IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 9, 2020, accepted December 18, 2020, date of publication December 22, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046637

A Machine Learning Approach for Backoff Manipulation Attack Detection in Cognitive Radio

WASSIM FASSI FIHRI[1], HASSAN EL GHAZI[1], BADR ABOU EL MAJD[2], AND FAISSAL EL BOUANANI[3], (Senior Member, IEEE)

[1]STRS Laboratory, National Institute of Posts and Telecommunications, Rabat 10140, Morocco
[2]LMSA Laboratory, FSR, Mohammed V University, Rabat 10000, Morocco
[3]ENSIAS, Mohammed V University, Rabat 10000, Morocco

Corresponding author: Wassim Fassi Fihri (wassim.fassifihri@gmail.com)

ABSTRACT Cognitive radio (CR) is a promising paradigm that comes to address the scarcity of the radio spectrum by providing opportunistic utilization of the underutilized licensed channels to attain higher spectrum efficiency. The efficient use of the vacant portion of the spectrum depends directly on the medium access control (MAC) layer that coordinates fairly the access of the CR nodes to the idle spectrum. However, the MAC layer is vulnerable to several attacks driven by malicious nodes. One of those attacks is the backoff manipulation attack (BMA), where the selfish attacker deviates from the defined contention mechanism to gain inequitable access to the available channels. This unfair access presents some specific characteristics of an attacker, which can be considered as an input to the supervised machine learning algorithm for classification. In this paper, we propose a support vector machine (SVM) based model in order to distinctively identify the attacker depending on the throughput and the average packet delay to classify/predict an eventual attack. Finally, theoretical predictions and simulation results are presented to validate the proposed framework while giving useful insights into CR systems' performance, vulnerable to BMA attacks.

INDEX TERMS Cognitive radio, medium access control, backoff manipulation attack, machine learning, support machine vector.

I. INTRODUCTION

The growing demand for radio frequencies and the proliferation of telecommunication standards have brought a noticeable scarcity in the available spectrum in recent years. Nevertheless, according to the federal communications commission (FCC), some frequency bands are partially occupied in specific locations and at specific times. This gave rise to a new intelligent spectrum management technique called the cognitive radio (CR) [1], [2].

The idea of CR is to allow unlicensed user, called secondary user (SU), to utilize the licensed channel owned by a primary user (PU) when it is vacant. Therefore, the SU must sense the PU signal and release the white spaces if any PU activity is observed. Generally, the CR network (CRN) relies on three key phases, namely sensing, spectrum access, and management. The spectrum access and management are orchestrated by the CR medium access control (CR MAC) layer in order to efficiently operate the information collected from channel sensing and manage the spectrum sharing between the CR nodes while preserving access priority for PUs [3].

CR MAC protocols are designed to provide fair sharing and access opportunities to all users. Owing to its main role, this layer is subject to several specific attacks that degrade the performance and cause a substantial denial of service (DoS) in a considerable time. Those attacks provoked by malicious CR nodes lead to unfair access to the idle channel and create a selfish utilization of the available spectrum, which directly compromises the CR protocol compliance. One of the major attacks affecting the MAC layer is the backoff manipulation attack (BMA) [4], which exploits the vulnerability of the wireless CR nodes network adapter. Essentially, it manipulates the backoff mechanism by using a lower time period, called a backoff window, with respect to the remaining users', in aim to obtain a higher throughput to the detriment of other SUs. For that, an identification mechanism of this attack is crucial for CRN security's enhancement purpose.

A. RELATED WORK

The BMA has recently attracted several researchers' attention due to its classification as one of the biggest threats impacting

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

the CRN's security. The malicious node, which is an authenticated and authorized member of the CRN, behaves greedily to increase its chances of accessing the media via fast negotiation of channel reservation. This negotiation is contention-based, following a variant of the carrier sense medium access with collision avoidance protocol (CSMA/CA). The malicious CR node manipulates the contention protocol parameters by systematically selecting a small backoff window to increase its spectrum access probability. Several techniques have been proposed to address the BMA MAC protocol. In [5], a technique for detecting MAC layer attacks based on the distributed coordination function (DCF) and the monitoring of the collision rate caused by a DoS attack has been proposed. This technique is very difficult to implement as it requires a modification of the 802.11 protocol. The authors in [6], propose a distributed matching algorithm to allow PUs to negotiate the time during which SUs are either (i) allowed to access the spectrum, or (ii) cooperatively relaying data from the PU. The approach is interesting for spectrum access; however, it did not clearly address the identification of BMAs. The authors in [7], use a pragmatic distributed algorithm (PDA) to efficiently enhance spectrum access by reaching equilibrium via a repeated game punishment-based concept. Moreover, [8] introduces a physical layer security (PLS) to enhance the confidentiality of the SU based PDA (SU-PDA) by invoking a self-concatenated convolutional code (SECCC) schemes that have unique puncturing patterns based on advanced encryption standard (AES) key. Nevertheless, these two approaches do not provide a true countermeasure proposition for selfish channel access attack identification even if they make BMA more difficult to realize.

The attack patterns' analysis by considering a cross-layer based intrusion detection method for malicious attacks has been investigated in [9]. Nonetheless, this method remains limited due to the significant false positive (FP) rate and the important number of dropped packets. In [10], a game theory-based approach is considered for BMA detection in CRN by addressing the conflict between attacking nodes and defense mechanisms while modeling the throughput for both normal and malicious SUs. The authors provide a theoretical framework and an analytical solution for the BMA problem without offering a real prevention system against BMA. In the same way, the authors in [11], address the behavior of selfish SUs and its effect on the system by using a game theory approach. They assume SUs as players, the throughput as their payoff, and the size of the contention window as their move to obtain a Pareto-optimal and Nash equilibrium point of the system operation. However, the authors rely uniquely on the throughput to identify the attacker.

The authors in [12], propose a proactive protection scheme in distributed cognitive radio networks (DCRNs) to guarantee an independent SUs' throughput in each transmission slot. The prerequisite of this scheme is that the greedy node has to exchange and prolong the transmission time of each packet to allow normal SU to transmit data with themselves simultaneously in certain slots. This assumption is trivial as it presents the attacker as already known by the CRN. Furthermore, in real-life scenarios, the attacker is always hiding from detection to exploit the system vulnerability. In [13], the authors present a method based on the logistic classification to detect BMA attackers while providing a mathematical analysis for selfish backoff attack throughput. Nonetheless, this approach relies on a single parameter, i.e., throughput, which is insufficient to comprehend the malicious SU nodes' behavior in CRN.

The BMA in IEEE 802.11 has been analyzed in [14] to identify the honest behavior of non-colluding participants and reduce the throughput of either selfish or malicious nodes compared to the well-behaved ones. Furthermore, a random backoff control (RBC) mechanism tracking the BMA has been proposed in [15]. However, its implementation in real-time is still limited. In [16], real-time detection of DoS attacks based on the system's statistical control aiming to detect greedy behavior by monitoring abnormal throughput and inter-packet interval for each node has been presented. In the same aim, an effective Markov-RED-FT based method for protecting the legitimate traffic by calculating the stream trust values is suggested in [17].

Despite numerous works dealing with the defense against BMA, the impact of various network parameters that can lead to misinterpretation of nodes' behaviors (e.g., network congestion), lacks in the literature. This triggers the need for a smart approach-based machine learning, dealing with network parameters to reduce the false positive rate for BMA detection and pinpoint eventual attackers under different CRN configurations.

On the other hand, the support vector machine (SVM) is a supervised learning technique that produces mapping input-output features from a collection of marked training data [18]. Because of its powerful statistical learning theory, SVMs have proved high efficiency in numerous applications such as computers security, face recognition, bioinformatics, text mining [19], [20], particularly when used jointly with other computing methods, for instance, fuzzy systems and neural networks [21].

The mapping feature can be either regression or classification functions. This second is categorized into two types; linear and nonlinear kernel functions. The last-mentioned are often used to transform input data into a high-dimensional feature space whereby the input data becomes more separable compared to the original input space. Leveraging on that, SVMs have been widely used for nonlinear regression and model classification issues.

B. CONTRIBUTION

Capitalizing on the above, we aim in this work to provide an extensive machine learning-based framework for BMA attack detection. Two node characteristics, namely (i) throughput and (ii) average transmission delay, are used to propose a machine learning approach based on a nonlinear classifier using SVM. The latter allows us to distinguish between

"malicious" and "normal" classes' nodes and predict the misbehaving one. Explicitly, the SVM is learning first by making the collected dataset as input features vector to the SVM to perform the training stage. By doing this, accurate classification of different input nodes is possible, and therefore the attacker identification becomes feasible.

Pointedly, the main contributions of this paper can be summarized as follows:

- Throughput modeling of both "normal" and "malicious" classes are retrieved based on Bianchi's model [22], [23], and various parameters, including the backoff parameter, impacting the network fairness are highlighted,
- Average packet transmission delay modeling of "normal" and "malicious" classes while emphasizing the impact of BMA on the successful packet transmission,
- A novel machine learning-based approach and its formulation for BMA detection are proposed. Herein, both the throughput and the average transmission delay are the main input features training the SVM to make it able to classify correctly any new input and prevent any eventual attack.

C. ORGANIZATION OF THE PAPER

Motivated by this introduction, the rest of this paper can be structured as follows. Section II describes our system model. In section III, we present the SVM for BMA classification; we start by giving an overview of the SVM, and then we provide an SVM formulation approach for BMA detection. The simulation results are discussed in section IV. Finally, section V gives some conclusions.

