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 Khoukhi, *Senior Member, IEEE*, Mohsen Guizani, *Fellow, IEEE*

***Abstract*— Industry 5.0 is rapidly growing as the next industrial evolution, aiming to improve production efficiency in the 21^{st} 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/Deep Learning (ML/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, Multi-access Edge Computing (MEC) and Federated Learning (FL) are two promising complementary technologies. MEC brings computing capabilities at the edge of the industrial systems, while federated learning leverages the edge resources to enable a privacy-aware collaborative learning, especially in multi-industrial systems context. In this paper, we design a novel MEC-based framework to secure IIoT applications leveraging federated learning, called FedGame. Specifically, FedGame enables multiple MEC domains to collaborate securely to deal with IIoT attack, while preserving the privacy of IIoT devices. Moreover, a non-cooperative game is formulated on 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, IIoT, Non-Cooperative Game, Security threats.**

I. Introduction

Industry 5.0 is rapidly growing as the next industrial evolution towards 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^{st} 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], 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. 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 have been 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.$, about 75 billion IIoT devices by the end of 2025 [6]), can give attackers more powerful tools to conduct IIoT attacks. As example, on the 2nd of October 2016, Mirai botnet conducted a huge attack using IIoT devices ($i.e.$, Closed Circuit Television Cameras (CCTV)), 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 and deep learning (ML/DL) are considered as powerful techniques to efficiently detect IIoT attacks. However, the centralized way in building learning models, that needs to share all 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, are still ongoing challenges, 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, Multi-access 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 a MEC node comprises a set of applications' instances that run as virtual machines, or containers, on 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 be not supported by the MEC computing resources, such as storage, CPU, and memory, especially when considering that 5G network is mainly based on MEC, to deploy several services such as collision avoidance, virtual and augmented reality, and data caching. Noting 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, and hence optimizing the MEC resource usage.

In this paper, 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 top of each

* Equal contribution

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

B. Brik is with the DRIVE EA1859, university of Bourgogne Franche-Comté, France, e-mail: (bouziane.brik@u-bourgogne.fr)

A. Ksentini is with the Communication Systems Department, EURECOM, France, e-mail: (adlen.ksentini@eurecom.fr)

L. Khoukhi is with ENSICAEN, Normandie University, GREYC CNRS, Caen, France, e-mail: (lyes.khoukhi@ensicaen.fr)

M. Guizani is with the Machine learning Department, Mohamed Bin Zayed University of Artificial Intelligence, The United Arab Emirates (e-mail: mguizani@ieee.org).

MEC node. Therefore, we model a non-cooperative 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 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 paper can be summarized as follows:

- 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.
- We model a non-cooperative 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 in accuracy, detection rate, 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. Also, FedGame demonstrates the efficiency of our non-cooperative game in providing required MECs resources, and thus dealing with IIoT attacks.

The rest of this paper 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 the paper.

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 artificial intelligence to detect intrusions in Software defined Internet of Things networks(SD-IoT). The authors used Bat scheme with two emergent techniques ($i.e.$, swarm division and binary differential mutation) to select the most informative input features. Then, the authors 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 EDL-WADS, that uses ensemble Deep Learning (DL) techniques to detect IoT attacks, including web attacks. More specially, the authors designed three DL models, namely the MRN model, the LSTM model, and the CNN model to detect these attacks. Then, the authors designed an ensemble leaning classifier for final classification/decision. Then, an ensemble classifier has been used to make the final decision. The authors have evaluated EDL-WADS using CSIC 2010 dataset. Jia *et al.* [14] proposed an edge-centric IoT defense scheme, called FlowGuard, to detect IoT Distributed Denial-of-Service (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 Fuzzy-IDS, a novel method that uses NN along with Sample Categorization (SC) 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, the authors used machine learning schemes ($i.e.$, k-Nearest Neighbor (k-NN), Random Forest (RF), and support vector machine (SVM)) to identify individual attacks stages in the generated alert. Ravi *et al.* [18] proposed LEDEM, a novel method to detect DDoS attacks in SDN. LEDEM focused on mitigating DDoS attacks triggered by malicious IoT devices; it used a semi supervised machine-learning algorithm $i.e.$, extreme learning machine (ELM) to detect DDoS attacks.

