Internet through the well-known firewall PFSense. The second part of the network platform is an Ubuntu VM platform. The later runs the Node-red tool [23] for the simulation of different IoT sensors that were connected with AWS (the public IoT hub).

The simulated IoT services mimic the IoT network behaviors and include IoT components like smart fridges, a garage doors, weather stations, motion-activated lights, and smart thermostats. The last platform is the feature extraction platform which uses the systems and network monitoring application called Argus for extracting related data features [7]. The dataset includes approximately fourty-six features and seventy-two million records. In our experimentation, we used a reduced dataset which is made of about 3.6 million records, representing 5% of the whole data provided by the same team who designed the testbed. More precisely, the adopted dataset contains 477 normal instances and 3668,118 attack instances. Table 1 gives a summary of the reduced 5% of the data:

The dataset contains simulated IoT network traffic. Most of the traffic is DoS (44.99%) and DDoS (52.51%) with an overall percentage of 97.50% The remaining traffic is classified as normal, information theft attack, or information gathering attack.

The authors identified the following best 10 features (described in Table 2): srate, drate, max, state_number, mean, min, stddev, seq, N_IN_Conn_P_SrcIP, and N_IN_Conn_P_DstIP. As previously mentioned, only these 10 best-selected features are used in our experimentations.

4.2. Machine learning models

4.2.1. Decision trees

Decision Trees represent a non-parametric supervised learning technique that may be adopted for regression/classification purposes. The objective is to develop a model in the form of a tree which predicts the values of target variables by learning simple decision rules extracted from the data features. This technique does not require complex data preparation like many other techniques. Moreover, it is fairly easy to understand/interpret since trees can be visualized.

4.2.2. Random forests

Random forest [20] is an easy-to-use supervised learning technique that gives excellent results, in most cases, even without hyper-parameter adjustment. Because of its simplicity and versatility, it is one of the most widely used algorithms. It can be used for classification as well as regression. It creates a "forest" out of an ensemble of decision trees, which are commonly trained using the "bagging" technique.

4.2.3. KNN

The K-Nearest Neighbors (KNN) technique [24] is a supervised machine learning technique which is simple and easy-to-implement technique. It may be used to address both regression and classification issues. The KNN algorithm believes that objects that are similar are close together. To put it another way, related items are close together. This algorithm saves all available data and classifies a new data point based on its resemblance to the existing data. This means that as new data comes, it may be quickly sorted into one of the well-suited categories.

4.2.4. MLP

The multi-layer perceptron (MLP) [25] is a technique that can approximate any continuous function and handle problems that aren't linearly separable. Pattern classification, recognition, approximation, and prediction are some of MLP's most common applications. MLP has three layers: one input layer, one output layer, and one/multiple hidden layers. The input signal is received by the input layer. The output layer is responsible for tasks such as classification and prediction. Finally, the hidden layers, placed between the input and output layers, correspond to the true computational engine of the MLP algorithm.

4.2.5. LSTM

Long short-term memory (LSTM) [26] is an artificial RNN (Recurrent Neural Network) deep learning architecture. The latter has feedback connections, unlike normal feedforward neural networks. It may process not only single data points (like photos) but also complete data sequences (like videos). For instance, activities like speech recognition, unsegmented, connected handwriting identification, and or intrusion detection systems (IDSs), or anomaly detection in network traffic can all benefit from LSTM [27, 28].

4.3. Model parameters

The previously described techniques were used for predicting the correct class of attack for every data sample. A min-max scaler was used for normalizing feature ranges between 0 and 1 before classification. For some classification algorithms, this operation is highly recommended, if not required. Please notice that all preprocessing, training, and testing operations were carried out in Python and well-known data science libraries like Keras, Scikit-learn, and Pandas. Classification results were obtained on a 16GB RAM PC powered by an Intel® CoreTM i7–8550 U processor. Steps for hyper-parameters tuning were investigated in order to determine optimal hyper-parameters for each algorithm and to avoid overfitting problems.

Indeed, the "gini" criterion was adopted for the quality of split in the Decision Tree algorithm. 100 was the best number of estimators for the Random Forest approach. We remind that RF is an ensemble method since it combines the predictions of several estimators. For KNN, 10 was the optimal number of neighbors giving the best outcomes. Moreover, because of their good performance, the rest of hyper-parameters (for these three algorithms) were set to default values as defined in the Scikit-learn package. For Scikit-learn, please notice that default parameters were selected based on the best findings in the literature.

