






# When Federated Learning Meets Game Theory: A Cooperative Framework to Secure IIoT Applications on Edge Computing

Zakaria Abou El Houda , Member, IEEE, Bouziane Brik , Adlen Ksentini , Senior Member, IEEE, Lyes Khroukhi , Senior Member, IEEE, and Mohsen Guizani , Fellow, IEEE

**Abstract**—Industry 5.0 is rapidly growing as the next industrial evolution, aiming to improve production efficiency in the 21<sup>st</sup> century. This evolution relies mainly on advanced digital technologies, including Industrial Internet of Things (IIoT), by deploying multiple IIoT devices within industrial systems. Such a setup increases the possibility of threats, especially with the emergence of IIoT botnets. This can provide attackers with more sophisticated tools to conduct devastating IIoT attacks. Besides, machine learning (ML) and deep learning (DL) are considered as powerful techniques to efficiently detect IIoT attacks. However, the centralized way in building learning models and the lack of up-to-date datasets that contain the main attacks are still ongoing challenges. In this context, multiaccess edge computing (MEC) and federated learning (FL) are two promising complementary technologies. MEC brings computing capabilities at the edge of the industrial systems, while FL leverages the edge resources to enable a privacy-aware collaborative learning, especially in multiindustrial systems context. In this article, we design a novel MEC-based framework to secure IIoT applications leveraging FL, called FedGame. Specifically, FedGame enables multiple MEC domains to collaborate securely to deal with an IIoT attack, while preserving the privacy of IIoT devices. Moreover, a noncooperative game is formulated on the top of FedGame, to enable MEC nodes acquiring the needed virtual resources from the centralized MEC orchestrator, to deal with each type of IIoT attacks. We evaluate FedGame using real-world IIoT attacks; the experimental results show

not only the accuracy of FedGame against centralized ML/DL schemes while preserving the privacy of Industrial systems but also its efficiency in providing required MECs resources and, thus, dealing with IIoT attacks.

**Index Terms**—Edge computing, federated learning (FL), Industrial Internet of Things (IIoT), noncooperative game, security threats.

## I. INTRODUCTION

INDUSTRY 5.0 is rapidly growing as the next industrial evolution toward more resilient, sustainable, and human-centric industry [1]. Industry 5.0 complements and extends Industry 4.0 in order to optimize the productivity of manufacturing systems in the 21<sup>st</sup> century. This evolution combines physical operations and production with advanced digital technologies and artificial intelligence (AI) to build a better and more holistic connected ecosystem for companies that focus on supply chain management and manufacturing [1] [2]. According to the Industry 4.0 standard [2], [3], the Industrial Internet of Things (IIoT) will play a vital role in taking decentralized and autonomous decisions by monitoring and supervising manufacturing systems in real time. The IIoT refers to a set of interconnected actuators, sensors, robots, and machines, which build a complex network of services [4]. This connectivity enables data collection, transmission, and analysis. Thus, it will help to optimize the whole production process. However, such a setup may lead to escalating security threats that can target the IIoT network.

Indeed, new emerging IIoT attacks are increasing in strength and sophistication; these attacks have become destructive causing huge collateral damage and financial losses of \$10.5 trillion (USD) by 2025 [5]. In addition, the recent emergence of IIoT botnets, such as Mirai botnet, and the rapidly increasing number of insecure IIoT devices (i.e., about 75 billion IIoT devices by the end of 2025 [6]) can give attackers more powerful tools to conduct IIoT attacks. As example, on October 2, 2016, Mirai botnet conducted a huge attack using IIoT devices (i.e., closed circuit television cameras); hence, several common Internet services, including Amazon and Twitter, were unavailable for a number of hours. To alleviate these issues, intrusion detection systems (IDSs) must be properly conceived to protect the IIoT network from attacks ranging from distributed denial-of-service (DDoS) attacks to scanning attacks. In this context, machine

Manuscript received 29 December 2021; revised 18 March 2022; accepted 12 April 2022. Date of publication 26 April 2022; date of current version 9 September 2022. This work was supported by the European Union's H2020 5G!Drones project under Grant 857031. Paper no. TII-21-5842. (Zakaria Abou El Houda and Bouziane Brik contributed equally to this work.) (Corresponding author: Zakaria Abou El Houda.)

Zakaria Abou El Houda is with L@bISEN, ISEN Yncréa Ouest, 44470 Carquefou, France (e-mail: zakaria.abou.el.houda@umontreal.ca).

Bouziane Brik is with DRIVE EA1859, University Bourgogne Franche-Comté, 25000 Besançon, France (e-mail: bouziane.brik@gmail.com).

Adlen Ksentini is with the Department of Communication Systems, EURECOM, 06410 Biot, France (e-mail: Adlen.Ksentini@eurecom.fr).

Lyes Khroukhi is with the École Nationale Supérieure d'ingénieurs de Caen, Normandie University, GREYC CNRS, 14050 Caen, France (e-mail: lyas.khroukhi@ensicaen.fr).

Mohsen Guizani is with the Department of Machine Learning, Mohamed Bin Zayed University of Artificial Intelligence, Masdar, Abu Dhabi, United Arab Emirates (e-mail: mguizani@ieee.org).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TII.2022.3170347>.

Digital Object Identifier 10.1109/TII.2022.3170347

learning (ML) and deep learning (DL) are considered as powerful techniques to efficiently detect IIoT attacks. However, the centralized way in building learning models, that needs to share all the data, even privacy ones, at a central node, in addition to the lack of up-to-date data that covers all the main IIoT attacks, is still an ongoing challenge, making it difficult to train efficient ML/DL-based models.

Federated learning (FL) has emerged as a promising technique to train a global attack detection model on several edge devices, without sharing their private sensitive data [7] [8]. Hence, FL can significantly reduce the privacy risks, which makes it an ideal candidate in multi-industrial systems. Besides, multiaccess edge computing (MEC) has emerged as a novel architecture that brings cloud computing capabilities, i.e., processing and storage capacity, at the edge of networks [9]. We note that an MEC node comprises a set of applications' instances that run as virtual machines, or containers, on the top of a virtualization platform. Thus, one pertinent solution is to deploy the IDS application at the MEC nodes to secure industrial systems. However, deploying such application may not be supported by the MEC computing resources, such as storage, central processing unit (CPU), and memory, especially when considering that the 5G network is mainly based on MEC to deploy several services, such as collision avoidance, virtual and augmented reality, and data caching. Note also that MEC nodes are limited in terms of resources as compared to traditional cloud computing. Therefore, it is critical for the network operators to ensure an efficient share of MECs' resources, hence optimizing the MEC resource usage.