II. SYSTEM MODEL

In this section, we first introduce the BMA misbehavior on the CR MAC layer that follows CSMA/CA as a channel coordination and access mechanism, and then the throughput alongside the average delay of packet transmission are modeled for later analysis.

A. BMA MISBEHAVIOR

We consider a CRN that consists of n_ℓ legitimate and n_m malicious SUs, denoted by $\left(\mathcal{U}_i^{(\ell)}\right)_{i\leq n_\ell}$ and $\left(\mathcal{U}_i^{(m)}\right)_{i\leq n_m}$, respectively. Let $b_{i,\alpha}^{(j,k)}$ represents the backoff time for node i corresponding to the jth retransmission of the kth packet with α accounts to the node type (i.e., ℓ or m). Initially, all $b_{i,\alpha}^{(0,k)}$ are assumed to be uniformly distributed in an interval [0,w-1] with w refers to the minimum size of the contention window (CW). This latter is doubled after each retransmission, up to a maximum value $w_m = 2^m w$, where m is the maximum number of stages allowed to retransmit the packet. Therefore

$$0 \leq b_{i,\alpha}^{(j,k)} \leq 2^{\min\left(m_{i,k}^{(\alpha)},m\right)}w - 1,$$

with $m_{i,k}^{(\alpha)}$ denotes the kth packet retransmission's number by $\mathcal{U}_i^{(\alpha)}$ given that successful transmissions have occurred before stage m.

Fig. 1 illustrates the transmission process based on backoff for three SUs. Initially, the backoff time of A, B, and C corresponding to their first packets are set $b_a^{(0,1)} = 10$, $b_b^{(0,1)} = 8$, and $b_c^{(0,1)} = 3$, respectively. Here, the index α has been omitted for ease of exposition as all nodes are considered legitimate. The nodes start decreasing their timers as long as the medium is sensed idle for a DCF inter-frame space (DIFS). Once $b_c^{(0,1)}$ reaches 0, the station C starts transmitting its frames and other nodes freeze the backoff decrementation (i.e., $b_a^{(0,1)} = 7, b_b^{(0,1)} = 5$) and then resume the process when the channel is detected as idle again for a DIFS interval. Here, either basic access (BA) or four-way handshake with a request to send/clear to send (RTS/CTS) will be employed to transmit the node data on the available channel. In the four-way handshake, the transmitter sends RTS to the receiver and waits until it successfully receives CTS before sending a data packet. An acknowledgment (ACK) is then immediately transmitted at the end of the packet, after a period of time called short interframe space (SIFS), to indicate a successful transmission.

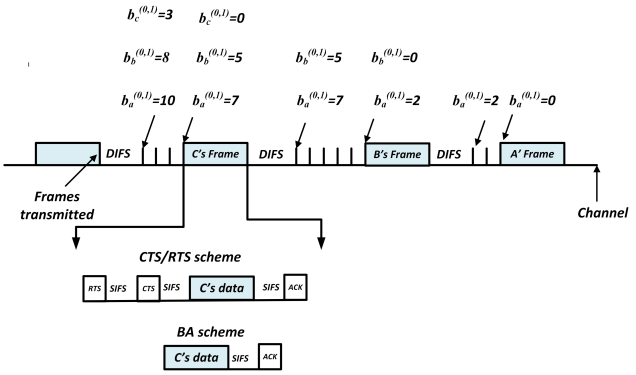


FIGURE 1. Backoff mechanism CR MAC.

It is worth mentioning that a collision occurs if two or more nodes decrease their backoff timers to 0 at the same time. In this situation, the CW is doubled for each retransmission until it reaches a maximum value. If a transmission succeeds, the node gets a random backoff time and reiterates the backoff operation for transmission of the next potential packet. In contrast, the selfish node can systematically modify its backoff time to increase its chances of reserving an idle channel compared to protocol-compliant nodes and considerably increases its throughput.

B. NODES' THROUGHPUTS UNDER BMA

In this subsection, the impact of BMA on the throughputs of both legitimate and malicious nodes is analyzed based on Bianchi's model [22]. To do so, let's first consider the following events

- $\mathcal{E}_t^{(\alpha,i)}$: Node $\mathcal{U}_i^{(\alpha)}$ is being transmitting a packet into a time slot,
- $\mathcal{E}_c^{(\alpha,i)}$: A packet transmitted by $\mathcal{U}_i^{(\alpha)}$ shall collide,

with probabilities $\tau_i^{(\alpha)} = \Pr\left(\mathcal{E}_t^{(\alpha,i)}\right)$ and $p_i^{(\alpha)} = \Pr\left(\mathcal{E}_c^{(\alpha,i)}\right)$, while $\Pr(\mathcal{E})$ refers to the probability of an event $\mathcal{E}$. One can

check that [22], [24]

$$\begin{cases} \tau_i^{(\ell)} = \dfrac{2}{1+w+p_i^{(\ell)} w \sum_{j=0}^{m-1} \left(2p_i^{(\ell)}\right)^j} \\ \tau_i^{(m)} = \dfrac{2}{1+w}. \end{cases} \tag{1}$$

Leveraging on (1), $\tau_i^{(m)}$ is denoted $\tau^{(m)}$ for the sake of simplicity with

$$p_i^{(\ell)} = 1 - \left[1-\tau^{(m)}\right]^{n_m} \prod_{\substack{j=1 \\ j\neq i}}^{n_\ell} \left(1-\tau_j^{(\ell)}\right). \tag{2}$$

Moreover, the probability that there is at least one active node transmitting a packet

$$P_{tr} = 1 - \left[1-\tau^{(m)}\right]^{n_m} \prod_{j=1}^{n_\ell} \left(1-\tau_j^{(\ell)}\right), \tag{3}$$

while the conditional probabilities $P_s^{(\alpha,i)}$ to have a successful transmission by $\mathcal{U}_i^{(\alpha)}$, given that at least one station is transmitting are given for the two considered types of SUs as

$$P_s^{(\ell,i)} = \frac{\tau_i^{(\ell)} \left[1-\tau^{(m)}\right]^{n_m} \prod_{\substack{j=1 \\ i\neq j}}^{n_\ell} \left(1-\tau_j^{(\ell)}\right)}{1-\left[1-\tau^{(m)}\right]^{n_m} \prod_{j=1}^{n_\ell} \left(1-\tau_j^{(\ell)}\right)}, \tag{4}$$

and

$$P_s^{(m,i)} = \frac{\tau^{(m)} \left[1-\tau^{(m)}\right]^{n_m-1} \prod_{j=1}^{n_\ell} \left(1-\tau_j^{(\ell)}\right)}{1-\left[1-\tau^{(m)}\right]^{n_m} \prod_{j=1}^{n_\ell} \left(1-\tau_j^{(\ell)}\right)}. \tag{5}$$

It follows that the conditional probability P_s to have a successful transmission given that at least one station is transmitting can be evaluated as

$$P_s = \sum_{i=1}^{n_\ell} P_s^{(\ell,i)} + \sum_{i=1}^{n_m} P_s^{(m,i)}. \tag{6}$$

Therefore, the probability that a station is experiencing a collision, i.e., two or more stations transmit at the same slot time, can be expressed as

$$P_c = P_{tr}\left(1-P_s\right). \tag{7}$$

Define $R_i^{(\alpha)}$ the throughput of $\mathcal{U}_i^{(\alpha)}$, as the fraction of time spent on successfully payload transmission of $\mathcal{U}_i^{(\alpha)}$

$$R_i^{(\alpha)} = \frac{E_p^{(\alpha,i)}}{E_s}, \tag{8}$$

with $E_p^{(\alpha,i)}$ refers to the average payload information transmitted of node $\mathcal{U}_i^{(\alpha)}$ in a slot time, while E_s is the average of time slot duration, given by

$$E_p^{(\alpha,i)} = P_{tr} P_s^{(\alpha,i)} L^{(\alpha,i)}, \tag{9}$$

and

$$E_s = (1-P_{tr})\,\sigma + P_{tr}\left(\sum_{i=1}^{n_\ell} P_s^{(\ell,i)} T_s^{(\ell,i)} + \sum_{i=1}^{n_m} P_s^{(m,i)} T_s^{(m,i)}\right) + P_c T_c^{(\alpha,i)}, \tag{10}$$

with $L^{(\alpha,i)}$, $T_s^{(\alpha,i)}$, and $T_c^{(\alpha,i)}$ refer respectively to the average packet payload size, the average successful transmission duration, and the average collision time for $\mathcal{U}_i^{(\alpha)}$, σ is the average duration of an empty slot time. Moreover, $T_s^{(\alpha,i)}$ and $T_c^{(\alpha,i)}$ can be evaluated as follows [22]

$$\begin{cases} T_s^{(\alpha,i)} = H + L^{(\alpha,i)} + SIFS + 2\delta + ACK + DIFS + \\ T_c^{(\alpha,i)} = H + L^{(\alpha,i)} + DIFS + \delta \\ H = H_{PHY} + H_{MAC}, \end{cases} \tag{11}$$

where H is the packet header, H_{PHY} and H_{MAC} are the packet headers for the physical and MAC layers, respectively, δ is the propagation delay, while $SIFS$, $DIFS$, and ACK represent the length of SIFS, DIFS, and ACK packets, respectively.