McDermott *et al.* [19] developed a new model that makes use of SVM and 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 (ARM) scheme to extract input features dynamically; and (2) An OGM classifier to detect network attacks using the best/informative features. The same authors [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 well-known public datasets $i.e.$, UNSW-NB15 and CPS dataset 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, in [22], the authors proposed a survey on MEC-based schemes for resource provisioning. However, most of cited works addressed the question of 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 work 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. Stage1: A Federated Learning-based Model

In this subsection, we describe our federated learning based model, ranging from the distributed architecture, to the developed multi-classifier model, through the used dataset.

MEC-enabled architecture: Fig. 2 shows the system architecture of our proposed solution. We consider four MEC domains ($i.e.$, MECs: A, B, C, 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 are 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 top of 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 sub-sections.

Description of IIoT Dataset: In our study, we use UNSW-NB15 dataset, which covers the main real-world IIoT attacks. UNSW-NB15 contains a variety of IIoT attacks, including 2218761 records for normal behavior, in addition to several IIoT attacks, divided as follows: analysis (2677 records), fuzzers (24246 records), DDoS (16353

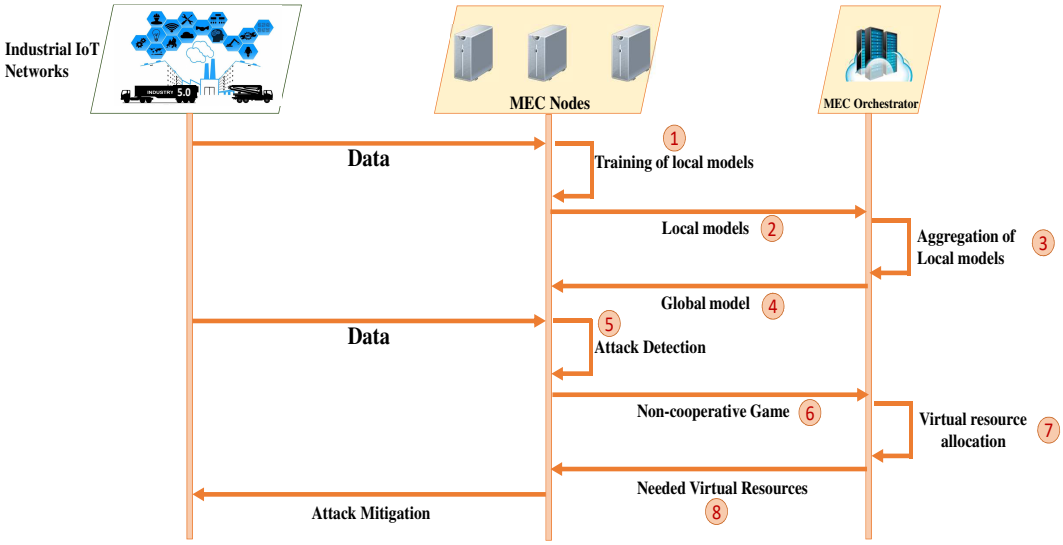


Fig. 1: Flow Diagram of FedGame.