Furthermore, for all deep learning models (MLP, RNN, and LSTM), a topology with two hidden layers separated by a dropout layer was found to be the most relevant architecture. The dropout layer was added to prevent overfitting problems. Each hidden layer is composed of either "50 dense nodes" or "50 RNN nodes" or "50 LSTM memory cells" depending on the used model. Similarly, for all approaches, the "relu" activation function is chosen for the intermediate layers while the 'softmax' function is adopted for the output layer. The optimizer was set the "adadelta" type and the loss parameter was set to the "categorical_crossentropy" function. For MLP, 15 epochs were needed to obtain the best results while RNN and LSTM have needed 25 epochs. The rest of the parameters was set to the default values as defined in the Keras library.

5. Performance evaluation

5.1. Evaluation metrics

5.1.1. Accuracy and F1-measure

Accuracy represents the fraction of correctly classified instances out of all samples. The F1-measure is another popular metric to

Table 3
Accuracy results (%) for all classifiers using multiple Looking-Back steps (from 0 to 5).

	Accuracy (%)					
Classifier/LB Steps	**Basic Approach**			**Looking-Back Approach**		
	0	**1**	**2**	**3**	**4**	**5**
DT	97.58	98.24	98.24	98.18	98.09	**98.35**
RF	99.75	99.76	99.78	99.77	**99.81**	99.80
KNN	**99.93**	99.54	98.37	97.02	95.45	91.67
MLP	**99.14**	99.01	98.57	98.41	97.97	98.06
RNN	**98.98**	98.40	98.64	98.58	97.91	96.92
LSTM	98.79	**99.11**	98.91	98.92	98.68	98.60

Table 4
Kappa results (%) for all classifiers using multiple Looking-Back steps (from 0 to 5).

	Kappa (%)					
Classifier/LB Steps	**Basic Approach**			**Looking-Back Approach**		
	0	**1**	**2**	3	4	**5**
DT	96.74	97.63	97.63	97.55	97.43	**97.78**
RF	99.67	99.67	99.70	99.69	**99.75**	99.74
KNN	**99.91**	99.39	97.80	95.98	93.87	88.77
MLP	**98.85**	98.67	98.08	97.86	97.27	97.39
RNN	**98.63**	97.85	98.17	98.09	97.17	95.84
LSTM	98.38	**98.80**	98.53	98.55	98.23	98.11

evaluate classification results. It computes the harmonic mean between Precision and Recall metrics as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

In the classification terminology, TP, FP, and FN stand for True Positives, False Positives, and False Negatives, respectively. When evaluating our models, we adopted the "micro" variant for both Precision and Recall. These "micro" variants compute rates by considering the total True Positives (total TP), total False Positives (total FP), and total False Negatives (total FN). In the multiclass prediction, both "total FP" and "total FN" correspond empirically to the same number of total misclassifications (the total number of errors). Consequently, the micro-Precision is found to be equivalent to the micro-Recall, equivalent to the micro-F1, and also equivalent to the Accuracy. Therefore, in the rest of this paper, we will focus essentially on presenting Accuracy figures.

5.1.2. Kappa

The kappa index is a statistic method originally used to measure inter-annotator agreement. It is commonly considered to be a more rigorous metric than the simplistic percent agreement estimation since it accounts for the probability of agreement happening by accident. The most frequently used variant of Kappa is Cohen's kappa. In a classification context, it is used to evaluate the agreement between the predicted classes and actual ones. The Cohens' kappa is expressed as follows:

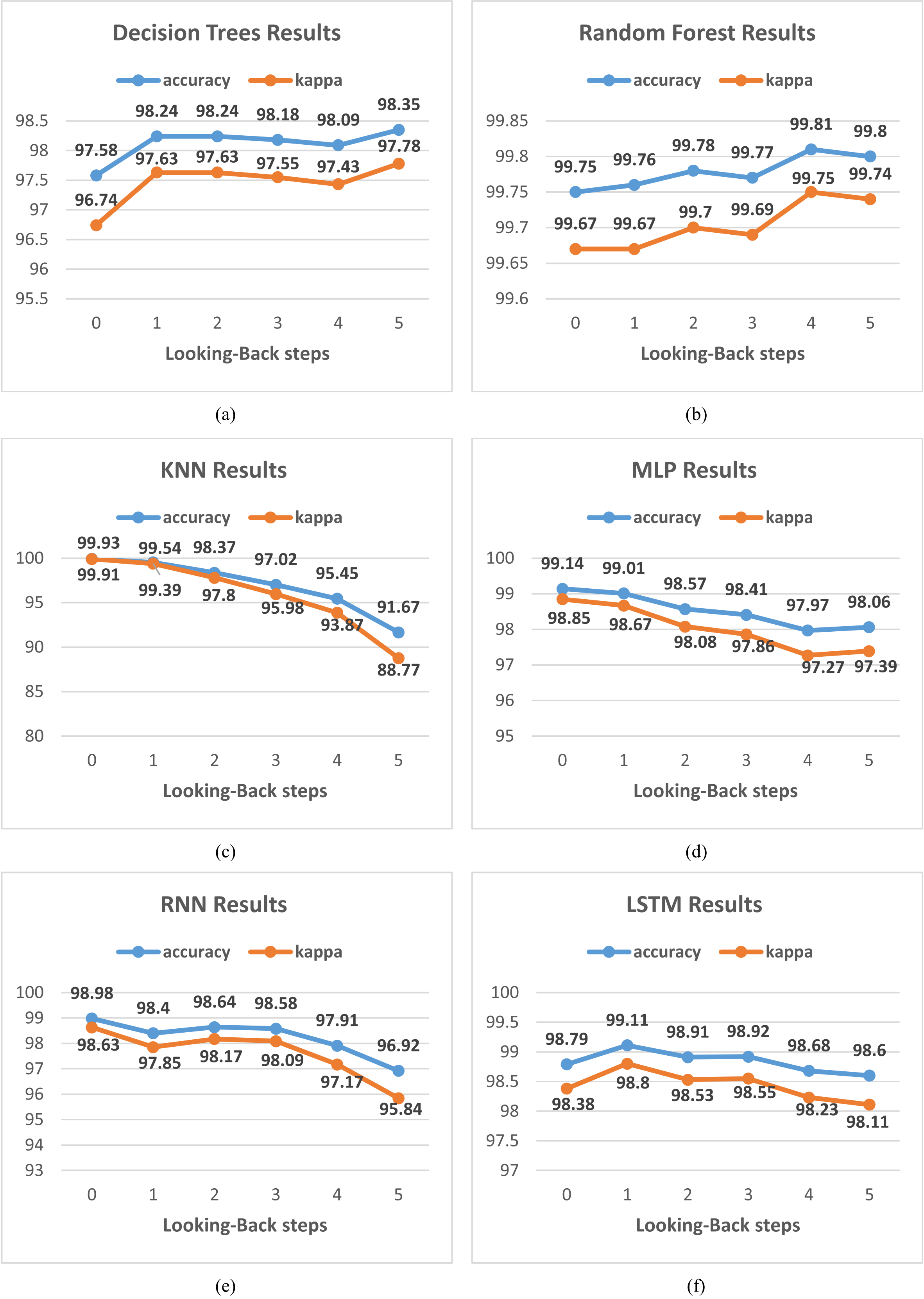
$$Cohen^{'}s\ kappa = 1 - \frac{1 - p_0}{1 - p_e}.$$

Here, p_0, similar to the precision metric, represents the percentage of empirical agreement between the estimated and the actual classes. For p_e, it represents the expected probability of a random classification agreement. The perfect classification gives a Kappa equal to 1, and if no classification agreement exists other than what can be predicted by chance, the Kappa will be equal to 0.

5.2. Results and discussion

Six classifiers were tested: three classifiers based on classic machine learning approaches namely Decision Trees, Random Forest, K-Nearest Neighbors, and three classifiers based on deep learning approaches namely Multi-Layer Perceptron, Recurrent Neural Networks and Long Short Term Memory. All models were tested for both basic and Looking-Back approaches.