In this article, we design a two-stage distributed and secure collaborative architecture, called FedGame. FedGame first leverages MEC and FL to allow multiple MEC-based domains to collaboratively build an efficient learning model. The latter is able to detect IIoT attacks, while preserving the privacy of IIoT devices' data. Then, when detecting an IIoT attack, the MEC nodes compete to get more virtual resources (i.e., memory, storage, and CPU) to be able in dealing with such attacks. However, the required quantity of virtual resources depends mainly on the type of detected attack, as well as the other critical applications that are already executed on the top of each MEC node. Therefore, we model a noncooperative game between MEC collaborators to scaling up or down their virtual resources, based on both the type of detected attack and each MEC's critical applications. We evaluate FedGame using the UNSW-NB15 dataset [10], [11], which contains the main IIoT attacks, including shellcode, generic, analysis, reconnaissance, fuzzers, exploits, DDoS, backdoors, and worms. The main contributions of this article are summarized as follows.

- 1) We design a two-stage distributed collaborative architecture (FedGame) that leverages MEC and FL to allow multiple MEC-based domains to collaboratively build an efficient learning model.
- 2) We model a noncooperative game between MEC collaborators to scaling up or down their virtual resources, based on both the type of detected attack and each MEC's critical applications.
- 3) We evaluate FedGame in accuracy, detection rate (DR), and F1 score using the UNSW-NB15 dataset. The results

of the experiments show that FedGame outperforms centralized ML and DL schemes in accuracy and F1 score, while preserving the privacy of industrial systems. In addition, FedGame demonstrates the efficiency of our noncooperative game in providing required MECs resources and, thus, dealing with IIoT attacks.

The rest of this article is organized as follows. In Section II, we present a review of related works. Section III describes the design and specification of the proposed two-stage FedGame. In Section IV, we evaluate FedGame. Finally, Section V concludes this article.

## II. RELATED WORK

The rapid development of ML and DL techniques has revolutionized many domains, including security domain; since then, several schemes have adopted ML and DL techniques to improve the efficiency of their IDSs. Li *et al.* [12] designed a two-stage intrusion/anomaly detection framework based on AI to detect intrusions in software-defined Internet of Things (IoT) networks. They used the Bat scheme with two emergent techniques (i.e., swarm division and binary differential mutation) to select the most informative input features. Then, they used random forest (RF) for classification. The proposed solution achieved high accuracy in detecting illegitimate flows with lower overhead. Luo *et al.* [13] proposed a novel framework, called ensemble deep learning based web attack detection system (EDL-WADS), that uses ensemble DL techniques to detect IoT attacks, including web attacks. More especially, they designed three DL models, namely, the MRN model, the long short-term memory (LSTM) model, and the convolutional neural network (CNN) model, to detect these attacks. Then, they designed an ensemble leaning classifier for final classification/decision. Then, an ensemble classifier has been used to make the final decision. They have evaluated EDL-WADS using the CSIC 2010 dataset. Jia *et al.* [14] proposed an edge-centric IoT defense scheme, called FlowGuard, to detect IoT DDoS attacks. FlowGuard includes the detection, classification, and the mitigation of this attack; it uses a novel algorithm that is based on traffic variation metric along with two ML algorithms (i.e., LSTM and CNN) to detect malicious traffic. They have evaluated the efficiency of FlowGuard with the well-known CICDDoS2019 dataset. Ashfaq *et al.* [15] proposed the fuzzy IDS, a novel method that uses neural network (NN) along with sample categorization to detect network anomalies/attacks. Sudheera *et al.* [16] proposed a distributed framework, called Adept, to effectively detect and identify individual IoT attack. Adept is a hierarchically distributed framework that works in three phases. First, Adept processed locally IoT network traffic for detecting malicious IoT devices. Then, once an IoT attack is detected, the security manager received a potential anomaly alert to detect patterns correlated across space and time. Finally, they used machine learning schemes [i.e.,  $k$ -nearest neighbor, RF, and support vector machine (SVM)] to identify individual attack stages in the generated alert. Ravi and Shalinie [18] proposed learning driven detection mitigation (LEDEM), a novel method to detect DDoS attacks in a software-defined network. LEDEM

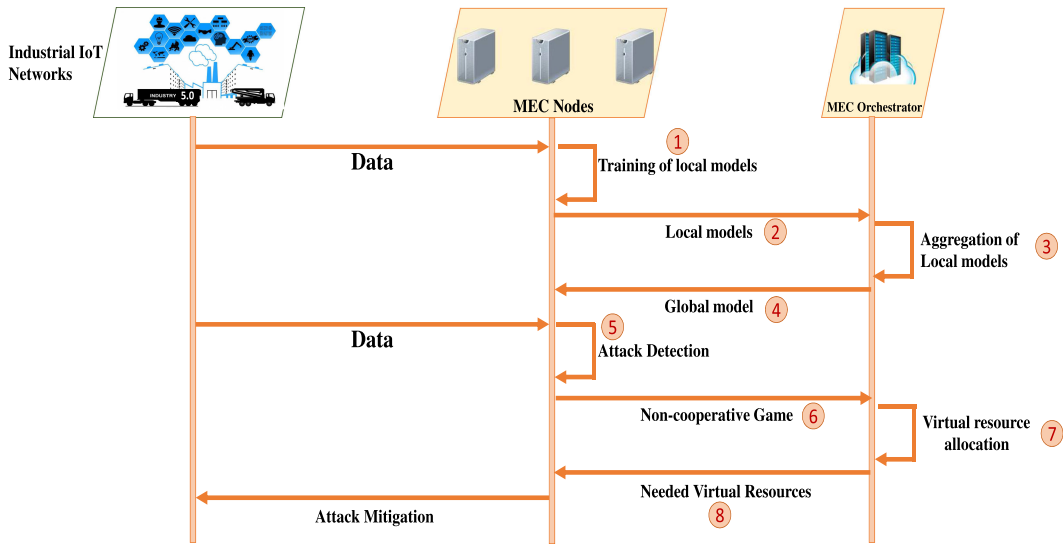


Fig. 1. Flow diagram of FedGame.

focused on mitigating DDoS attacks triggered by malicious IoT devices; it used a semisupervised machine learning algorithm, i.e., extreme learning machine, to detect DDoS attacks.

McDermott and Petrovski [19] developed a new model that makes use of the SVM and the NN to detect network anomalies in wireless sensor networks (WSNs). Moustafa *et al.* [20] developed a new model that makes use a Gaussian mixture of outliers (OGM)-based architecture to detect web attacks; it consists of 1) an association rule mining scheme to extract input features dynamically and 2) an OGM classifier to detect network attacks using the best/informative features. Moustafa *et al.* [21] designed a novel framework that uses beta mixture hidden Markov models (MHMMs) to detect network attacks/anomalies in the context of the Industry 4.0. MHMMs was evaluated on both the well-known public datasets, i.e., UNSW-NB15 and CPS datasets of sensors.

Based on our review of these existing ML- and DL-based schemes [12]–[21], they are based on some specific networks, which generally leads to inaccurate IIoT attack detection models, especially when encountering new IIoT attacks. Besides, Spinelli and Mancuso [22] proposed a survey on MEC-based schemes for resource provisioning. However, most of the cited works addressed the question: where should MEC nodes be deployed? or how to enable an efficient users' tasks offloading while ensuring a low latency delay? To the best of our knowledge, we found only one work that addresses virtual resources provisioning from the centralized orchestrator to the MEC nodes, proposed in [23]. However, this article targets a very specific application of collision detection/avoidance between vehicles. In our game formulation, we consider not only the requirement of IDS application, but also the other applications that are running at the MEC level.