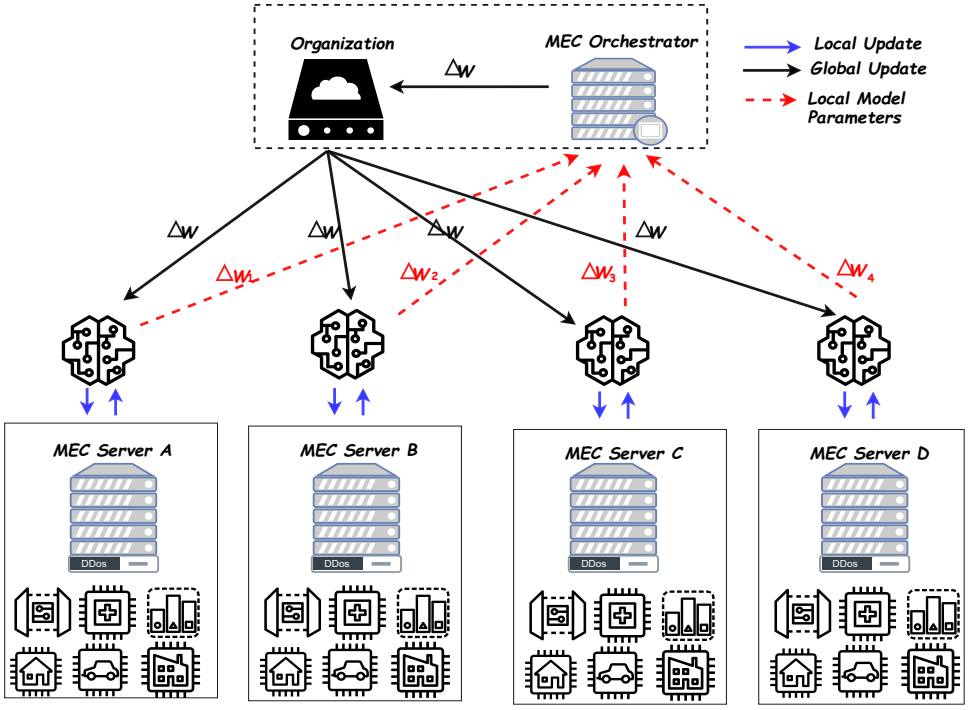


Fig. 2: MEC-Based Architecture for IIoT attack detection.

records), backdoors (2329 records), reconnaissance (13987 records), generic (215481 records), exploits (44525 records), shellcode (1511 records), and worms (174 records).

Federated learning-based Model: We formalize the problem of federated collaborative learning across multiple MEC-based domains as a problem of optimization *et al.* [7]. For optimization, we use a local Stochastic Gradient Descent (SGD) 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 neural network with an input layer of 49 neurons that corresponds to the dimension of UNSW-NB15 dataset, four hidden layers with LeakyReLU, and an output layer of ten neurons that correspond to the category of the attack class. **Algorithm 1** illustrates the steps executed by MEC collaborator i. Finally, the MEC orchestrator, i.e. 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 MEC orchestrator level.

B. Stage2: MEC Resources Provisioning

Once detecting an IIoT attack (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

Algorithm 1 MEC Collaborator i

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

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

Algorithm 2 The global MEO

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

Initialize GLM_0
for $r = 1$ **to** $Round$ **do**
  M = set of MEC collaborators
  for MEC domain $i \in M$ in parallel **do**
    $L_i^{r+1} \leftarrow MECUpdate(i, L^r)$
  end for
  $GLM^{r+1} \leftarrow \frac{1}{|M|} \sum_{i=1}^{i=|M|} L_i^{r+1}$
end for
return GLM^{r+1} to MEC collaborators.

applications that are already executed on top of each MEC node. For instance, the MEC nodes need more vCPU resources to deal with a DDoS attack, as compared to scanning attacks *e.g.*, User to Root Attack (U2R). Therefore, we model a non-cooperative 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 (steps 6, 7, and 8 in Fig. 1). We note that we focus more on virtual CPU (vCPU) resources, however, our scheme can be easily extended/applied for other virtual resources such as storage and bandwidth.

1) Non-cooperative game formulation: We model the competitive behavior of MEC nodes to get vCPU resources using a non-cooperative game, $G = (P, S_i, \Phi_i)_{i \in P}$, as follows:

1) *Players ($p_1, \ldots, p_i, \ldots, p_m$)*: a set P of m MEC players that are connected to the same MEC orchestrator, MEO_j.
2) *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 η^{max}. Thus, $S_i = [0, \eta_i^{max}]$ and $S = \prod_{i=1}^{m} S_i = [0, \eta_1^{max}] \times \ldots \times [0, \eta_i^{max}] \times \ldots \times [0, \eta_m^{max}]$ represents the strategy profile for all MEC players.
3) *Payoff function,* $\Phi_i : S_i \rightarrow \mathbb{R}$: each MEC player, $p_i; \forall i \in P$, has to maximize Φ, in order to increase its profit in getting more vCPU (η_i).