Specifically, for an RTS/CTS mechanism, the above equation becomes

$$\begin{cases} T_s^{(\alpha,i)} = RTS + 3SIFS + 4\delta + CTS + H \\ \qquad\qquad + L^{(\alpha,i)} + ACK + DIFS \\ T_c^{(\alpha,i)} = RTS + DIFS + \delta, \end{cases} \tag{12}$$

where RTS is the length of the RTS packet, while CTS represents the length of CTS.

Finally, the sum-rate can be expressed as

$$\begin{aligned} R &= \sum_{i=1}^{n_\ell} R_i^{(\ell)} + \sum_{i=1}^{n_m} R_i^{(m)} \\ &= P_{tr} L^{(\alpha,i)} \frac{\sum_{i=1}^{n_\ell} P_s^{(\ell,i)} + \sum_{i=1}^{n_m} P_s^{(m,i)}}{(1-P_{tr})\,\sigma + P_{tr} \sum_{\substack{i=1 \\ \alpha=\ell,m}}^{n_\alpha} P_s^{(\alpha,i)} T_s^{(\alpha,i)} + P_c T_c^{(\alpha,i)}}. \end{aligned} \tag{13}$$

C. AVERAGE PACKET DURATION UNDER BMA

In a similar manner, let's define the following events allowing to evaluate the average packet delay

- $\mathcal{E}_{S=j}^{(\alpha,i)}$: $\mathcal{U}_i^{(\alpha)}$ is transmitting a packet without collision at backoff stage j with $0 \le j \le m$,
- $\mathcal{E}_d^{(\alpha,i)}$: A packet of $\mathcal{U}_i^{(\alpha)}$ is dropped,
- $\mathcal{E}_{nd}^{(\alpha,i)}$: $\mathcal{U}_i^{(\alpha)}$ is transmitting at the jth backoff stage ($0 \le j \le m$) given that the packet is not dropped,

with probabilities given by

$$\Pr\left(\mathcal{E}_{S=j}^{(\alpha,i)}\right) = \left(1-p_i^{(\alpha)}\right)\left(p_i^{(\alpha)}\right)^j, \tag{14}$$

$$\Pr\left(\mathcal{E}_d^{(\alpha,i)}\right) = \left(p_i^{(\alpha)}\right)^{m+1}, \tag{15}$$

and

$$\begin{aligned} \Pr\left(\mathcal{E}_{nd}^{(\alpha,i)}\right) &= \frac{\Pr\left(\mathcal{E}_{S=j}^{(\alpha,i)}\right)}{1-\Pr\left(\mathcal{E}_d^{(\alpha,i)}\right)} \\ &= \frac{\left(1-p_i^{(\alpha)}\right)\left(p_i^{(\alpha)}\right)^j}{1-\left(p_i^{(\alpha)}\right)^{m+1}}. \end{aligned} \tag{16}$$

Define $D_j^{(\alpha,i)}$ as the average duration for $\mathcal{U}_i^{(\alpha)}$ to transmit a packet at j backoff stage without collision. Obviously, this

value can be seen as the summation of three ordered terms (i) duration of the backoff decrementation average, (ii) duration of j retransmissions, and (iii) the one of successful transmission, namely

$$D_j^{(\ell,i)} = E_s \sum_{r=0}^{j} \frac{w_r - 1}{2} + jT_c^{(\ell,i)} + T_s^{(\ell,i)}; \quad 1 \le i \le n_\ell, \tag{17}$$

with $(w_r - 1)/2$ is the average number of slot times that the node defers at the rth stage.

By its turn, the malicious node may face as well a collision if its backoff reaches 0 at the same time with one or more nodes. However, that node keeps the same initial CW w during all its transmission retries. Therefore, the delay for malicious node is $D_j^{(m,i)}$ denoted as follows

$$D_j^{(m,i)} = E_s \frac{(j+1)\,(w-1)}{2} + jT_c^{(m,i)} + T_s^{(m,i)}; \quad 1 \le i \le n_m. \tag{18}$$

Finally, relying on using (16), (17), and (18), the average packet duration for $\mathcal{U}_i^{(\alpha)}$ can be expressed as

$$\begin{aligned} \overline{D}_i^{(\alpha)} &= \mathbb{E}\left[D^{(\alpha,i)}\right] \\ &= \sum_{j=0}^{m} D_j^{(\alpha,i)} \Pr\left(\mathcal{E}_{nd}^{(\alpha,i)}\right). \end{aligned} \tag{19}$$

III. BMA DETECTION BASED ON SVM CLASSIFIERS

In our approach, we treat the BMA detection as a two-class pattern classification problem. We apply the SVM mechanism in CRN to identify a possible BMA. We refer to these two classes throughout as a *normal* and *malicious* node. Let $\mathrm{x}^{(\alpha)} \in (\mathbb{R} \times \mathbb{R})^{n_\alpha}$ a matrix of two rows and n_α columns denoting a pattern to be classified. Explicitly, each row of $\mathrm{x}^{(\alpha)}$ is formed of both throughput and average transmission delay

$$x_i^{(\alpha)} = \left(R_i^{(\alpha)}; \overline{D}_i^{(\alpha)}\right)^T, \quad i = 1, \ldots, n_\alpha; \tag{20}$$

where the symbol T denotes the vector's transposition.

Analogously, define the vector $\mathrm{y}^{(\alpha)} \in \mathbb{R}^{n_\alpha}$ as an identification vector such that its ith element $y_i^{(\alpha)} = 1$ refers to a *normal* node, while $y_i^{(\alpha)} = -1$ is denoting the *malicious* node. Let's define the pairs $\mathcal{P}_i^{(\alpha)} = \left(x_i^{(\alpha)}; y_i^{(\alpha)}\right)^T$.

Let's $\mathcal{T} = \left\{\mathcal{P}_i^{(\mathcal{T})}\right\}_{i \le n_\mathcal{T}}$ denotes the training database containing $n_\mathcal{T}$ datasets of pairs $\mathcal{P}_i^{(\mathcal{T})} = \left(x_i^{(\mathcal{T})}; y_i^{(\mathcal{T})}\right)^T$. For the simplicity of notations, we denote $\mathcal{P}_i = (x_i; y_i)$, $i = 1..n_\delta$ irrespective of the database's category, where $\delta = \mathcal{T}$ and $\delta = \mathcal{V}$ refer to the training and test databases, respectively.

Based on the above inputs, we need to find a well-fitting classifier, i.e., a decision function $f(x_i)$, that allows classifying an input pattern x_i independently of the database category. It is worthwhile that the linear classifiers are particular cases of nonlinear ones. In the sequel, a nonlinear classifier framework applied to BMA is introduced for clarity purposes.

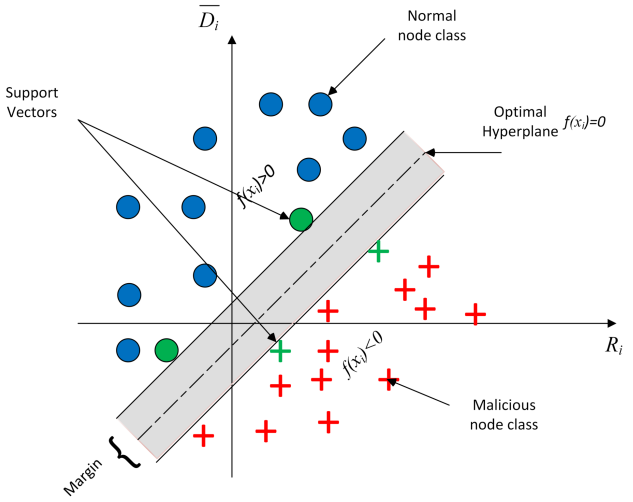


FIGURE 2. SVM classification with a hyperplane that maximizes the separating margin between the two classes (indicated by data points marked by the red "+"s and blue "O"s). Support vectors are elements of the dataset that lie on the boundary hyperplanes of the two classes (indicated by data points marked by the green "+"s and "O"s).

A. NONLINEAR SVM CLASSIFIERS

Given a database $\mathrm{x} = (x_i)_{i \le n_\delta}$, the nonlinear SVM classifier is characterized as

$$f(x_i) = \boldsymbol{\rho}^T \boldsymbol{\phi}(x_i) + b, \quad i = 1..n_\delta; \tag{21}$$

where $\boldsymbol{\rho} \in \mathbb{R}^q$ is a weight vector perpendicular to the hyperplane $\mathcal{H} = \{z \in \mathbb{R}^q, f(z) = 0\}$, b is a scalar allowing to find the hyperplane's offset from the origin, while $\boldsymbol{\phi}(.)$ is a nonlinear real-valued vector of dimension q. Particularly, the above classifier is called linear if $\boldsymbol{\phi}(x_i) = \boldsymbol{A}x_i + \boldsymbol{c}$ with $\boldsymbol{A}$ is a matrix of $\mathcal{M}_{q,2}(\mathbb{R})$ and $\boldsymbol{c} \in \mathbb{R}^q$.