### III. FEDGAME: TWO-STAGE MEC-ENABLED SCHEME

In this section, we describe the main steps of our FedGame scheme. Fig. 1 illustrates the general flow diagram of FedGame steps.

#### A. Stage 1: An FL-Based Model

In this subsection, we describe our FL-based model, ranging from the distributed architecture to the developed multiclassifier model through the used dataset.

1) **MEC-Enabled Architecture:** Fig. 2 shows the system architecture of our proposed solution. We consider four MEC domains (i.e., MECs: A, B, C, and D), where each MEC-based domain supports the requirements defined for the MEC ETSI standards; each one covers a particular geographical area, in which a set of IIoT devices is already deployed. In addition, the four MEC domains are connected to the MEC orchestrator (MEO) and the organization. MEO is in charge of deploying the MEC applications, such as IDS, collision detection/avoidance between mobile robots, and entertainment applications, on the top of a virtualized platform at each MEC server. Our architecture enables not only building learning models for IIoT-related intrusion detection, but also to deploy efficiently the IDS application at the MEC domain level, as detailed in the next subsections.

2) **Description of the IIoT Dataset:** In our study, we use the UNSW-NB15 dataset, which covers the main real-world IIoT attacks. The UNSW-NB15 dataset contains a variety of IIoT attacks, including 2 218 761 records for normal behavior, in addition to several IIoT attacks, divided as follows: analysis (2677 records), fuzzers (24 246 records), DDoS (16 353 records), backdoors (2329 records), reconnaissance (13 987 records), generic (21 5481 records), exploits (44 525 records), shellcode (1511 records), and worms (174 records).

3) **FL-Based Model:** We formalize the problem of federated collaborative learning across multiple MEC-based domains as a problem of optimization [7]. For optimization, we use a local stochastic gradient descent on each MEC-based domain. At the beginning of each round  $r$ , each MEC-based domain calculates the average gradient independently at the actual shared global model  $w_r$  using its own local dataset (see steps 1 and 2 in Fig. 1). To test FedGame, we use a deep NN with an input layer of 49 neurons that corresponds to the dimension of the UNSW-NB15 dataset, four hidden layers with LeakyReLU, and

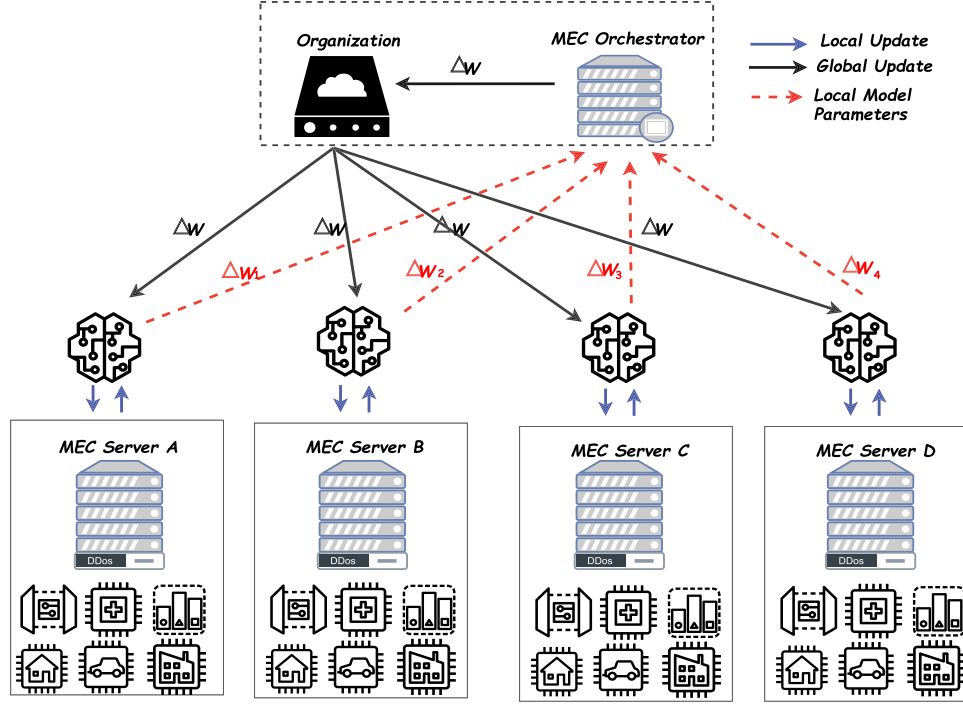


Fig. 2. MEC-based architecture for IIoT attack detection.

**Algorithm 1: MEC Collaborator  $i$ .**

**Require:** Local Data  $D_i$ , size of batches  $Siz$ , Epochs  $e$ , learning rate  $\eta$ .  
**Ensure:** Updated model  $Upd^{k+1}$ .

- 1:
- 2:  $MECUpdate(i, Upd)$
- 3: **for** epoch from 1 to  $e$  **do**
- 4:   batches  $\leftarrow D_i / Siz$
- 5:   **for** Batch  $B \in$  batches **do**
- 6:      $Upd \leftarrow Upd - \eta \nabla f(Upd, T)$  ( $\nabla f(Upd, T)$  is the average gradient on batch  $B$  at the model  $Upd$ )
- 7:   **end for**
- 8: **end for**
- 9: **return**  $Upd$  to server aggregator.

**Algorithm 2: Global MEO.**

**Require:** Number of MEC collaborators  $M$  and rounds  $Round$ , Size of batches  $Siz$ , Local epochs  $e$ , Learning rate  $\eta$ .  
**Ensure:** Aggregated model  $GLM^{k+1}$ .

- 1:
- 2: Initialize  $GLM_0$
- 3: **for**  $r = 1$  to  $Round$  **do**
- 4:    $M =$  set of MEC collaborators
- 5:   **for** MEC domain  $i \in M$  in parallel **do**
- 6:      $L_i^{r+1} \leftarrow MECUpdate(i, L^r)$
- 7:   **end for**
- 8:    $GLM^{r+1} \leftarrow \frac{1}{|M|} \sum_{i=1}^{|M|} L_i^{r+1}$
- 9: **end for**
- 10: **return**  $GLM^{r+1}$  to MEC collaborators.

an output layer of ten neurons that correspond to the category of the attack class. Algorithm 1 illustrates the steps executed by the MEC collaborator  $i$ . Finally, the MEO aggregates local updates and transmits the aggregated value to the MEC collaborators (see steps 3 and 4 in Fig. 1). This procedure is, then, repeated until a maximum round  $r_{\max}$  is achieved. Algorithm 2 describes the main steps of the global model runs at the MEO level.