Accuracy and Kappa metrics were used to evaluate these algorithms, as presented in Table 3,



(caption on next page)

Fig. 4. Plots of Accuracy and Kappa results over multiple Looking-Back steps (from 0 to 5) for all classifiers as follows: (a) Decision Trees results, (b) Random Forest results, (c) KNN results, (d) MLP results, (e) RNN results, and (f) LSTM results.

Table 5
Classifiers were organized from left to right according to the best-recorded accuracies. The used LB step is mentioned within the parenthesis.

	Best Configurations					
Best accuracies in order	99.93	99.81	99.14	99.11	98.98	98.35
Corresponding Classifier (LB step)	KNN (0)	RF (4)	MLP (0)	LSTM (1)	RNN (0)	DT (5)

Table 6
Confusion matrix of the KNN classifier (LB step = 0, Accuracy = 99.93%).

KNN (*LBstep* = 0)		**Predicted**					
		DDoS_HTTP	DDoS_TCP	DDoS_UDP	DoS_HTTP	DoS_TCP	DoS_UDP
Actual	DDoS_HTTP	**190**	**10**	**0**	**3**	**0**	**0**
	DDoS_TCP	**0**	**194,986**	**0**	**0**	**166**	**0**
	DDoS_UDP	**0**	**3**	**189,941**	**0**	**0**	**10**
	DoS_HTTP	**5**	**16**	**0**	**278**	**2**	**0**
	DoS_TCP	**0**	**265**	**0**	**0**	**122,920**	**0**
	DoS_UDP	**0**	**3**	**8**	**0**	**0**	**206,615**

Table 7
Confusion matrix of the RF classifier (LB step = 4, Accuracy = 99.81%).

RF (*LBstep* = 4)		**Predicted**					
		DDoS_HTTP	DDoS_TCP	DDoS_UDP	DoS_HTTP	DoS_TCP	DoS_UDP
Actual	DDoS_HTTP	**201**	**2**	**0**	**0**	**0**	**0**
	DDoS_TCP	**0**	**194,661**	**0**	**1**	**487**	**1**
	DDoS_UDP	**0**	**1**	**189,949**	**0**	**0**	**3**
	DoS_HTTP	**2**	**1**	**0**	**298**	**0**	**0**
	DoS_TCP	**0**	**820**	**1**	**0**	**122,364**	**0**
	DoS_UDP	**0**	**0**	**0**	**0**	**5**	**206,620**

Table 8
Confusion matrix of the MLP classifier (LB step = 0, Accuracy = 99.14%).

MLP (*LBstep* = 0)		**Predicted**					
		DDoS_HTTP	DDoS_TCP	DDoS_UDP	DoS_HTTP	DoS_TCP	DoS_UDP
Actual	DDoS_HTTP	**109**	**49**	**0**	**45**	**0**	**0**
	DDoS_TCP	**0**	**190,287**	**6**	**7**	**4851**	**1**
	DDoS_UDP	**0**	**3**	**189,627**	**0**	**3**	**321**
	DoS_HTTP	**42**	**53**	**0**	**177**	**29**	**0**
	DoS_TCP	**0**	**696**	**0**	**0**	**122,488**	**1**
	DoS_UDP	**0**	**1**	**21**	**0**	**3**	**206,601**

Table 4, and Fig. 4. The best accuracy was 99.93% obtained by The KNN classifier without any Looking-Back step (LB step = 0). The second-best classifier was the RF with an accuracy rate equal to 99.81% and a LB step = 4. The third best classifier was the MLP with an accuracy rate equal to 99.14% and a LB step = 0. Similarly, the best Kappa results were obtained by KNN (99.91%), RF (99.75%), and MLP (98.85%). The confusion matrices of these three best classifiers are presented in Tables 6-8. Confusion matrices show the relevance of our models in detecting the right classes. Nevertheless, some errors had occurred especially for the classes DDos_HTTP and DoS_HTTP. This is due mainly to the limited number of occurrences of these classes in the training set compared to other classes. In addition, examinating DDoS_TCP and DoS_TCP detection results show some mutual errors between these two particular classes, which may be explained by the similarity of their data.