Because numerous hyperplanes can separate the two aforementioned classes, as shown in Fig. 2, the suitable SVM classifier is the one maximizing the separation margin. Interestingly, it has been proven that such problem is equivalent to find an optimal solution of the following optimization problem [25], [26]

$$\begin{cases} \min \mathcal{J}(\boldsymbol{\rho}, \boldsymbol{\xi}) = \dfrac{1}{2}\|\boldsymbol{\rho}\|^2 + \mathcal{C}\sum_{i=1}^{n_\mathcal{T}} \xi_i \\ \text{s.t. } y_i\left(\boldsymbol{\rho}^T \boldsymbol{\phi}(x_i) + b\right) \ge 1 - \xi_i \\ \xi_i \ge 0, 1 \le i \le n_\mathcal{T}, \end{cases} \tag{22}$$

with $\mathcal{C}$ represents the user-defined positive regularization parameter, and ξ_i the slack variables, leading to a solution of the form

$$\boldsymbol{\rho} = \sum_{j=1}^{n_\mathcal{T}} \theta_j y_j \boldsymbol{\phi}(x_j), \tag{23}$$

where θ_j are the Lagrange multipliers, solutions of the following second optimization problem

$$\begin{cases} \max \sum_{j=1}^{n_{\mathcal{T}}} \theta_j \left(1 - \frac{y_j}{2} \sum_{i=1}^{n_{\mathcal{T}}} \theta_i y_i \mathcal{K}\left(x_j, x_i\right)\right) \\ \text{s.t. } \sum_{j=1}^{n_{\mathcal{T}}} \theta_j y_j = 0 \\ 0 \leq \theta_j \leq \mathcal{C}, 1 \leq j \leq n_{\mathcal{T}}, \end{cases} \tag{24}$$

with $\mathcal{K}(.,.)$ is a kernel function defined as

$$\mathcal{K}\left(x_j, x_i\right) = \boldsymbol{\phi}^T\left(x_j\right)\boldsymbol{\phi}\left(x_i\right). \tag{25}$$

Subsequently,

$$f(x_i) = \sum_{j=1}^{n_{\mathcal{T}}} \theta_j y_j \mathcal{K}\left(x_j, x_i\right) + b. \tag{26}$$

It is worthwhile that the interval of θ_j narrows with the decrease of its upper bound $\mathcal{C}$. That the smaller $\mathcal{C}$, the smaller θ_j and the objective function in (26) becomes steady. Towards this end, such constant should be greater enough to avoid such situation. Owing to the above, three main cases can be distinguished

- $\theta_j = 0$: In this case, $y_i f(x_i) > 1$ and x_i is correctly classified.
- $0 < \theta_j < \mathcal{C}$: Here, $y_i f(x_i) = 1$, and x_i is strictly located on the decision margin of $f(x_i)$. Under this condition, x_i is called a margin support vector of $f(x_i)$.
- $\theta_j = \mathcal{C}$: In this case, $y_i f(x_i) < 1$, and x_i is inside the decision margin. Though it may still be correctly classified. Accordingly, x_i is called in this case error support vector of $f(x_i)$.

To simplify further (26), one can ignore the null-terms corresponding to $\theta_j = 0$. To this end, let's

- $n_{\mathcal{T}}^{(1)}$ denotes the total number of support vectors, i.e. $n_{\mathcal{T}}^{(1)} = |\{i \in 1..n_{\mathcal{T}}, y_i f(x_i) \leq 1\}|$,
- $n_{\mathcal{T}}^{(1,\ell)}$ and $n_{\mathcal{T}}^{(1,m)}$ refer to the number of *normal* and *attack* support vectors,
- k_j the indices, in the set $1..n_{\mathcal{T}}^{(1)}$, of the support vectors, i.e. $\theta_{k_j} > 0$, assumed to be sorted such that $\left(k_j\right)_{j \leq n_{\mathcal{T}}^{(1,\ell)}}$ and $\left(k_j\right)_{n_{\mathcal{T}}^{(1,\ell)}+1 \leq j \leq n_{\mathcal{T}}^{(1)}}$ are associated with the two different classes. It is worthwhile that the sorting is applied only on j and that k_j are not necessarily sorted in increasing order.

Subsequently, (26) can be rewritten as

$$\begin{aligned} f(x_i) &= \sum_{j=1}^{n_{\mathcal{T}}^{(1)}} \theta_{k_j} y_{k_j} \mathcal{K}\left(x_{k_j}, x_i\right) + b \\ &\overset{(a)}{=} \sum_{j=1}^{n_{\mathcal{T}}^{(1,\ell)}} \theta_{k_j} \mathcal{K}\left(x_{k_j}, x_i\right) - \sum_{j=n_{\mathcal{T}}^{(1,\ell)}+1}^{n_{\mathcal{T}}^{(1)}} \theta_{k_j} \mathcal{K}\left(x_{k_j}, x_i\right) + b \\ &\overset{(b)}{=} \boldsymbol{\phi}^T(x_i)\boldsymbol{\psi} + b, \end{aligned} \tag{27}$$

with

$$\boldsymbol{\psi} = \sum_{j=1}^{n_{\mathcal{T}}^{(1,\ell)}} \theta_{k_j} \boldsymbol{\phi}\left(x_{k_j}\right) - \sum_{j=n_{\mathcal{T}}^{(1,\ell)}+1}^{n_{\mathcal{T}}^{(1)}} \theta_{k_j} \boldsymbol{\phi}\left(x_{k_j}\right), \tag{28}$$

is the known template used to compare the input vector feature x_i in the $\mathcal{H}$ space. Here step (a) holds with the help of (31), while identity (b) follows using (25) and taking advantage of the inner product symmetry property. Remarkably from (27), $\boldsymbol{\psi}$ is composed of the two support vectors' classes. As a consequence, a high positive and negative scores are predictable when the input vector is from *normal* and *malicious* nodes class, respectively. Moreover, the SVM decision function in (27) is provided in template-matching detector form in the nonlinear transform space $\mathcal{H}$. Interestingly, the constant b in (27) can be evaluated using any upper or lower support vector $x_i^{(sup)}$, and retrieve the following identity

$$b = f\left(x_i^{(sup)}\right) - \sum_{j=1}^{n_{\mathcal{T}}^{(1)}} \theta_{k_j} y_{k_j} \mathcal{K}\left(x_{k_j}, x_i^{(sup)}\right). \tag{29}$$

Promisingly, two well-known kernels types satisfying Mercer's condition are considered in this paper, namely, (i) polynomial, and (ii) radial basis function (RBF) kernels

$$\mathcal{K}\left(x_{k_j}, x_i\right) = \begin{cases} \left(x_{k_j}^T x_i + 1\right)^d, & \text{case (i)} \\ \exp\left(-\gamma \left\|x_{k_j} - x_i\right\|^2\right), & \text{case (ii),} \end{cases} \tag{30}$$

where d and $\gamma > 0$ are two parameters determined during the training phase.

For the two above cases, it is obvious that the kernel function is maximal if the vectors x_i and x_{k_j} are collinear or identical, respectively. Furthermore, the greater γ and the smaller d, the smaller the kernel for all pairs x_i and x_{k_j}, and approaches 0 and 1, respectively. It follows that the objective function should be close to b in the first case or independent of the kernel in the second one, leading to the worst misclassification. To this end, the values of γ and d should be chosen carefully so that to get appropriate classifications.

Finally, the retrieved decision $y_i^{(dec)}$ to compare with the predicted one y_i provided in the set $\mathcal{G}$ can be computed as follows

$$y_i^{(dec)} = \begin{cases} 1, & f(x_i) > 0 \\ -1, & f(x_i) \leq 0, \end{cases} \tag{31}$$

based on which the generalization error G_e, defined as the ratio of the incorrectly classified nodes' number to the total number of validation examples, is evaluated for both training and validation sets as follows

$$G_e^{(\mathcal{V})} = \frac{n_f^{(\mathcal{V})}}{n_{\mathcal{V}}}, \quad G_e^{(\mathcal{T})} = \frac{n_f^{(\mathcal{T})}}{n_{\mathcal{T}}}, \tag{32}$$

where $n_f^{(\mathcal{T})}$ and $n_f^{(\mathcal{V})}$ represent the total number of misclassified data in $\mathcal{T}$ and $\mathcal{V}$, respectively. Precisely, the misclasification of a *normal* node as an *attacker* (i.e., false positive) and vice-versa (i.e., false negative).

B. OPTIMIZED SVM KERNEL PARAMETERS

The optimal kernel parameters are obtained via two processes:

- The first process is the cross-validation described in **Algorithm 2**. It consists of finding the best SVM classifier parameters of a training dataset $\mathcal{T}$ reflecting the minimum generalization error for $\mathcal{T}$, i.e., $G_e^{(\mathcal{T},\min)}$. Once the optimum values for $\mathcal{T}$, namely, $\mathcal{C}^{(\mathcal{T},*)}$ and $p^{(\mathcal{T},*)}$, are obtained. They are introduced as an input to **Algorithm 1**.
- The second process is to find the global optimized SVM classifier parameters. It involves validation of the optimized classifier obtained with $\mathcal{C}^{(\mathcal{T},*)}$ and $p^{(\mathcal{T},*)}$ against the test validation dataset $\mathcal{V}$. This validation is detailed in **Algorithm 1** and consists of fine-tuning $\mathcal{C}$ and p until finding the lowest G_e for $\mathcal{V}$. Once $G_e^{(\mathcal{V},min)}$ is attained, the global optimum parameters are found, i.e., $\mathcal{C}^{(\mathcal{V},*)}$ and $p^{(\mathcal{V},*)}$.

Algorithm 1 presents the main algorithm allowing to get, for a given kernel, the optimal values of $\mathcal{C}$, and those of the two considered kernels d and γ described above. The algorithm contains essentially three phases.

1) DATASET PREPARATION PHASE

Based on a set $\mathcal{G}$ of normal and malicious nodes' transmission information, we start by computing, for each *legitimate* and *attacker* node, the two following metrics

- the throughput using (8) jointly with equations (1)-(7) and (9)-(12),
- the average packet transmission evaluated from (17) and (18) alongside equations (1)-(7), (10)-(12), and (14)-(16).