**B. Stage 2: MEC Resources Provisioning**

Once detecting an IIoT attack (see step 5 in Fig. 1), the MEC nodes compete to get more virtual resources from the centralized MEO in order to be able in dealing with such attacks. However, the required quantity of virtual resources (storage, CPU, and

bandwidth) depends mainly on the type of detected attack, as well as the other critical applications that are already executed on the top of each MEC node. For instance, the MEC nodes need more virtual CPU (vCPU) resources to deal with a DDoS attack, as compared to scanning attacks e.g., user-to-root attack. Therefore, we model a noncooperative game between MEC collaborators to scaling up or down their virtual resources, based on both the type of detected attack and each MEC's critical applications (see steps 6–8 in Fig. 1). We note that we focus more on vCPU resources; however, our scheme can be easily extended/applied for other virtual resources, such as storage and bandwidth.



1) **Noncooperative Game Formulation:** We model the competitive behavior of MEC nodes to get vCPU resources using a noncooperative game,  $G = (P, S_i, \Phi_i)_{i \in P}$ , as follows.

- players**  $(p_1, \dots, p_i, \dots, p_m)$ : a set  $P$  of  $m$  MEC players that are connected to the same MEO, MEO<sub>j</sub>;
- players' strategies**,  $S_i$ : the actions that each MEC player  $p_i$  can take during the game  $\forall i \in P$ . MEC players may ask for vCPU resources between zero and  $\eta_i^{\max}$ . Thus,  $S_i = [0, \eta_i^{\max}]$  and  $S = \prod_{i=1}^m S_i = [0, \eta_1^{\max}] \times \dots \times [0, \eta_i^{\max}] \times \dots \times [0, \eta_m^{\max}]$  represent the strategy profile for all the MEC players;
- payoff function**,  $\Phi_i : S_i \rightarrow \mathbb{R}$ : each MEC player,  $p_i; \forall i \in P$ , has to maximize  $\Phi$  in order to increase its profit in getting more vCPU ( $\eta_i$ ).

Beside, we model the MECs' payoff function to include three main functions: 1) MEC nodes' objectives to maximize the obtained vCPU resources from the centralized MEO (utility); 2) priority of the detected attack (attack priority cost); and 3) critical applications that are executed at each MEC node (critical applications cost). These functions are defined as follows.

- Utility:** It reflects MEC profit when they got more vCPU resources. We note that there exist many functions, which can be used as utility functions, such as sigmoidal, logarithmic, exponential, linear, and square root [24]. We select the square root function for each MEC player  $p_i$  due to its strictly concave, as follows:

$$v_i(\eta_i) = \sqrt{\eta_i + 1}, \text{ with } i = 1, 2, \dots, m. \quad (1)$$

- Attack priority cost:** This cost reflects both the priority of each attack,  $j$ , and the number of attackers performing such attack. We assign a priority,  $Pri_j = [0, 1]$ , to each attack type based on the needed vCPU. Therefore, attacks need more vCPU resources to have more priority than those require less vCPUs to deal with. In addition, this cost is directly impacted by the number of involved attackers (*Attackers*). Indeed, an attack with multiple sources has a high impact on the network and, thus, will require more vCPU resources. We define this cost as follows:

$$\Upsilon_i(\eta_i, j) = \begin{cases} \eta_i * \left( \frac{1}{Pri_j * Attackers} \right), & \text{if there is an attack} \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

- Critical applications cost:** The quantity of assigned vCPU resources to each MEC node must consider the other MEC's applications. We classified the IIoT applications into two main classes: a) critical applications, which include safety-related application requiring ultralow latency, such as collision detection/avoidance between mobile robots in industrial systems and b) no-critical applications that cover the other type of applications, such as entertainment and publicity applications. In our model, we choose to assign more resources to MEC nodes ensuring safety critical applications (*Cri\_App*). Thus, we

define the critical applications cost of player  $p_i$  as follows:

$$\varrho_i(\eta_i, Cri\_App_i) = \eta_i * \left( 1 - \frac{Cri\_App_i}{Total\_Apps_i} \right) \quad \forall i \in P \quad (3)$$

where  $Cri\_App_i$  is the number of critical applications executed at the MEC node  $i$ , while  $Total\_Apps_i$  is the total number of MEC  $i$  applications (critical and no critical).

Finally, the payoff function of each MEC player  $p_i$  is defined as follows:

$$\Phi_i(\eta_i, \eta_{-i}) = \alpha_i v_i(\eta_i) - \beta_i \Upsilon_i(\eta_i, j) - \psi_i \varrho_i(\eta_i, Cri\_App_i). \quad (4)$$

$\eta_{-i} = [\eta_L]_{L \in P}$  and  $i \neq L$  are the requested vCPU resources by all the MEC players (strategies) except MEC player  $p_i$ , and  $\alpha_i, \beta_i$ , and  $\psi_i$  are MECs' coefficients for the three functions  $v_i, \Upsilon_i$ , and  $\varrho_i$ , respectively, where  $\alpha_i, \beta_i$ , and  $\psi_i > 0 \forall i \in P$ . The values of these parameters are chosen in such a way that the global requirements of our model are met. For instance, if the value of  $\psi_i$  is greater, the difference between vCPU resources ( $\eta_i$ ) of MEC nodes having high number of critical applications and those having low number of applications is higher and *vice versa*.

2) **Proof of Nash Equilibrium:** Nash equilibrium (NE) reflects the state where no MEC player can benefit by changing its strategy, while the other players keep theirs unchanged. Therefore, if this state exists, the modeled game admits a solution.

In our game, a set of requested vCPU resources (strategies),  $s^* \in S$  with  $s^* = [\eta_1^*, \dots, \eta_i^*, \dots, \eta_m^*]$ , corresponds to an NE state if no MEC node can improve its payoff, as it changes its action. More specifically, NE is  $N$ -tuple  $\{\eta_i^*\}$  ensuring

$$\Phi(\eta_i^*, \eta_{-i}^*) \geq \Phi(\eta_i, \eta_{-i}^*) \quad \forall i \in P, \eta_i^* \neq \eta_i. \quad (5)$$

In this subsection, we prove the uniqueness and existence of NE for our game  $G$ .

**a) NE existence:** To show the existence of NE state, we are based on the Nikaido–Isoda theorem [25].

**Theorem 1 (Nikaido–Isoda):** Our game  $G = (P, S_i, \Phi_i)_{i \in P}$  admits an NE state if and only if the set of MECs' strategies  $S_i$  is convex and compact, and MECs' payoff function  $\Phi(\eta_i, \eta_{-i})$  is concave in  $S_i$ , and continuous on all the strategies  $s \in S$ .