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

1) Utility: it reflects MECs 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:

$$\upsilon_i(\eta_i) = \sqrt{\eta_i + 1}, with\ i = 1, 2, \ldots, m \quad (1)$$

2) 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. So, attacks need more vCPU resources 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, attack with multiple source has a high impact on the network and thus will require more vCPU resources. We define this cost as follows:

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

3) 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: (i) critical applications, which include safety-related application requiring ultra-low latency, such as collision detection/avoidance between mobile robots in industrial systems. (ii) 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:

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

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:

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

With $\eta_{-i} = [\eta_L]_{L \in P}$ and $i \neq L$ is the requested vCPU resources by all MEC players (strategies) except MEC player p_i, and α_j, β_j, and ψ_j are MECs' coefficients for the three functions υ_i, Υ_i, and ϱ_i, respectively, where α_i, β_i, and $\psi_i > 0$, $\forall i \in P$. The values of these parameters are chosen in such way that the global requirement of our model are met. For instance, if the value of ψ_i is greater, the difference between vCPU resources (η_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 $s^* = [\eta_1^*, \ldots, \eta_i^*, \ldots, \eta_m^*]$, corresponds to a Nash equilibrium state if no MEC node can improve its payoff, as it changes its action. More specifically, NE is N-tuple $\{\eta_i^*\}$ ensuring:

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

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

Nash Equilibrium Existence:
To show the existence of NE state, we are based on Nikaido-Isoda theorem [25]:

Theorem 1 (Nikaido-Isoda): Our game $G = (P, S_i, \Phi_i)_{i \in P}$ admits a 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. So, S_i is compact.
  Considering $a_1, a_2 \in S_i$ and $= [0, 1]$. Thus, it is clear that $0 \leq a_1 + (1-) a_2 \leq \eta_i^{max}$. As the point $a_1 + (1-) a_2 \in S_i$, 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 propriety:

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

Where $h_{kl} = \left(\frac{\partial^2 \Phi_k}{\partial \eta_k \partial \eta l} \right), \forall k, l \in P$. Thus, we get:

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

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-Isodra theorem*, we can deduce that our game G has at least one Nash Equilibrium state.

■

Nash Equilibrium Uniqueness:
We consider an array of positive random values $r = (r_1, r_2, \ldots, r_m)$. 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:

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

And the pseudo-gradient of $\delta(\eta_i, \eta_{-i}; r)$ is equal to:

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

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

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

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

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

As the symmetric matrix $\left[J + J^T \right]$ 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 an unique NE, based on the same theorem.

TABLE I: Simulation Parameters.

Parameters	**Values**
Simulation Time	600 s
Learning Parameters	**Values**
Deep learning tool	Pytorch
Number of hidden layers	4
Regularization technique	Dropout
Loss function	Cross-Entropy
Optimiser gradient	Adam (Adaptive Moment Estimation)
Activation function	Leaky Rectified Linear Unit
Game Parameters	**Values**
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)**
10	98%	99%	98%	98%	58
25	99%	99%	99%	99%	172

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 Multi-class classifier (*i.e.*, federated learning-based IDS), we use several metrics, including Accuracy, detection rate (DR), Precision, and F1-score. The F1 score merges the precision and detection rate measures into a single measure. Also, we study the performance of our proposed Multi-class classifier using 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:

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

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

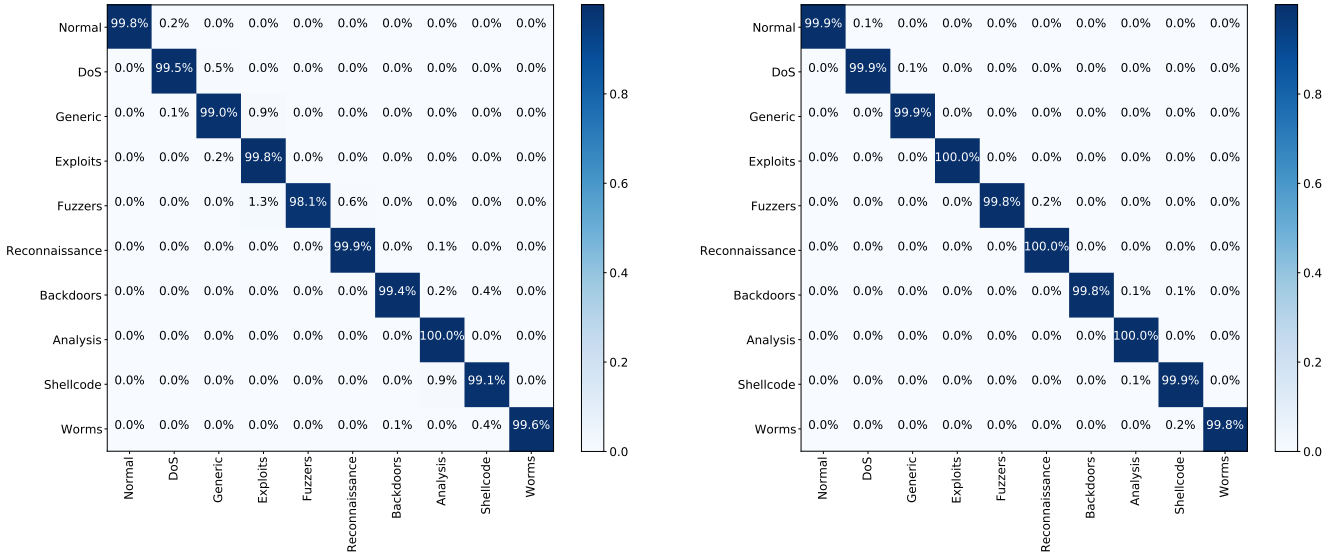
We test the global shared model on a realistic IIoT dataset UNSW-NB15. We have been varying the number of rounds and epochs from 10 to 25 and from from 1 to 5, respectively. On the other hand, we consider a MEC orchestrator (MEO) that has 500 vCPU resources to share among the four MEC nodes (A, B, C, and D). Each MEC node ensures a number of critical applications that we varied between 2 to 20. Once an IIoT attack is detected at a MEC domain, our non-cooperative game is established between the MEC players and the centralized MEO, till a Nash Equilibrium state is reached. Moreover, we compared our game-based scheme with two other schemes: (i) 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. (ii) 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 Federated-based Multi-classifier

Table II shows detailed performance of FedGame. For 10 rounds of training, FedGame achieves 98%, 98%, 98%, 99% in recall, accuracy, F1 score, and precision, respectively, with only 58 seconds of federated training. For 25 rounds of training, FedGame achieves

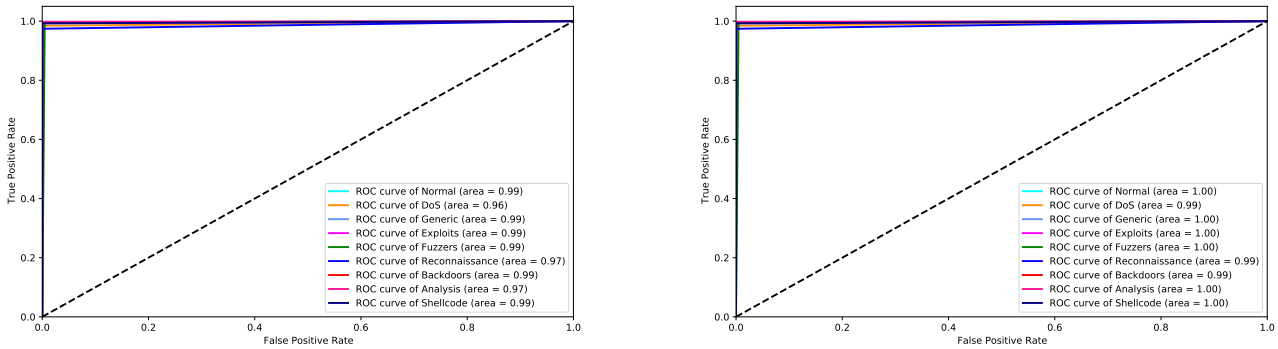
(a) (b)

Fig. 3: Model loss for FedGame using $UNSW-NB15$ dataset for: a) 10 rounds case; and b) 25 rounds case.



(a) (b)

Fig. 4: Confusion matrices of FedGame using $UNSW-NB15$ dataset for: a) 10 rounds case; and b) 25 rounds case.