Thereby, $\mathcal{G}$ is updated accordingly by adding the two aforementioned metrics for each node. Towards this end, we start by randomly splitting the global dataset $\mathcal{G}$ into two subdatasets (i) training $\mathcal{T}$ and (ii) test validation $\mathcal{V} = \left\{\mathcal{P}_i^{(\mathcal{V})}\right\}_{i\leq n_\mathcal{V}}$ datasets, i.e., $\mathcal{G} = \mathcal{T} \cup \mathcal{V}$ with $\mathcal{G} = \left\{\mathcal{P}_i^{(\mathcal{G})}\right\}_{i\leq n_\mathcal{G}}$ and $n_\mathcal{G} = n_\mathcal{T} + n_\mathcal{V}$.

2) CROSS-VALIDATION AND TRAINING PHASE

Algorithm 2 represents the cross-validation process to get the optimum values for $\mathcal{C}$ and p for the training dataset $\mathcal{T}$. The algorithm includes three main steps.

a: RANDOM PARTITION OF DATASET $\mathcal{T}$

The first step of cross-validation process consists of preparing the dataset by randomly dividing the training dataset $\mathcal{T}$ into r equally seized subsets $\mathcal{T} = \cup_{k=1}^{r}\mathcal{S}_k$, i.e.,

$$n_{\mathcal{S}_k} = |\mathcal{S}_k| = \frac{n_\mathcal{T}}{r}, \quad k = 1..r. \tag{33}$$

The subsets $\mathcal{S}_k$ is afterward used for SVM training and validation step.

b: TRAINING AND VALIDATION FOR DATASET $\mathcal{T}$

The SVM classifier is trained r times for each model-parameter setting $(\mathcal{C}, p)$ in ranges $[\mathcal{C}_{\min}, \mathcal{C}_{\max}]$ and $[p_{\min}, p_{\max}]$ split in equidistant subintervals of length $\mathcal{C}_{\text{step}}^{(\mathcal{T})}$ and $p_{\text{step}}^{(\mathcal{T})}$, respectively. During the kth process $k \leq r$, the subsets $\mathcal{S}_k$ and its complementary $\overline{\mathcal{S}_k} = \mathcal{T} \setminus \mathcal{S}_k$ are dedicated for SVM validation and training, respectively. Note that $n_{\overline{\mathcal{S}_k}} = \left|\overline{\mathcal{S}_k}\right| = \frac{r-1}{r} n_\mathcal{T}$.

Precisely, the optimization problem (24) is solved with the help of successive minimal optimization (SMO) technique (27), based on which, $f\left(x_i^{(\overline{\mathcal{S}_k})}\right)$ are evaluated for all pairs of throughput and average delay transmission $x_i^{(\overline{\mathcal{S}_k})}$ belonging to the set $\overline{\mathcal{S}_k}$ using jointly (27) and (29). Thereby, the classifier $f(.)$ is then defined and the decisions on the $\mathcal{S}_k$ nodes classification $y_i^{(dec,\mathcal{S}_k)}$ can be calculated using (31).

c: GENERAL ERROR EVALUATION OF DATASET $\mathcal{T}$

By iterating the process r times, $G_e^{(\mathcal{S}_k)}$ is evaluated by averaging the disconcordance between $y_i^{(dec,\mathcal{S}_k)}$ and $y_i^{(\mathcal{S}_k)}$

$$\begin{aligned} G_e^{(\mathcal{S}_k)} &= \frac{\sum_{i=1}^{r}\left|y_i^{(dec,\mathcal{S}_k)} - y_i^{(\mathcal{S}_k)}\right|}{2n_{\mathcal{S}_k}} \\ &= \frac{r\sum_{i=1}^{r}\left|y_i^{(dec,\mathcal{S}_k)} - y_i^{(\mathcal{S}_k)}\right|}{2n_\mathcal{T}}. \end{aligned} \tag{34}$$

If k^* denotes the set's index associated with the minimum value of the generalized errors in training subsets, evaluated at a pair of values $\left(\mathcal{C}^{(k^*)}, p^{(k^*)}\right)$, i.e.,

$$k^* = \arg\min_{1\leq k\leq r} G_e^{(\mathcal{S}_k)}, \tag{35}$$

then

$$G_e^{(\mathcal{T},\min)} = G_e^{(\mathcal{S}_k^*)}, \mathcal{C}^{(\mathcal{T},*)} = \mathcal{C}^{(k^*)}, p^{(\mathcal{T},*)} = p^{(k^*)}. \tag{36}$$

3) VALIDATION AND OPTIMIZATION PHASE

Leveraging the three above optimum values, we first enlarge significantly the range of both $\mathcal{C}$ and p around $\mathcal{C}^{(\mathcal{T},*)}$ and $p^{(\mathcal{T},*)}$ by $2\triangle_\mathcal{C}$ and $2\triangle_p$, respectively. In a similar manner to the optimization process made in the previous phase, for each value of $\mathcal{C}$ and p in their intervals, we train this time the entire set $\mathcal{T}$, based on which the generalized error of $\mathcal{V}$, $G_e^{(\mathcal{V})}$is evaluated. Such a process is reiterated until $G_e^{(\mathcal{V})}$ falls below $G_e^{(\mathcal{T},*)}$, found in the previous step. To this end, the optimum values of $\mathcal{C}$ and p holding this condition are retrieved.

IV. RESULTS AND DISCUSSION

A. DATASET PREPARATION

In this section, we investigate the performance of the proposed algorithm, and insightful discussions are provided. Towards this end, the choice of an appropriate dataset and its preparation is a preliminary step of paramount importance. To prepare a dataset, we developed an application using

Algorithm 1 SVM Algorithm for BMA Detection

input : $\mathcal{G}, n_{\mathcal{V}}$

output : $\mathcal{C}^{(\mathcal{V},*)}, p^{(\mathcal{V},*)}, G_e^{(\mathcal{V},\min)}$ // $p^{(\mathcal{V},*)}$ denotes the optimum value of either d or γ in (30) corresponding to the minimum value $G_e^{(\mathcal{V},\min)}$.

parameter : $\tau_i^{(\ell)}, \tau^{(m)}, p_i^{(\ell)}, n_\ell, n_m, w, m, L^{(\alpha,i)}, ACK, DIFS, H_{PHY}, H_{MAC}, RTS, SIFS, \delta, n_{\mathcal{T}}, n_{\mathcal{V}}, \mathcal{K}_t, \mathcal{C}_{\min}, \mathcal{C}_{\max}, p_{\min}, p_{\max}, \mathcal{C}_{\text{step}}^{(\mathcal{T})}, p_{\text{step}}^{(\mathcal{T})}, \mathcal{C}_{\text{step}}^{(\mathcal{V})}, p_{\text{step}}^{(\mathcal{V})}, r$
// $\mathcal{K}_t$: Kernel type, can be either polynomial or RBF.
// r: Number of equisize training subsets $\mathcal{S}$.

$\left\{\boldsymbol{R}^{(\ell)}, \boldsymbol{R}^{(m)}\right\} \leftarrow$ `ComputeThroughput` $(\mathcal{G})$

$\left\{\overline{\boldsymbol{D}}^{(\ell)}, \overline{\boldsymbol{D}}^{(m)}\right\} \leftarrow$ `ComputeAverageDelay` $(\mathcal{G})$

// $\boldsymbol{R}^{(\alpha)} = \left(R_i^{(\alpha)}\right)_{i\le n_\alpha}$, $\overline{\boldsymbol{D}}^{(\alpha)} = \left(\overline{D}_i^{(\alpha)}\right)_{i\le n_\alpha}$.

$\mathcal{G} \leftarrow$ `UpdateSet` $\left(\boldsymbol{R}^{(\ell)}, \boldsymbol{R}^{(m)}, \overline{\boldsymbol{D}}^{(\ell)}, \overline{\boldsymbol{D}}^{(m)}, \mathcal{G}\right)$

$\{\mathcal{T}, \mathcal{V}\} \leftarrow$ `Extract` $(\mathcal{G}, n_{\mathcal{T}}, n_{\mathcal{V}})$ $\mathcal{K} \leftarrow$ `SetKernel` $(\mathcal{K}_t, d, \gamma)$; // Set Kernel according to eq. (30).

$\left\{\mathcal{C}^{(\mathcal{T},*)}, p^{(\mathcal{T},*)}, G_e^{(\mathcal{T},\min)}\right\} \leftarrow$ `CrossValidation` $\left(\mathcal{T}, r, \mathcal{K}, \mathcal{C}_{\min}, \mathcal{C}_{\max}, p_{\min}, p_{\max}, \mathcal{C}_{\text{step}}^{(\mathcal{T})}, p_{\text{step}}^{(\mathcal{T})}\right)$ // Call **Algorithm 2** to determine the minimum value.