**Proof:**

- Since  $S_i = [0, \eta_i^{\max}] \forall i \in P$ , the set of MECs' strategy is bounded and closed. Therefore,  $S_i$  is compact. Consider  $a_1, a_2 \in S_i$  and  $\zeta = [0, 1]$ . Thus, it is clear that  $0 \leq \zeta a_1 + (1 - \zeta) a_2 \leq \eta_i^{\max}$ . As the point  $\zeta a_1 + (1 - \zeta) a_2 \in S_i$ , the strategy set,  $S_i \forall i \in P$ , is convex.
- We are based on the Hessian matrix of our payoff function,  $\Phi(s)$ , to prove its concavity property

$$H(s) = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \dots & h_{mm} \end{bmatrix} \quad (6)$$

where  $h_{kl} = (\frac{\partial^2 \Phi_k}{\partial \eta_k \partial \eta_l}) \forall k, l \in P$ . Thus, we obtain

$$h_{kl} = \begin{cases} -\frac{\alpha_k}{(2\sqrt{\eta_k}+1)^2} < 0, & \text{if } k = l \forall k, l \in P \\ 0, & \text{if } k \neq l \forall k, l \in P \end{cases} \quad (7)$$

We clearly see that  $H(s)$  is negative definite for each strategy  $s \in S$ . Thus,  $\Phi(\eta_i, \eta_{-i})$  is strictly concave in  $S_i$ , according to leading principal minor of  $H(s)$ . Based on the *Nikaido–Isoda theorem*, we can deduce that our game  $G$  has at least one NE state. ■

**b) NE uniqueness:** We consider an array of positive random values  $r = (r_1, r_2, \dots, r_m)$ . According to the theorem of Rosen [26], the weighted positive sum of  $\Phi(\eta_i, \eta_{-i}) \forall i \in P$ , is defined as follows:

$$\delta(\eta_i, \eta_{-i}; r) = \sum_{i=1}^m r_i \Phi_i(\eta_i, \eta_{-i}), \quad r_i \geq 0 \forall i \in P. \quad (8)$$

The pseudogradient of  $\delta(\eta_i, \eta_{-i}; r)$  is equal to

$$g(\eta_i, \eta_{-i}; r) = \begin{bmatrix} r_1 \nabla \Phi_1(\eta_1, \eta_{-1}) \\ r_2 \nabla \Phi_2(\eta_2, \eta_{-2}) \\ \vdots \\ r_m \nabla \Phi_m(\eta_m, \eta_{-m}) \end{bmatrix} \quad (9)$$

where  $\nabla \Phi_i(\eta_i, \eta_{-i}) = \frac{\alpha_i}{2\sqrt{\eta_i}+1} - \beta_i(\frac{1}{Pri_j * Attackers}) - \psi_i(1 - \frac{Cri\_App_i}{Total\_Apps_i})$ .

Then, we compute the Jacobian matrix  $J(\eta_i, \eta_{-i}, r)$  of  $g$  as follows:

$$J(\eta_i, \eta_{-i}, r) = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix} \quad (10)$$

with  $b_{ij} = r_i h_{ij} \forall i, j \in P$ .

The symmetric matrix  $[J + J^T]$  is negative definite for all  $(\eta_i, \eta_{-i}) \in S$ . Based on Rosen's theorem [26], we can deduce that the function  $\delta(\eta_i, \eta_{-i}; r)$  is diagonally strictly concave. Therefore, our game  $G$  admits a unique NE, based on the same theorem.

#### IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our two-stage FedGame in terms of several metrics. To test the effectiveness of our multiclass classifier (i.e., FL-based IDS), we use several metrics, including accuracy, DR, precision, and F1 score. The F1 score merges the precision and DR measures into a single measure. In addition, we study the performance of our proposed multiclass classifier using receiver operating characteristic (ROC) curves and confusion matrices. ROC curves show true positive rate (TPR) according to false positive rate (FPR). When training the global model, we try to maximize the accuracy and F1 score and to minimize the cross entropy loss function, defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^n z_i * \log(\hat{z}_i) \quad (11)$$

TABLE I  
SIMULATION PARAMETERS

Parameters	Values
Simulation Time	600 s
<b>Learning Parameters</b>	<b>Values</b>
DL tool	Pytorch
Number of hidden layers	4
Regularization technique	Dropout
Loss function	Cross-Entropy
Optimizer gradient	Adam (Adaptive Moment Estimation)
Activation function	Leaky Rectified Linear Unit
<b>Game Parameters</b>	<b>Values</b>
vCPU resources	500 vCPUs
MEC nodes	4
Critical applications	[0, 20] Apps
Generated IIoT Attacks	UNSW-NB15 dataset

TABLE II  
PERFORMANCE METRICS OF FEDGAME

Rounds	Accuracy	DR	Precision	F1	Time (s)
<b>10</b>	98%	99%	98%	98%	58
<b>25</b>	99%	99%	99%	99%	172

where  $z_i$  and  $\hat{z}_i$  represent the actual and predicted values of the  $j$ th class, respectively.

We test the global shared model on a realistic IIoT dataset UNSW-NB15. We vary the number of rounds and epochs from 10 to 25 and from 1 to 5, respectively. On the other hand, we consider an MEO that has 500 vCPU resources to share among the four MEC nodes (A, B, C, and D). Each MEC node ensures a number of critical applications that we varied between 2 and 20. Once an IIoT attack is detected at an MEC domain, our noncooperative game is established between the MEC players and the centralized MEO, till an NE state is reached. Moreover, we compared our game-based scheme with two other schemes: 1) selfish scheme, where each MEC node competes to get a maximum of vCPU resources in selfish way, i.e. without considering the performance of the centralized MEO as well as the other MEC nodes. Thus, MEO assigns a maximum number of vCPU to each MEC node and 2) minimum vCPUs scheme; the centralized MEO in this case allocates a minimum number of vCPUs to each MEC. Table I gives more details about the simulation parameters.

##### A. Evaluation of the Federated-Based Multiclassifier

Table II shows the detailed performance of FedGame. For ten rounds of training, FedGame achieves 98%, 98%, 98%, and 99% in recall, accuracy, F1 score, and precision, respectively, with only 58 s of federated training. For 25 rounds of training, FedGame achieves 99% in recall, accuracy, F1 score, and precision, with only 172 s of federated training.

Fig. 3 shows the learning curves of our MEC-based tested models over rounds; it shows the loss values during training and testing phases for ten and 25 rounds of training, respectively. We observe that during the federated training phase, the loss of each MEC-based model decreases until a minimum is reached (almost zero in the test case). Fig. 4 shows the confusion matrices of FedGame using the UNSW-NB15 dataset for ten and 25 rounds of federated training, respectively. For ten rounds of training, we

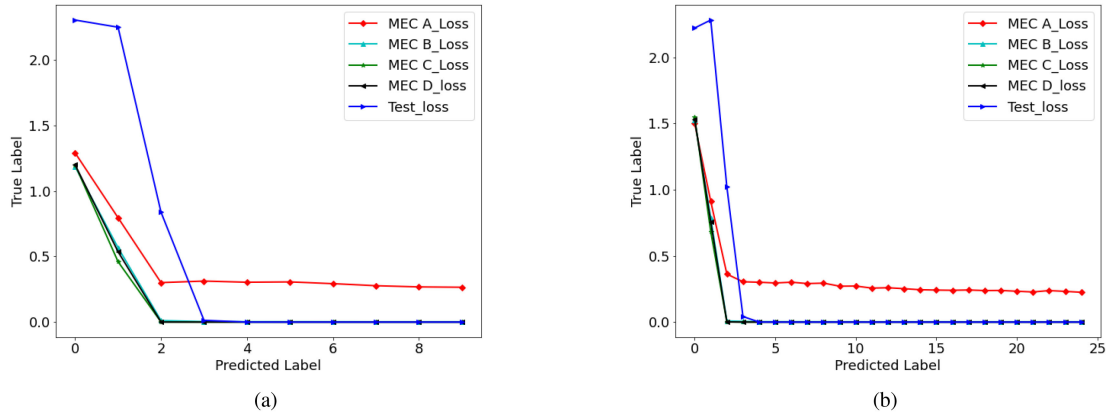


Fig. 3. Model loss for FedGame using the UNSW-NB15 dataset for (a) ten-round case and (b) 25-round case.