(a) (b)

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

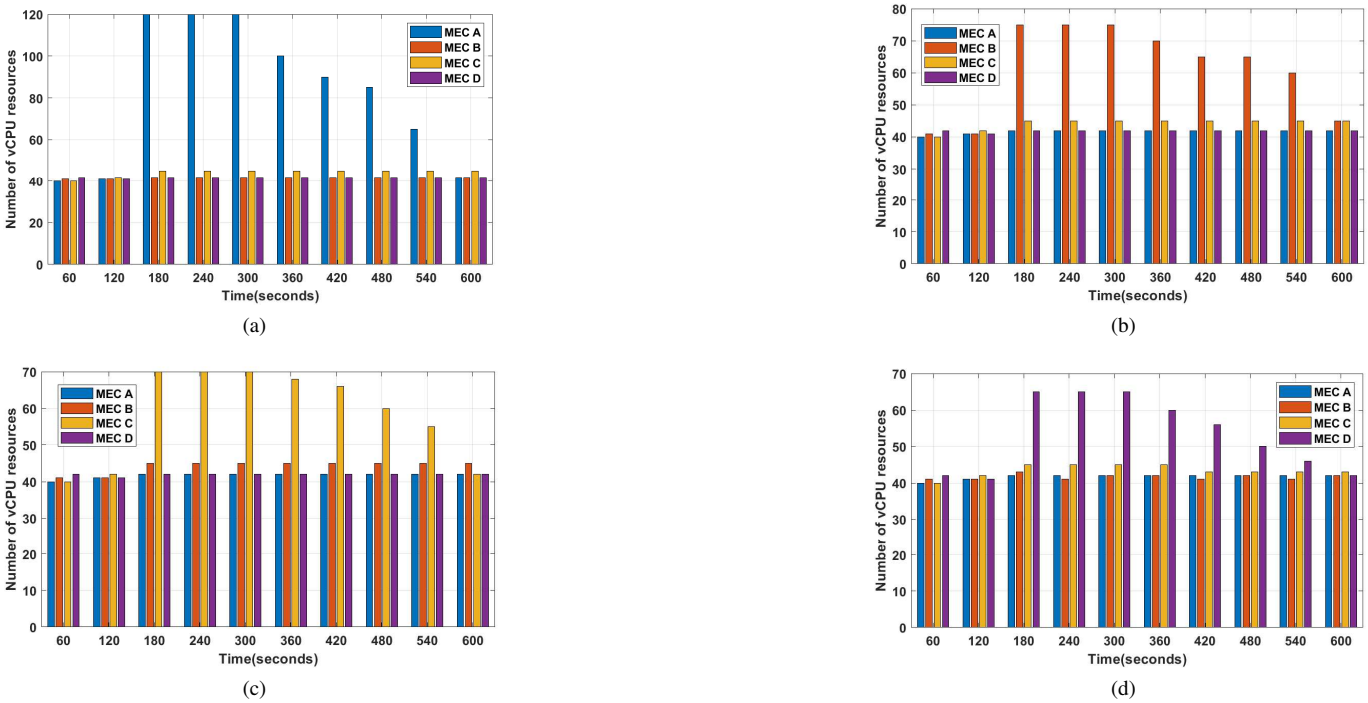


Fig. 6: Performance evaluation of FedGame, when generating IIoT's single source attacks at 180s. (a) A DDoS attack on MEC A; (b) An analysis attack on MEC B; (c) A fuzzers attack on MEC C; (d) A backdoors attack on MEC D.

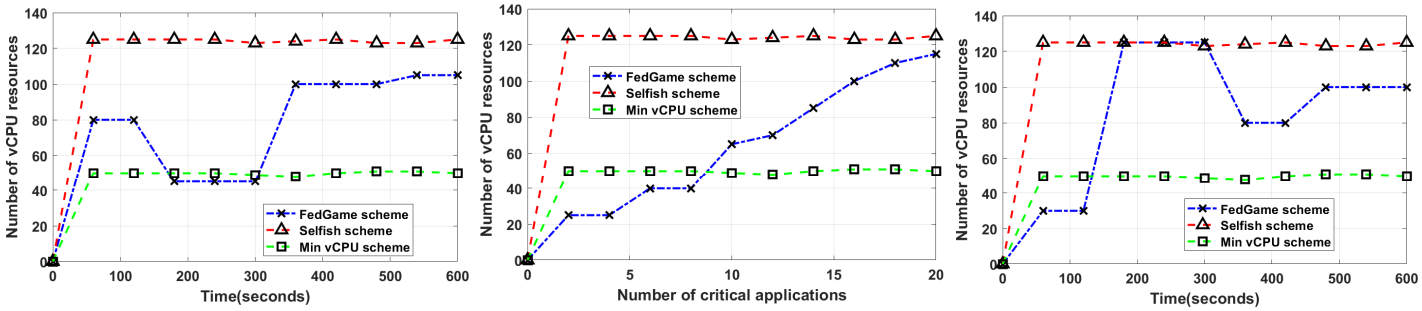


Fig. 7: Performance comparison between FedGame, selfish, and min vCPUs schemes.