Moreover, seeing all classifiers and comparing the basic approach (LB step = 0) with the Looking-Back approaches (LB steps > 0) shows an advantage for the basic approach, as shown in

Table 5. This finding may be explained by the eventual absence of temporal relationships between attacks. The absence of temporal correlation is probably due to the artificial aspect of the dataset. We remind that the used data were generated by machines and do not necessarily reflect elaborated scenarios and real serial attacks. This remark explains also why robust temporal deep learning models such as RNN and LSTM were outperformed by more classic approaches such as MLP, RF, and KNN. Furthermore, looking at Fig. 4, especially (c), (d), (e), and (f) subplots, all accuracies of the corresponding models were negatively impacted when increasing the LB

Table 9
Training Testing Time (in seconds) for all classifiers using multiple Looking-Back steps (from 0 to 5).

	Training/Testing Time (Sec)											
LB Steps	0		1		2		3		4		5	
Classifier/Setting	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
DT	28	0.07	31	0.08	32	0.08	34	0.08	37	0.08	39	0.08
RF	803	9	832	11	917	12	904	15	937	17	968	17
KNN	1743	90	9073	215	7434	376	5604	538	5154	553	5036	521
MLP	1180	5	1245	5	1488	7	1361	8	1184	6	1312	5
RNN	2451	7	2599	11	2507	9	2607	9	2495	7	2517	9
LSTM	3264	9	4024	12	4006	11	3309	9	3623	10	4372	14

step parameter. This decrease is remarkably observed for the KNN classifier (from 99.93 to 91.67). The LB approach was only beneficial to the DT classifier, which presents lower performance than other models. Even for the RF model, the contribution of the LB approach is very low: with an LB step = 0, the accuracy is 99.75% while using an LB step = 4, the accuracy was slightly ameliorated to 99.81%.

Besides accuracy and kappa metrics, both training and testing time were recorded for all classifiers in all configurations, as shown in

Table 9. Firstly, as expected, deep learning models spent more time to ensure the training. For instance, giving LB step=0, 3264 s were needed for the LSTM training, 2451 s for the RNN model versus only 28 s for the DT model. Secondly, looking at the KNN model, although it gives the best accuracy, it seems to be the longest model in terms of training and testing times. For LB step=0, the KNN model spent nearly 90 s for testing while all other models' timings were inferior to 10 s. It also recorded the longest time in training namely 9073 s for LB step = 1. Thirdly, the DT model, despite giving the lowest accuracy, presents the best timing scores for both training and testing. For instance, in all configurations, its testing times were all inferior to 1 s. Overall, the RF model seems to be the best tradeoff between accuracy/kappa metrics and training/testing times. While the accuracy range was between 99.75 and 99.81, the maximum training time was near 968 s and the maximum testing time was about 17 s.

Conclusion

In this paper, we have proposed an architecture which is designed for detecting and mitigating DoS/DDoS attacks for Internet of Things. The DoS/DDoS detection provides fine-granularity detection, as it identifies the type of attack: DDoS or DoS, and the packet type used in the attack. Based on the prediction attack result, we have applied the corresponding mitigation countermeasure. To detect DoS/DDoS attacks, we have a multi-class classifier that adopts the Looking-Back concept, and is evaluated on Bot-IoT dataset. Evaluation results have shown promising results as Looking-Back-enabled Random Forest achieves an accuracy of 99.81%. As future work, we plan to apply the proposed architecture on other IoT datasets. Also, it would be interesting to test the resiliency of the architecture against adversarial learning attacks.

Declaration of Competing Interest

The authors declare no conflict of interest.

References

[1] Mershad K, Cheikhrouhou O, Ismail L. Proof of accumulated trust: a new consensus protocol for the security of the IoV. Veh Commun Dec. 2021;32:100392. https://doi.org/10.1016/j.vehcom.2021.100392.

[2] Frikha T, Chaari A, Chaabane F, Cheikhrouhou O, Zaguia A. Healthcare and fitness data management using the IoT-based blockchain platform. J Healthcare Eng Jul. 2021;2021:e9978863. https://doi.org/10.1155/2021/9978863.

[3] Mihoub A. A deep learning-based framework for human activity recognition in smart homes. Mobile Inf Syst Sep. 2021;2021. https://doi.org/10.1155/2021/6961343.

[4] 'Global IoT and non-IoT connections 2010-2025'. Statista 2021. https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/. accessed May 29.