Fig. 4. Confusion matrices of FedGame using the UNSW-NB15 dataset for (a) ten-round case and (b) 25-round case.

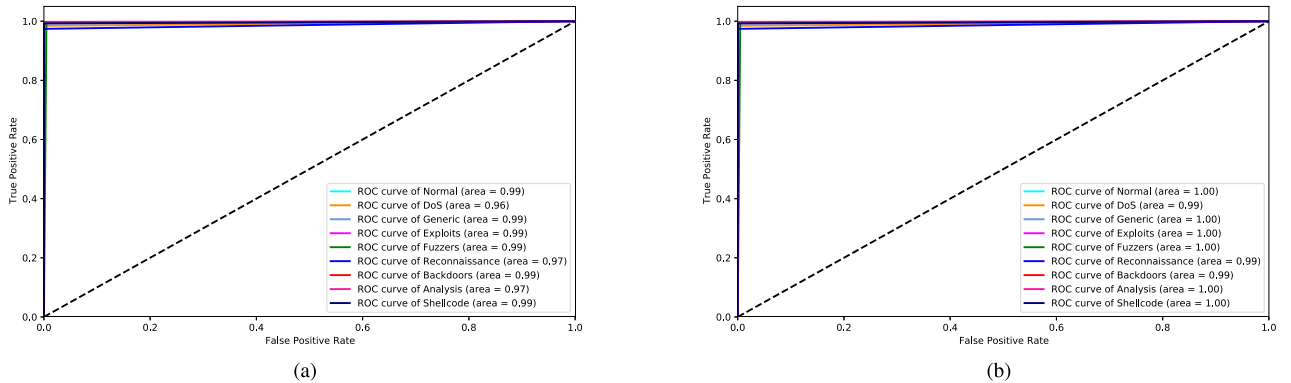


Fig. 5. ROC curves of FedGame using UNSW – NB15 dataset for: a) 10 rounds case and b) 25 rounds case.

observe that 99% of almost all the IIoT attack traffic is correctly classified as malicious traffic and also 99% of normal traffic (i.e., benign data samples) is correctly classified as benign traffic. For 25 rounds of training, we observe that 99% of almost all the IIoT attack traffic is correctly classified as malicious traffic and

also 99% of benign traffic is correctly classified as benign traffic. Fig. 5 shows the ROC curves of FedGame on the UNSW-NB15 dataset for ten and 25 rounds of federated training, respectively. The ROC curves show TPR according to FPR. For ten rounds of training, we observe that FedGame has an area under the ROC

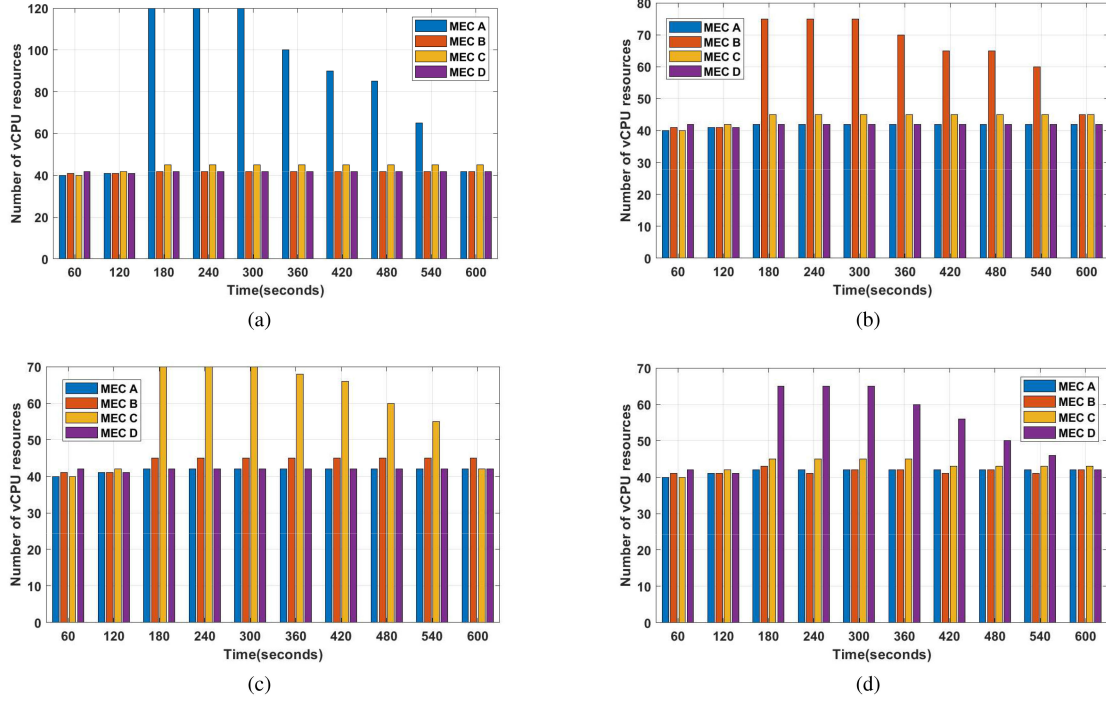


Fig. 6. Performance evaluation of FedGame, when generating IIoT's single source attacks at 180 s. (a) A DDoS attack on MEC A. (b) An analysis attack on MEC B. (c) A fuzzers attack on MEC C. (d) A backdoors attack on MEC D.

TABLE III  
COMPARISON OF PERFORMANCE METRICS

Model	Accuracy	DR	Precision	F1-score	Time (second)
Fuzzy-IDS [15]	0.86	0.85	N/A	N/A	N/A
RF-IDS [17]	0.93	0.92	N/A	N/A	N/A
WSN-IDS [19]	0.92	0.91	N/A	N/A	N/A
OGM [20]	0.95	0.94	N/A	N/A	N/A
MHMM [21]	0.96	0.95	N/A	N/A	N/A
<b>FedGame</b>	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	<b>172</b>

curve (AUC) score of 0.99, while it has an AUC score of 0.99 for 25 rounds of training.