99% in recall, accuracy, F1 score, and precision, respectively, with only 172 seconds 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 10 rounds 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 Confusion matrices of FedGame using the $UNSW-NB15$ dataset for 10 rounds and 25 of federated training, respectively. For 10 rounds of training, we observe that 99% of almost all 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 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 10 rounds and 25 rounds of federated training, respectively. The ROC curves show TPR according to FPR. For 10 rounds of training, we observe that FedGame has an Area Under the ROC Curve (AUC) score of 0.99, while it has a 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 seconds 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 600s. We generated different types of IIoT attacks: DDoS, analysis, fuzzers, and backdoors, addressing MEC A, B, C, D, respectively,

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
FedGame	**0.99**	**0.99**	**0.99**	**0.99**	**172**

at instant t=180s. 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 a 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 vCPUs assignment depends strongly on the attacks' priority in addition to the number of attackers (see Equation 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 vCPUs resources, to the compromised MEC nodes, while also ensuring a minimum and stable vCPUs 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 600s. Fig. 7(a) shows that the number of allocated vCPU is almost stable over time for both selfish and Min vCPU schemes, while it may vary for the FedGame. This is because either the number of critical applications which 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 vCPU remains stable for both selfish and Min vCPU schemes, whatever the number of critical applications. Indeed, the FedGame considers the number of critical applications, in allocating the vCPU resources to the MEC nodes, and the higher the number of applications, the more vCPUs are assigned (see Equation 3). Fig. 7(c) compares between the three schemes when generating a DDoS attack at t=180s and an analysis attack at t=480s. We clearly observe that the number of assigned vCPUs increases at t=180s 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. Afterwards, it increases again to 100 vCPUs, at t=480s due the analysis attack. We note that both selfish and Min vCPU schemes give a stable assignment behavior, given that they did not consider neither the IIoT attacks nor the critical applications, in their vCPUs assignment. Even the selfish way can provide the needed vCPUs to deal with IIoT attacks, however, most of time the allocated vCPUs remain unused, especially when there is 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 efficient way, while preserving the privacy of industrial systems' users. Furthermore, once an attack is detected, FedGame ensures a dynamic and efficient virtual resources allocation to MEC domains, which considers both attack priority and MECs' critical applications, in addition to the global system performance.

V. Conclusion

In this paper, we designed a new two-stage scheme, called FedGame, to secure Industrial systems against IIoT-based attacks. FedGame first leverages deep learning in a federated way to build a MEC-enabled prediction model for intrusion detection, while preserving users' privacy of industrial systems. In addition, once detecting an intrusion, a non-cooperative 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, and 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 source as well as multi-source attacks, to cover most of attacks that can target industrial systems.

References

[1] P. Maddikunta, Q. Pham, B. Prabadevi, N. Deepa, D. Kapal, T. Gadekallu, R. Ruby, M. Liyanage, Industry 5.0: A survey on enabling technologies and potential applications, Journal of Industrial Information Integration, 2021, 100257, ISSN 2452-414X.

[2] U. Kannengiesser, H. Müller, Towards viewpoint-oriented engineering for Industry 4.0: A standards-based approach. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 2018; pp. 51–56.

[3] Brik, B. Bettayeb, M. Sahnoun and A. Louis, "Accuracy and Localization-Aware Rescheduling for Flexible Flow Shops in Industry 4.0," 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), 2019, pp. 1929-1934, doi: 10.1109/CoDIT.2019.8820445.

[4] A. Omar, B. Imen, S. M'hammed, B. Bouziane and B. David, "Deployment of Fog Computing Platform for Cyber Physical Production System Based on Docker Technology," 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID), 2019, pp. 1-6, doi: 10.1109/ICAAID.2019.8934949.