$G_e^{(\mathcal{V},\min)} \leftarrow G_e^{(\mathcal{T},\min)}$ // Set an initial value for $G_e^{(\mathcal{V},\min)}$.

for $\mathcal{C} \leftarrow \mathcal{C}^{(\mathcal{T},*)} - \Delta_{\mathcal{C}}$ **to** $\mathcal{C}^{(\mathcal{T},*)} + \Delta_{\mathcal{C}}$ **by** $\mathcal{C}_{step}^{(\mathcal{V})}$ **do**
  if $\mathcal{C} \neq \mathcal{C}^{(\mathcal{T},*)}$ **then**
    for $p \leftarrow p^{(\mathcal{T},*)} - \Delta_p$ **to** $p^{(\mathcal{T},*)} + \Delta_p$ **by** $p_{step}^{(\mathcal{V})}$ **do**
      if $p \neq p^{(\mathcal{T},*)}$ **then**
        $\boldsymbol{\theta} \leftarrow$ `ApplySMO` $(\mathcal{K}, \mathcal{C}, p, \mathcal{T})$; // find $\boldsymbol{\theta}$ according to eq. (24) and (27) with $\boldsymbol{\theta} = (\theta_i)_{i\le n_{\mathcal{T}}}$.
        $f(\mathrm{x}^{(\mathcal{T})}) \leftarrow$ `Classif` $(\mathcal{K}, \mathcal{C}, p, \mathcal{T}, \boldsymbol{\theta})$ // $\mathrm{x}^{(\mathcal{T})} = \left\{x_i^{(\mathcal{T})}\right\}_{i\le n_{\mathcal{T}}^{(1)}}$, $n_{\mathcal{T}}^{(1)} = |i = 1..n_{\mathcal{T}}, \theta_i \neq 0|$.
        $\mathrm{y}^{(dec,\mathcal{V})} \leftarrow$ `ClassifTest` $\left(f(\mathrm{x}^{(\mathcal{V})})\right)$ // $\left(\mathrm{x}^{(\mathcal{V})}, \mathrm{y}^{(dec,\mathcal{V})}\right) = \left\{x_i^{(\mathcal{V})}, y_i^{(dec,\mathcal{V})}\right\}_{i\le n_{\mathcal{V}}}$.
        $G_e^{(\mathcal{V})} \leftarrow$ `EvalClass` $\left(\mathrm{y}^{(dec,\mathcal{V})}, \mathrm{y}^{(\mathcal{V})}\right)$ // $\mathrm{y}^{(\mathcal{V})} = \left\{y_i^{(\mathcal{V})}\right\}_{i\le n_{\mathcal{V}}}$.
        if $G_e^{(\mathcal{V})} \le G_e^{(\mathcal{V},\min)}$ **then**
          $G_e^{(\mathcal{V},\min)} \leftarrow G_e^{(\mathcal{V})}$ $\mathcal{C}^{(\mathcal{V},*)} \leftarrow \mathcal{C}$ $p^{(\mathcal{V},*)} \leftarrow p$
        end
      end
    end
  end
end

Algorithm 2 Cross-Validation Process

input : $\mathcal{K}, \mathcal{T}, n_{\mathcal{T}}$

output : $\mathcal{C}^{(\mathcal{T},*)}, p^{(\mathcal{T},*)}, G_e^{(\mathcal{T},\min)}$ // $p^{(\mathcal{T},*)}$ denotes the optimum value of p corresponding to the minimum value $G_e^{(\mathcal{T},\min)}$.

parameter : $r, \mathcal{K}, \mathcal{C}_{\min}, \mathcal{C}_{\max}, p_{\min}, p_{\max}, \mathcal{C}_{\text{step}}^{(\mathcal{T})}, p_{\text{step}}^{(\mathcal{T})}$
// split $\mathcal{C}$ and p intervals into subintervals of lengths $\mathcal{C}_{\text{step}}$ and p_{step}.

$\boldsymbol{\mathcal{S}} \leftarrow$ `SplitData` $(\mathcal{T}, r)$; // Randomly split the $\mathcal{T}$ to r subsets $\boldsymbol{\mathcal{S}} = (\mathcal{S}_k)_{1\le k\le r}$.

$G_e^{(\mathcal{T},\min)} \leftarrow 1$; // Set an initial value for $G_e^{(\mathcal{T},\min)}$.

for $k \leftarrow 1$ **to** r **do**
  for $\mathcal{C} \leftarrow \mathcal{C}_{\min}$ **to** $\mathcal{C}_{\max}$ **by** $\mathcal{C}_{step}^{(\mathcal{T})}$ **do**
    for $p \leftarrow p_{\min}$ **to** $p_{\max}$ **by** $p_{step}^{(\mathcal{T})}$ **do**
      $\boldsymbol{\theta} \leftarrow$ `ApplySMO` $(\mathcal{K}, \mathcal{C}, p, \overline{\mathcal{S}_k})$; // find $\boldsymbol{\theta}$ according to eq.(24) and (27) with $\boldsymbol{\theta} = (\theta_i)_{i\le n_{\overline{\mathcal{S}_k}}}$.
      $f\left(\mathrm{x}^{\left(\overline{\mathcal{S}_k}\right)}\right) \leftarrow$ `Classif` $(\mathcal{K}, \mathcal{C}, p, \overline{\mathcal{S}_k}, \boldsymbol{\theta})$ // $\mathrm{x}^{(\overline{\mathcal{S}_k})} = \left\{x_i^{(\overline{\mathcal{S}_k})}\right\}_{i\le n_{\overline{\mathcal{S}_k}}^{(1)}}$, $n_{\overline{\mathcal{S}_k}}^{(1)} = \left|i = 1..n_{\overline{\mathcal{S}_k}}, \theta_i \neq 0\right|$.
      $\mathrm{y}^{(dec,\mathcal{S}_k)} \leftarrow$ `ClassifTest` $\left(f(\mathrm{x}^{(\mathcal{S}_k)})\right)$ // $\left(\mathrm{x}^{(\mathcal{S}_k)}, \mathrm{y}^{(dec,\mathcal{S}_k)}\right) = \left\{x_i^{(\mathcal{S}_k)}, y_i^{(dec,\mathcal{S}_k)}\right\}_{i\le n_{\mathcal{S}_k}}$.
      $G_e^{(\mathcal{S}_k)} \leftarrow$ `EvalClass` $\left(\mathrm{y}^{(dec,\mathcal{S}_k)}, \mathrm{y}^{(\mathcal{S}_k)}\right)$ // $\mathrm{y}^{(\mathcal{S}_k)} = \left\{y_i^{(\mathcal{S}_k)}\right\}_{i\le n_{\mathcal{S}_k}}$.
      if $G_e^{(\mathcal{S}_k)} \le G_e^{(\mathcal{T},\min)}$ **then**
        $G_e^{(\mathcal{T},\min)} \leftarrow G_e^{(\mathcal{S}_k)}$ $\mathcal{C}^{(\mathcal{T},*)} \leftarrow \mathcal{C}$ $p^{(\mathcal{T},*)} \leftarrow p$
      end
    end
  end
end

NS3 to simulate the behavior of a BMA in the CRN network based on CSMA/CA RTS/CTS mechanism with different simulation parameters as summarized in Table 1. Particularly, the parameters H_{PHY}, H_{Mac}, ACK, RTS, and CTS correspond to the rate of 36 $Mbit/s$. To collect a consistent dataset reflecting as much as possible the reality, we start by computing both R_i and $\overline{D}_i$ for each node i, given in (8) and (19), respectively. The collected $\mathcal{G}$ size contains around 2000 datasets,

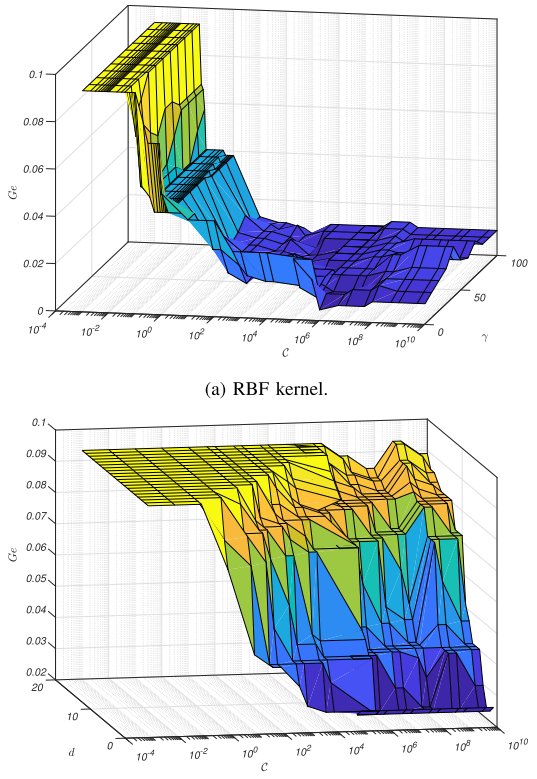
TABLE 1. Fixed parameters of dataset preparation.

Parameters	Values	Parameters	Values
H_{PHY}	192 bits	H_{MAC}	224 bits
ACK	304 bits	RTS	352 bits
CTS	304 bits	σ	20 μs
m	4	w	64
$DIFS$	50 μs	$SIFS$	10 μs
δ	1 μs	r	5
$\mathcal{C}_{min}$	10^{-3}	$\mathcal{C}_{max}$	10^{10}
γ_{min}	0	γ_{max}	100
d_{min}	0	d_{max}	20

from which, both training and validation sets are defined. Explicitly, the training set $\mathcal{T}$ contains 1253 *normal* and 273 *malicious* nodes, respectively, while validation set $\mathcal{V}$ is composed of the remaining datasets (i.e., 472 nodes).