[5] Ben Fredj O, Mihoub A, Krichen M, Cheikhrouhou O, Derhab A. CyberSecurity attack prediction: a deep learning approach. In: 13th International Conference on Security of Information and Networks; 2020. p. 1–6.

[6] Jemal I, Haddar MAMA, Cheikhrouhou O, Mahfoudhi A. Performance evaluation of convolutional neural network for web security. Comput Commun Jul. 2021; 175:58–67. https://doi.org/10.1016/j.comcom.2021.04.029.

[7] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: bot-IoT dataset. Future Generat Comput Syst Nov. 2019;100:779–96. https://doi.org/10.1016/j.future.2019.05.041.

[8] Kumar P, Kumar R, Gupta GPGP, Tripathi R. A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT systems by leveraging Fog computing. Trans Emerg Telecommun Technol 2022:e4112. https://doi.org/10.1002/ett.4112. n/a, no. n/a.

[9] B. A. N.g. and S. S.. Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment. Future Generat Comput Syst Dec. 2020;113:255–65. https://doi.org/10.1016/j.future.2020.07.020.

[10] Alkadi O, Moustafa N, Turnbull B, Choo K-KRKR. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet Things J 2020:1. https://doi.org/10.1109/JIOT.2020.2996590.

[11] AlKadi O, Moustafa N, Turnbull B, Choo K-KRKR. Mixture localization-based outliers models for securing data migration in cloud centers. IEEE Access 2019;7: 114607–18. https://doi.org/10.1109/ACCESS.2019.2935142.

[12] Huong TTTT, et al. LocKedge: low-complexity cyberattack detection in IoT edge computing. IEEE Access 2021;9:29696–710. https://doi.org/10.1109/ACCESS.2021.3058528.
[13] Galeano-Brajones J, Carmona-Murillo J, Valenzuela-Valdés JFJF, Luna-Valero F. Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: an experimental approach. Sensors Jan. 2020;20(3). https://doi.org/10.3390/s20030816. Art. no. 3.
[14] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A. A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics Nov. 2019;8(11):11. https://doi.org/10.3390/electronics8111210. Art.
[15] Ge M, Fu X, Syed N, Baig Z, Teo G, Robles-Kelly A. Deep learning-based intrusion detection for IoT networks. In: 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC); Dec. 2019. p. 256–25609. https://doi.org/10.1109/PRDC47002.2019.00056.
[16] Ferrag MAMA, Maglaras L. DeepCoin: a novel deep learning and blockchain-based energy exchange framework for smart grids. IEEE Trans Eng Manage Nov. 2020;67(4):1285–97. https://doi.org/10.1109/TEM.2019.2922936.
[17] Aldhaheri S, Alghazzawi D, Cheng L, Alzahrani B, Al-Barakati A. DeepDCA: novel network-based detection of IoT attacks using artificial immune system. Appl Sci Jan. 2020;10(6):6. https://doi.org/10.3390/app10061909. Art.
[18] Soe YNYN, Santosa PIPI, Hartanto R. DDoS Attack detection based on simple ANN with SMOTE for IoT environment. In: 2019 Fourth International Conference on Informatics and Computing (ICIC); Oct. 2019. p. 1–5. https://doi.org/10.1109/ICIC47613.2019.8985853.
[19] Derhab A, Aldweesh A, Emam AZAZ, Khan FAFA. Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering. Wirel Commun Mobile Comput Dec. 2020;2020:e6689134. https://doi.org/10.1155/2020/6689134.
[20] Breiman L. Random forests. Mach Learn Oct. 2001;45(1):5–32. https://doi.org/10.1023/A:1010933404324.
[21] 'MQTT - the standard for IoT messaging'. https://mqtt.org/(accessed Sep. 22, 2021).
[22] 'sFlow.org - making the network visible'. https://sflow.org/(accessed May 29, 2021).
[23] 'Node-RED'. https://nodered.org/(accessed May 24, 2021).
[24] Mucherino A, Papajorgji PJPJ, Pardalos PMPM, Mucherino A, Papajorgji PJ, Pardalos PM. k-nearest neighbor classification. Data mining in agriculture. New York, NY: Springer; 2009. p. 83–106. https://doi.org/10.1007/978-0-387-88615-2_4. Eds.
[25] 'Multilayer perceptron - an overview | sciencedirect topics'. https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron (accessed Jun. 01, 2021).
[26] 'Long short-term memory | neural computation | MIT Press'. https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory (accessed Sep. 22, 2021).
[27] Jemal I, Haddar MAMA, Cheikhrouhou O, Mahfoudhi A. M-CNN: a new hybrid deep learning model for web security. In: 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA); 2020. p. 1–7.
[28] Jemal I, Haddar MAMA, Cheikhrouhou O, Mahfoudhi A. Malicious Http request detection using code-level convolutional neural network. In: Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, November 4–6, 2020, Revised. 15; 2021. p. 317–24. *Selected Papers*.