We compared the results achieved by FedGame with the following recent ML and DL models: fuzzy IDS [15], RF-IDS [17], WSN-IDS [19], OGM [20], and MHMM [21]. Table III shows the metric values of FedGame and the centralized ML and DL models. We observe that FedGame achieves the highest accuracy of 99%, the highest DR of 99%, and the highest F1 score of 99% with only 172 s of training time. The results of experiments confirm that FedGame outperforms centralized ML and DL models in accuracy, DR, and F1 score, while preserving the privacy of industrial systems' users.

### B. Evaluation of Game-Based MEC Resource Provisioning

Fig. 6 shows the vCPUs assignment to each MEC node during 600 s. We generated different types of IIoT attacks: DDoS, analysis, fuzzers, and backdoors, addressing MEC A, B, C, and

D, respectively, at instant  $t = 180$  s. Once generating an attack, Fig. 6 shows that the number of allocated vCPUs increases at the corresponding MECs, while it remains stable in the other MEC nodes. However, the number of assigned vCPUs differs from an MEC to another (120 vCPUs for MEC A, 75 vCPUs for MEC B, 70 vCPUs for MEC C, and 65 vCPUs for MEC D). This is mainly due to the type of generated attack at each MEC. Indeed, in our scheme, the vCPU assignment depends strongly on the attacks' priority in addition to the number of attackers [see (2)]. In addition, it is clear that DDoS attack requires more vCPUs to deal with, as compared to the other attacks. Moreover, these results show clearly that our scheme enables to provide the needed vCPU resources to the compromised MEC nodes, while also ensuring a minimum and stable vCPU resources for the other MEC nodes to meet the other MEC applications' requirement.

Fig. 7 shows the performance comparison between the FedGame, Selfish, and Min vCPU schemes in terms of vCPU assignment in the MEC node B and during 600 s. Fig. 7(a) shows that the number of allocated vCPUs is almost stable over time for both the selfish and Min vCPU schemes, while it may vary for the FedGame. This is due to either the number of critical applications that may increase or decrease or IIoT attacks that may be produced at any time. To study the impact of the number of critical applications on the vCPU assignment, Fig. 7(b) shows that the FedGame increases the number of allocated vCPUs as the number of critical applications increases. However, the number of assigned vCPUs remains stable for both the selfish and Min vCPU schemes, whatever the number of critical applications. Indeed, the FedGame considers the number



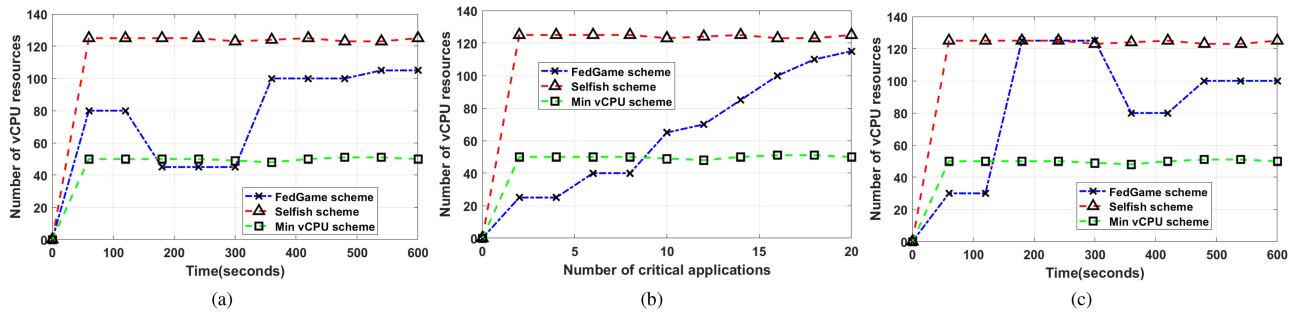


Fig. 7. (a)–(c) Performance comparison between FedGame, selfish, and min vCPU schemes.

of critical applications in allocating the vCPU resources to the MEC nodes, and the higher the number of applications, the more the vCPUs are assigned [see (3)]. Fig. 7(c) compares between the three schemes when generating a DDoS attack at  $t = 180$  s and an analysis attack at  $t = 480$  s. We clearly observe that the number of assigned vCPUs increases at  $t = 180$  s to the maximum number of 125 vCPUs, before starting to decrease till 80 vCPUs. This is due to the generated DDoS attack at  $t = 180$ . Afterward, it increases again to 100 vCPUs, at  $t = 480$  s due to the analysis attack. We note that both the selfish and Min vCPU schemes give a stable assignment behavior, given that they consider neither the IIoT attacks nor the critical applications in their vCPU assignment. Even the selfish way can provide the needed vCPUs to deal with IIoT attacks; however, most of the time, the allocated vCPUs remain unused, especially when there are no attack and critical applications, which may degrade the global performance of the system. In general, we can deduce that FedGame enables to detect collaboratively IIoT-related attacks in an efficient way, while preserving the privacy of industrial systems' users. Furthermore, once an attack is detected, FedGame ensures a dynamic and efficient virtual resource allocation to MEC domains, which considers both attack priority and MECs' critical applications, in addition to the global system performance.

## V. CONCLUSION

In this article, we designed a new two-stage scheme, called FedGame, to secure industrial systems against IIoT-based attacks. FedGame first leverages DL in a federated way to build an MEC-enabled prediction model for intrusion detection, while preserving users' privacy of industrial systems. In addition, once detecting an intrusion, a noncooperative game is established between MEC nodes to ensure provisioning the required virtual resources in order to deal with the attack. Therefore, FedGame enables not only to detect industrial systems' intrusions but also to provide the needed resources, thus dealing with any type of intrusion. Experimental results demonstrated the efficiency of FedGame, while improving users' privacy. As future work, we plan to consider different datasets, including other single- and multisource attacks, to cover most of the attacks that can target industrial systems.