B. SVM ASSESSMENT

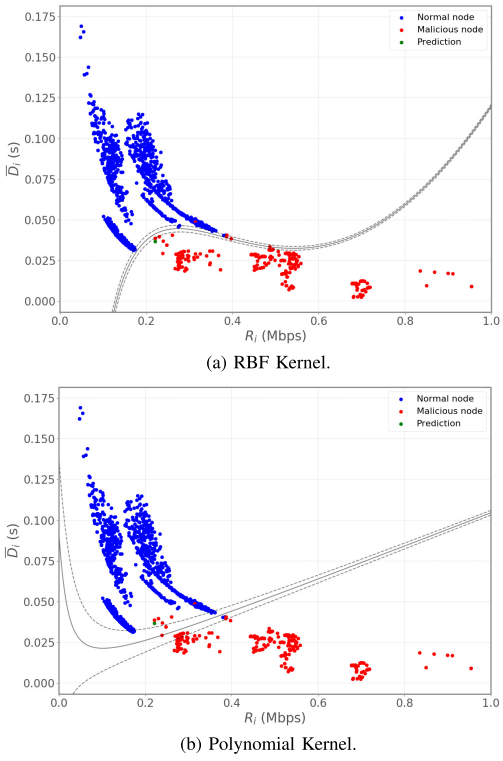
In the assessment process of SVM kernels, we adopt a generalization error G_e as a metric of efficiency. Fig. 3a, depicts G_e versus both $\mathcal{C}$ and γ, for RBF kernel. One can ascertain that the $G_e^{(\mathcal{V},min)}$, i.e., the optimal parameters' values for the RBF kernel, is located in the area delimited by $10^4 \le \mathcal{C} \le$



(a) RBF kernel.

(b) Polynomial kernel.

FIGURE 3. G_e versus $\mathcal{C}$, γ,and d for (a) RBF, and (b) polynomial kernels.



(a) RBF Kernel.

(b) Polynomial Kernel.

FIGURE 4. Hyperplane and support vectors for normal and malicious nodes classes using RBF and polynomial kernels.

10^8 and $10 \leq \gamma \leq 40$. Consequently, the test dataset $\mathcal{V}$ classification in this zone will have less misclassified input features and, therefore, a higher chance for a correct BMA detection. Moreover, one can see that G_e is higher for small values of $\mathcal{C}$ and is increasing with the increase in $\mathcal{C}$ until reaching the zone characterized by small fluctuations of G_e. Besides, the parameter γ comes to fine-tune the decision function to find the optimal $G_e^{(\mathcal{V},min)} = 0.0042$ evaluated based on the main algorithm (i.e., Algorithm 1), corresponding to the optimum values $C^{(\mathcal{V},*)} = 10^6$ and $\gamma^{(\mathcal{V},*)} = 1$. Similarly, Fig. 3b presents G_e versus both $\mathcal{C}$ and d, for the polynomial kernel. Again, the greater d and the smaller $\mathcal{C}$ are, the greater G_e is. That is, the optimum value of this latter corresponds to significant small values of d and high values of $\mathcal{C}$. Interestingly, the optimal $G_e^{(\mathcal{V},min)} = 0.0233$ is achieved over different sets of parameters ($d^{(\mathcal{V},*)} = 2$ and $10^5 \leq \mathcal{C}^{(\mathcal{V},*)} \leq 10^8$) as well as ($d^{(\mathcal{V},*)} = 3$ and $10^7 \leq \mathcal{C}^{(\mathcal{V},*)} \leq 10^{10}$). Nevertheless, The first parameters reaching the $G_e^{(\mathcal{V},min)}$ are considered the optimal values for the polynomial kernel, i.e., $d^{(\mathcal{V},*)} = 2$ and $\mathcal{C}^{(\mathcal{V},*)} = 10^5$ for its fast convergence.

Capitalizing on the above results, we can pinpoint that the SVM classifier's performance depends essentially on the kernel and its corresponding parameters. Owing to this fact, better performance is achieved using the RBF kernel providing the two optimum values, $C^{(\mathcal{V},*)} = 10^6$ and $\gamma^{(\mathcal{V},*)} = 1$.

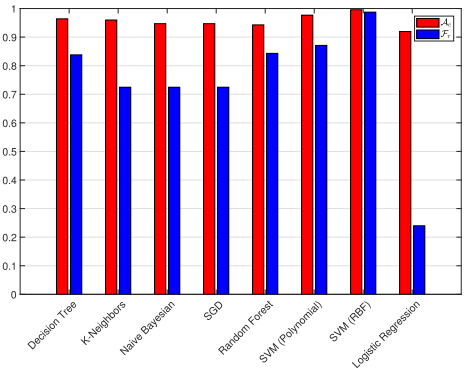
Some insights on the SVM classifier are clearly obtained by observing the hyperplane and support vectors, i.e., $0 < \theta_j < C$, produced with the optimal parameters, namely $C^{(\mathcal{V},*)}$, $\gamma^{(\mathcal{V},*)}$, and $d^{(\mathcal{V},*)}$, for each kernel. Fig. 4 shows the support vectors for both *normal* and *malicious* node classes with RBF and polynomial kernels. It can be seen from Fig. 4a that the RBF kernel gives more accurate classifications based on the optimal hyperplane, i.e., $\theta_j = 0$, of all input data with really few uncounted misclassification for both classes. Besides, as expected, the polynomial kernel in Fig. 4b gives a higher classification error in comparison to the RBF one. This can be justified by the considerable number of nodes found on the decision margin, i.e., support vectors, allowing the RBF to form a more precise optimal hyperplane shape compared to its polynomial counterpart, which gives the RBF kernel a possibility to achieve a better prediction for any new eventual attack. Furthermore, in both Fig. 4a and Fig. 4b, one can ascertain that the RBF provides a correct prediction of the attacker; its counterpart is less efficient.

To emphasize the proposed SVM-based algorithm's efficiency, we present a comparison between SVM and other well-known classifiers. Explicitly, the two metrics, namely accuracy, and the F-score, are evaluated in Fig. 5a, as follows

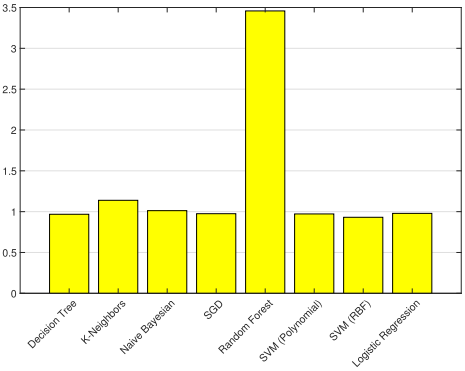
$$\mathcal{A}_c = \frac{n_c^{(\mathcal{V})}}{n_{\mathcal{V}}} = 1 - G_e \tag{37}$$

$$\mathcal{F}_s = \frac{2\mathcal{S}_e\mathcal{P}_r}{\mathcal{S}_e + \mathcal{P}_r} \tag{38}$$

where $n_c^{(\mathcal{V})}$ represents the total number of correctly classified nodes, $\mathcal{S}_e$ and $\mathcal{P}_r$ are the sensitivity, and the precision,



(a) Comparison of the Accuracy and F-score.



(b) Running time comparison (in seconds).

FIGURE 5. Comparison of the Performance and Computational complexity of various BMA detection classifiers.

respectively, given by

$$\mathcal{S}_e = \frac{n_p^{(\mathcal{V})}}{n_p^{(\mathcal{V})} + n_e^{(\mathcal{V})}} \tag{39}$$

$$\mathcal{P}_r = \frac{n_n^{(\mathcal{V})}}{n_n^{(\mathcal{V})} + n_a^{(\mathcal{V})}} \tag{40}$$

with $n_p^{(\mathcal{V})}$ and $n_e^{(\mathcal{V})}$ denote the number of "True positive" and "False positive" classifications, respectively. $n_n^{(\mathcal{V})}$ and $n_a^{(\mathcal{V})}$ the number of "True negative" classifications and "False negative" classifications, respectively.

As shown in Fig. 5a, the SVM with RBF kernel classifier gives better results than other classifiers. Additionally, the Decision Tree provides an interesting accuracy among the rest classifiers, even though it gives a smaller F-score than the Random Forest. Besides, in Fig. 5b, one can see that the SVM RBF gives the lowest running time compared to the other classifiers (i.e., 0.931s), followed by the Decision Tree, SVM polynomial, SGD, Logistic Regression, Naive Bayesian, and K-Neighbors with 0.968s, 0.972s, 0.975s, 1.012s, and 1.139s, respectively. Lastly, the Random Forest corresponds to the worst algorithm in terms of computational complexity (i.e., 3.458s). It is worth mentioning that all classifiers' methods were trained with the same dataset $\mathcal{T}$, and the results, namely accuracy, F-score, and running time are obtained against $\mathcal{V}$.

TABLE 2. Accuracy and F-score for methods using only the throughput as an attack detection parameter.

Method	$\mathcal{A}_c$	$\mathcal{F}_s$
Game theory	*0.702*	*0.764*
Bayesian theorem	*0.656*	*0.757*

Interestingly, Table 2 shows the accuracy and F-score for methods that follow one verification condition, namely the throughput, calculated based on the same dataset. As one can see, the game theory approach and the Bayesian theorem provide a low accuracy with 0.702 and 0.656, respectively, compared to the classifiers mentioned above in Fig. 5a. Moreover, although the presented F-scores for the above methods are not the lowest, they fall behind the Decision Tree, Random Forest, SVM polynomial, and SVM RBF classifiers.

V. CONCLUSION

In this paper, an SVM based approach for BMA detection in CRN was proposed. Explicitly, the analytical model of the throughput and the average transmission delay of packets for a CR network in the presence of *malicious* nodes alongside the SVM classifier model have been presented. With the help of an important dataset containing both *normal* and *malicious* nodes information, these two metrics have been evaluated, and the model has been trained and tested for several values of the kernel parameters. We demonstrate that the SVM classifier with RBF kernel produces a small generalization error compared to its polynomial counterpart when classifying samples not included in the training set. Moreover, the RBF kernel achieved the best performance with low computational complexity, compared to several well-known classifiers.