Alaeddine Mihoub received his PhD degree in Computer Science from Grenoble-Alpes University (France, 2015). Before, he received his MSc degree in Computer Science from Paris Descartes University (France, 2012) and his engineering diploma in Statistics and Data Analysis from ESSAIT (Tunisia, 2011). Actually, Dr. Alaeddine Mihoub is an Assistant Professor of Computer Science in the College of Business and Economics at the Qassim University (Saudi Arabia). Previously, he was a Research and Data Scientist at Maskott (France) and earlier, a Postdoctoral Researcher at Orange Labs (Meylan, France). He is the co-author of many innovative patents as well as several publications in well-ranked international journals and conferences. His-research interests include Social Signal Processing, Statistical and Probabilistic Models for Interactive Systems, and more largely Artificial Intelligence, Machine Learning, and Data Science

Ouissem Ben Fredj received the BE degree in computer science from the University Manar II, Tunisia in 2002. He obtained the MS in computer science from University of Henri Poincare, France in 2003. He Obtained the PhD degree in computer science from University of Val-d'Essonnes, France in 2007. He is currently an assistant professor of computer science at University of Kairouan, Tunisia. His-research interests include vulnerability assessment, network security, and forensics

Omar Cheikhrouhou is currently an Assistant Professor at Higher Institute of Computer Science of Mahdia, University of Monastir, Tunisia. He is also a researcher at CES Lab (Computer and Embedded System), University of Sfax, National School of Engineers, Tunisia, Dr. Omar Cheikhrouhou has received his Ph.D. degrees in Computer Science from the National School of Engineers of Sfax in March 2012. His-Ph.D. deals with security of Wireless Sensor Networks and more precisely in "Secure Group Communication in Wireless Sensor Networks". Currently, his research interests span over several areas related to Wireless Sensor Networks, CyberSecurity, Edge Computing, Blockchain, Multi-Robot System Coordination, Smart and Secure Healthcare, etc.Dr. Omar has several publications in high-quality international journals and conferences. He has received some awards, including the "Governor Prize" from the Governor of Sfax in 2005

Abdelouahid Derhab received the Engineer's, M.*Sc*., and Ph.D. degrees in computer science from the University of Sciences and Technology Houari Boummediene (USTHB), Algiers, in 2001, 2003, and 2007, respectively. He was a Computer Science Engineer and a full-time Researcher with the CERIST Research Center, Algeria, from 2002 to 2012. He was an Assistant Professor with King Saud University, from 2012 to 2018. He is currently an Associate Professor with the Center of Excellence in Information Assurance (COEIA), King Saud University. He also served as a workshop chair, a technical committee chair, and a reviewer for many journals and international conferences. He is also a cyber-security policy analyst at Global Foundation for Cyber Studies and Research (GFCYBER). His-research interests are malware analysis, network security, intrusion detection, mobile security, the Internet of Things, smart grid, blockchain, and cyber security policies

Moez Krichen obtained his HDR (Ability to Conduct Researches) in Computer Science from the University of Sfax (Sfax, Tunisia) in 2018. He obtained his PhD in Computer Science in 2007. He is currently an Associate Professor at the University of Al-Baha (KSA) and a member of the Research Laboratory on Development and Control of Distributed Applications - REDCAD (Tunisia). His-main Research Interest is Model-Based Testing Methodologies for Real-Time & Distributed Systems. Moreover, he works on applying Formal Methods to several Modern Technologies like Smart Cities, Smart Vehicles, Healthcare, etc.