## REFERENCES

- [1] P. Maddikunta *et al.*, "Industry 5.0: A survey on enabling technologies and potential applications," *J. Ind. Inf. Integr.*, vol. 26, 2021, Art. no. 100257.
- [2] U. Kannengiesser and H. Müller, "Towards viewpoint-oriented engineering for Industry 4.0: A standards-based approach," in *Proc. IEEE Ind. Cyber-Phys. Syst.*, 2018, pp. 51–56.
- [3] B. B. Brik, M. Sahnoun, and A. Louis, "Accuracy and localization-aware rescheduling for flexible flow shops in Industry 4.0," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol.*, 2019, pp. 1929–1934, doi: [10.1109/CoDIT.2019.8820445](https://doi.org/10.1109/CoDIT.2019.8820445).
- [4] A. Omar, B. Imen, S. M'hamed, B. Bouziane, and B. David, "Deployment of fog computing platform for cyber physical production system based on docker technology," in *Proc. Int. Conf. Appl. Autom. Ind. Diagnostics*, 2019, pp. 1–6, doi: [10.1109/ICAID.2019.8934949](https://doi.org/10.1109/ICAID.2019.8934949).
- [5] S. Morgan, "Cybercrime to cost the world 10.5 trillion annually by 2025." Accessed: Dec. 2021. [Online]. Available: <https://cybersecurityventures.com/>
- [6] L. Horwitz, "The future of IoT mini guide: The burgeoning IoT market continues." Accessed: Dec. 2021. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.htm>
- [7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artif. Intell. Statist.*, vol. 54, pp. 1273–1282, 2017.
- [8] B. Brik, M. Messaadia, M. Sahnoun, B. Bettayeb, and M. A. Benatia, "Fog-supported low latency monitoring of system disruptions in Industry 4.0: A federated learning approach," *ACM Trans. Cyber-Phys. Syst.*, to be published, doi: [10.1145/3477272](https://doi.org/10.1145/3477272).
- [9] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled multi-access edge computing in 5G," *IEEE Netw.*, vol. 34, no. 2, pp. 99–105, Mar./Apr. 2020, doi: [10.1109/MNET.001.1900261](https://doi.org/10.1109/MNET.001.1900261).
- [10] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.
- [11] N. Moustafa, "The future of IoT mini guide: The burgeoning IoT market continues." Accessed: Dec. 2021. [Online]. Available: [www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets](http://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets)
- [12] J. Li, Z. Zhao, R. Li, and H. Zhang, "AI-based two-stage intrusion detection for software defined IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2093–2102, Apr. 2019.
- [13] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for Internet of Things via ensemble classification," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5810–5818, Aug. 2021.
- [14] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9552–9562, Oct. 2020.
- [15] R. Ashfaq, X. Wang, J. Huang, H. Abbas, and Y. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, 2017.
- [16] K. L. K. Sudheera, D. M. Divakaran, R. P. Singh, and M. Gurusamy, "ADEPT: Detection and identification of correlated attack stages in IoT networks," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6591–6607, Apr. 2021.

- [17] K. Singh, S. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Inf. Sci.*, vol. 278, pp. 488–497, 2014.
- [18] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, Apr. 2020.
- [19] C. McDermott and A. Petrovski, "Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks," *Int. J. Comput. Netw. Commun.*, vol. 9, no. 4, pp. 45–56, 2017.
- [20] N. Moustafa, G. Misra, and J. Slay, "Generalized outlier Gaussian mixture technique based on automated association features for simulating and detecting web application attacks," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 2, pp. 245–256, Apr.–Jun. 2021.
- [21] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding Industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32910–32924, 2018.
- [22] F. Spinelli and V. Mancuso, "Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flex," *Commun. Surv. Tut.*, vol. 23, pp. 596–630, 2021.
- [23] B. Brik and A. Ksentini, "Toward optimal MEC resource dimensioning for a vehicle collision avoidance system: A deep learning approach," *IEEE Netw.*, vol. 35, no. 3, pp. 74–80, May/Jun. 2021.
- [24] L. Wang and G.-S. G. S. Kuo, "Mathematical modeling for network selection in heterogeneous wireless networks—A tutorial," *IEEE Commun. Surv. Tut.*, vol. 15, no. 1, pp. 271–292, First Quarter 2013.
- [25] H. Nikaidô and K. Isoda, "Note on non-cooperative convex games," *Pacific J. Math.*, vol. 5, pp. 807–815, 1955.
- [26] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–34, 1965.



**Zakaria Abou El Houda** (Member, IEEE) received the M.Sc. degree in computer networks from Paul Sabatier University, Toulouse, France, in 2017, the Ph.D. degree in computer science from the University of Montreal, Montreal, QC, Canada, and the Ph.D. degree in computer engineering from the University of Technology of Troyes, Troyes, France, both in 2021.

His current research interests include machine/deep-learning-based intrusion detection, federated learning, and blockchain.



**Bouziane Brik** received the Engineering degree in computer science and the Magister degree in data collection and aggregation in vehicular networks from the University of Laghouat, Laghouat, Algeria, in 2010 and 2013, respectively, and the Ph.D. degree in data collection and aggregation in vehicular networks from the University of Laghouat and La Rochelle University, La Rochelle, France, in 2017.

He is currently an Associate Professor with DRIVE Laboratory, University Bourgogne Franche-Comté, Besançon, France. Before joining University Bourgogne Franche-Comté, he was a Postdoctoral Researcher with the University of Troyes, Troyes, France; CESI School, Nanterre, France; and Eurecom School, Biot, France. He is involved in research on network slicing in the context of H2020 European projects on 5G, including MonB5G and 5GDrones. He was/is a reviewer of many ACM and IEEE transactions and IFIP, ACM, and IEEE conferences, including IEEE International Conference on Communications, IEEE Global Communications Conference, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, and IEEE Wireless Communications and Networking Conference. His research interests include the Internet of Things (IoT), IoT in industrial systems, smart grids, and vehicular networks.



**Adlen Ksentini** (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Cergy-Pontoise, Cergy, France, in 2005.

Since 2016, he has been a Professor with the Department of Communication Systems, EU-RECOM, Biot, France. His current research interests include architectural enhancements to mobile core networks, mobile cloud networking, network function virtualization, and software-defined networking.

Dr. Ksentini received the Best Paper Award at 2018 IEEE Wireless Communications and Networking Conference, 2016 IEEE International Wireless Communications and Mobile Computing Conference, 2012 IEEE International Conference on Communications, and 2005 ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, and the IEEE Fred W. Ellersik Prize for the Best *IEEE Communications Magazine* for 2017. He is an IEEE Communications Society Distinguished Lecturer on topics related to 5G and Network Softwarization.



**Lyes Khroukhi** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Sherbrooke, Sherbrooke, QC, Canada, in 2006.

From 2007 to 2008, he was a Postdoctoral Researcher with the Department of Computer Science and Operations Research, University of Montreal, Montreal, QC. He is currently a Full Professor with the École Nationale Supérieure d'ingénieurs de Caen, Normandie University, GREYC CNRS, Caen, France. His current research interests include cybersecurity, attacks detection, and performance evaluation in advanced networks, such as cloud networking, 5G/software-defined networking, Internet of Things/vehicle-to-everything, and cyber-physical systems.

His current research interests include cybersecurity, attacks detection, and performance evaluation in advanced networks, such as cloud networking, 5G/software-defined networking, Internet of Things/vehicle-to-everything, and cyber-physical systems.



**Mohsen Guizani** (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering, and the Ph.D. degree in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1987 and 1990, respectively.

He is currently the Associate Provost for Faculty Affairs and Institutional Advancement with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates. Previously, he worked at different institutions: the University of Idaho, Moscow, ID, USA; Qatar

University, Doha, Qatar; Western Michigan University, Kalamazoo, MI, USA; University of West Florida, Pensacola, FL, USA; University of Missouri–Kansas City, Kansas City, MO, USA; University of Colorado–Boulder, Boulder, CO, USA; and Syracuse University, Syracuse, NY, USA. He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019 and 2020. He has more than 800 publications in his research areas. His research interests include wireless communications and mobile computing, applied machine learning, cloud computing, and security and its application to healthcare systems.

Dr. Guizani received the 2015 IEEE Communications Society Best Survey Paper Award and four best paper awards at IEEE International Conference on Communications and IEEE Global Communications Conference.