REFERENCES

[1] W. Zhang, J. Yang, G. Zhang, L. Yang, and C. Kiat Yeo, "TV white space and its applications in future wireless networks and communications: A survey," *IET Commun.*, vol. 12, no. 20, pp. 2521–2532, Dec. 2018.
[2] W. Fassi Fihri, H. El Ghazi, B. Abou El Majd, and F. El Bouanani, "A decision-making approach for detecting the primary user emulation attack in cognitive radio networks," *Int. J. Commun. Syst.*, vol. 32, no. 15, p. e4026, Oct. 2019.
[3] J. Li, H. Zhao, S. Zhang, A. S. Hafid, D. Niyato, and J. Wei, "Cross-layer analysis and optimization on access delay in channel-hopping-based distributed cognitive radio networks," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4654–4668, Jul. 2019.
[4] Y. Zhang and L. Lazos, "Vulnerabilities of cognitive radio MAC protocols and countermeasures," *IEEE Netw.*, vol. 27, no. 3, pp. 40–45, May 2013.
[5] R. Negi and A. Rajeswaran, "DoS analysis of reservation based MAC protocols," in *Proc. IEEE Int. Conf. Commun. ICC*, May 2005, pp. 3632–3636.
[6] S. Bayat, R. H. Y. Louie, Y. Li, and B. Vucetic, "Cognitive radio relay networks with multiple primary and secondary users: Distributed stable matching algorithms for spectrum access," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–6.
[7] W. Liang, S. X. Ng, J. Feng, and L. Hanzo, "Pragmatic distributed algorithm for spectral access in cooperative cognitive radio networks," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1188–1200, Apr. 2014.
[8] B. Ali, N. Zamir, S. X. Ng, and M. F. U. Butt, "Distributed matching algorithms for spectrum access: A comparative study and further enhancements," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 4, pp. 1594–1617, 2018.
[9] S. Bose and A. Kannan, "Detecting denial of service attacks using cross layer based intrusion detection system in wireless ad hoc networks," in *Proc. Int. Conf. Signal Process., Commun. Netw.*, Jan. 2008, pp. 182–188.

[10] J. Parras and S. Zazo, "Wireless networks under a backoff attack: A game theoretical perspective," *Sensors*, vol. 18, no. 2, p. 404, Jan. 2018.
[11] H. Kahsay, Y. Z. Jembre, and Y.-J. Choi, "Game-theoretic analysis of selfish secondary users in cognitive radio networks," *J. Commun. Netw.*, vol. 17, no. 4, pp. 440–448, Aug. 2015.
[12] F. Liu and H. Zhao, "Proactive protection scheme for small backoff window attack over cognitive radio networks," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1573–1576.
[13] J. Kim and K. S. Kim, "Detecting selfish backoff attack in IEEE 802.15.4 CSMA/CA using logistic classification," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 26–27.
[14] C. Alocious, H. Xiao, and B. Christianson, "Analysis of DoS attacks at MAC layer in mobile Ad Hoc networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Dubrovnik, Croatia, Aug. 2015, pp. 811–816.
[15] Y. El Hajj Shehadeh, M. Hotait, K. Tout, and D. Hogrefe, "Random backoff control to thwart malicious Behavior in WLANs," in *Proc. 19th IEEE Workshop Local Metrop. Area Netw. (LANMAN)*, Apr. 2013, pp. 1–6.
[16] A. Aaroud, M.-A. El Houssaini, A. El Hore, and J. Ben-Othman, "Real-time detection of MAC layer misbehavior in mobile ad hoc networks," *Appl. Comput. Informat.*, vol. 13, no. 1, pp. 1–9, Jan. 2017.
[17] S. K. Thambi and N. K. Sakthivel, "Real-time MAC-layer selfish misbehavior detection and prevention technique for wireless networks," *Indian J. Sci. Technol.*, vol. 8, no. 21, pp. 1–8, Oct. 2015.
[18] Y. Yang, J. Li, and Y. Yang, "The research of the fast SVM classifier method," in *Proc. 12th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2015, pp. 121–124.
[19] E. U. Haq, X. Huarong, and M. I. Khattak, "Face Recognition by SVM Using Local Binary Patterns," in *Proc. Web Inf. Syst. Appl. Conf. (WISA)*, Liuzhou, China, Nov. 2017, pp. 172–175.
[20] A. F. M. Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," in *Proc. 10th Int. Conf. Mach. Learn. Comput.*, Feb. 2018, pp. 26–30.
[21] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, Sep. 2010.
[22] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[23] G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF performance analysis," *IEEE Commun. Lett.*, vol. 9, no. 8, pp. 765–767, Aug. 2005.
[24] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 159–172, Feb. 2007.
[25] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa, "A support vector machine approach for detection of microcalcifications," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1552–1563, Dec. 2002.
[26] G. Camps-Valls, L. Gomez-Chova, J. Calpe-Maravilla, J. D. Martin-Guerrero, E. Soria-Olivas, L. Alonso-Chorda, and J. Moreno, "Robust support vector method for hyperspectral data classification and knowledge discovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1530–1542, Jul. 2004.
[27] S. S. Keerthi and S. K. Shevade, "SMO algorithm for least-squares SVM formulations," *Neural Comput.*, vol. 15, no. 2, pp. 487–507, Feb. 2003.

WASSIM FASSI FIHRI was born in Rabat, Morocco, in 1982. He received the M.S. degree in computer science and telecommunication from Ibn Tofail University, Kenitra, Morocco, in 2008. He is currently pursuing the Ph.D. degree with the Communications Systems Department, National Institute of Posts and Telecommunications (INPT), Morocco. He is also a Security Network Engineer and certified Project Manager Profession (PMP). His research interests include cybersecurity, networking, and cognitive radio networks. He has authored over five publications in well-known conferences and journals, such as WINCOM, CCWC, and the *International Journal of Communication Systems*.

HASSAN El GHAZI received the M.S. degree in wireless communications and the Ph.D. degree in electrical engineering from the Polytechnic University of Hauts-de-France, France, in 2004 and 2008, respectively. He is currently an Associate Professor with the Communications Systems Department, National Institute of Posts and Telecommunications (INPT), Morocco. He has advised many Ph.D. and graduate students at INPT and at Mohammed V University, Rabat, Morocco. So far, his research contributions have culminated in more than 50 papers in a wide variety of international journals and conferences. He served as a Reviewer for IEEE Access, Elsevier, and Springer. His main research interests include cyber-physical security, smart grid systems, and cognitive radio networks. He served as the General Chair for the IWTSC'18 conference and the Conference Chair for NISS'19 conference.

BADR ABOU EL MAJD received the master's degree in numerical analysis from Pierre and Marie Curie University (Paris 6) and the Ph.D. degree in applied mathematics from INRIA Sophia Antipolis, with a thesis titled "Hierarchical algorithms and game strategies for multidisciplinary optimization." He is currently an Associate Professor with the Mathematical Department, Mohammed V University, Rabat. He has completed the Post-Doctoral Research in "Application to the optimization of a business aircraft wing" at Prime Institute (CNRS/ENSMA/University of Poitiers), and taught at the University of Avignon, before going to École Centrale Paris, as a Scientist Researcher. He conducts researches with academic and industrial laboratories (Piaggio Aero, Dassault Aviation, and Eurodecision). He participated in several national and international projects (ACTIVOPT, OPSIM, ANR CALINS, CNRST, and IRESEN). He is the author of more than 50 publications in international journals and conferences, four book chapters, one book, and several patents and technical reports. His current research interests include multidisciplinary optimization, concurrent optimization, model reduction, and decision making. He has been a member of more than ten program committees and an editor of two books. He is the Founder and General Chair of International Conference on Transportation and Supply Chain Engineering (TSC-ENG).

FAISSAL EL BOUANANI (Senior Member, IEEE) was born in Nador, Morocco, in 1974. He received the M.S. and Ph.D. degrees in network and communication engineering from Mohammed V University-Souissi, Rabat, Morocco, in 2004 and 2009, respectively. He has served as a Faculty Member of the University of Moulay Ismail, Meknes, from 1997 to 2009, before joining the National High School, IT/ENSIAS College of Engineering, Mohammed V University, Rabat, in 2009, where he is currently an Associate Professor. He advised many Ph.D. and master's students at both Mohammed V and Moulay Ismail Universities. So far, his research efforts have culminated in more than 75 papers in a wide variety of international conferences and journals, including the IEEE Transactions on Communications, IEEE Access, the IEEE Wireless Communications Letters, the IEEE Transactions on Sustainable Computing, PIMRC, GLOBECOM, IWCMC, and CROWNCOM. His current research interests include coding, cryptography, and performance analysis of wireless communication systems. He has also been involved as a TPC member of various conferences and IEEE journals. His Ph.D. thesis was awarded the best one by Mohammed V University-Souissi, in 2010. He served as the TPC Chair for the ICSDE conferences and the General Co-Chair for ACOSIS'16 and CommNet'18 and '19 conferences. He currently serves as the General Chair for the 2019 CommNet Conference and a Reviewer for the IEEE Communications Letters and the IEEE Transactions on Cognitive Communications and Networking. He is also an Associate Editor of IEEE Access and an Editor of *Frontiers in Communications and Networks* journals.