[5] S. Morgan, "Cybercrime To Cost The World 10.5 Trillion Annually By 2025." [Online]. Available: https://cybersecurityventures.com/

[6] L.Horwitz,"The future of iot mini guide: The burgeoning iot market continues."[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," in AISTATS, 2017.

[8] Bouziane Brik, Mourad Messaadia, M'hammed Sahnoun, Belgacem Bettayeb, and Mohamed Amin Benatia. 2021. Fog-supported Low Latency Monitoring of System Disruptions in Industry 4.0: A Federated Learning Approach. ACM Trans. Cyber-Phys. Syst. Just Accepted (July 2021). DOI:10.1145/3477272

[9] A. Ksentini and P. A. Frangoudis, "Toward Slicing-Enabled Multi-Access Edge Computing in 5G," in IEEE Network, vol. 34, no. 2, pp. 99-105, March/April 2020, doi: 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)," 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1-6.

[11] N. Moustafa,"The future of iot mini guide: The burgeoning iot market continues."[Online].Available: www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets

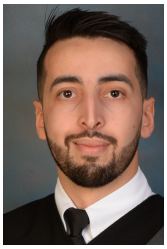
[12] J. Li, Z. Zhao, R. Li and H. Zhang, "AI-Based Two-Stage Intrusion Detection for Software Defined IoT Networks," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2093-2102, 2019.

[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," in IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5810-5818, 2021.

[14] Y. Jia, F. Zhong, A. Alrawais, B. Gong and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9552-9562, 2020.

[15] R. Ashfaq, X. Wang, J. Huang, H. Abbas, Y. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Information Sciences, Volume 378, 2017, Pages 484-497, ISSN 0020-0255,

[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," in IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6591-6607, 2021.
[17] K. Singh, S. Guntuku, A. Thakur, C. Hota, Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests, Information Sciences, Volume 278, 2014, Pages 488-497, ISSN 0020-0255.
[18] N. Ravi and S. M. Shalinie, "Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3559-3570, 2020.
[19] C. Mcdermott, A. Petrovski, 2017. Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks. International journal of computer networks and communications [online], 9(4), pages 45-56.
[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," in IEEE Transactions on Sustainable Computing, vol. 6, no. 2, pp. 245-256, 2021.
[21] N. Moustafa, E. Adi, B. Turnbull and J. Hu, "A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems," in 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 Communications Surveys Tutorials, 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," in IEEE Network, vol. 35, no. 3, pp. 74-80, 2021.
[24] Wang and G. -S. G. S. Kuo, "Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks — A Tutorial," in IEEE Communications Surveys Tutorials, vol. 15, no. 1, pp. 271-292, 2013.
[25] H. Nikaidô and K. Isoda, "Note on non-cooperative convex games" in Pacific Journal of Mathematics, 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, [Wiley, Econometric Society], 1965, pp. 520–34.



Zakaria Abou El Houda received the M.Sc. degree in computer networks from Paul Sabatier University, Toulouse, France, the Ph.D. degree in computer science from the University of Montreal, Canada and the Ph.D. degree in computer engineering from the University of Technology of Troyes, Troyes, France. His current research interests include ML/DL-based intrusion detection, Federated learning, and Blockchain.

Bouziane BRIK received engineering degree in computer science, ranked first in his class, and the Magister degree from the University of Laghouat, Algeria, in 2010 and 2013, respectively, and the Ph.D. degree from Laghouat and La Rochelle (France) universities, in 2017. He is currently working as associate professor at Burgundy (Bourgogne) university and DRIVE laboratory. Before joining Burgundy university, he was a post-doc at university of Troyes, CESI school, and Eurecom school. He has been working on network slicing in the context of H2020 European projects on 5G including MonB5G and 5GDrones. His research interests also include the Internet of Things (IoT), the IoT in industrial systems, smart grid, and vehicular networks. He also acted or still acts as a Reviewer of many IFIP, ACM, and IEEE conferences (ICC, Globecom, PIMRC, WCNC, etc.) and journals such as ACM and IEEE TRANSACTIONS.

Adlen Ksentini received the Ph.D. degree in computer science from the University of Cergy-Pontoise on QoS provisioning in the IEEE 802.11-based networks. Since 2016, he has been a Professor with the Communication Systems Department, EURECOM. He is currently an IEEE COMSOC Distinguished Lecturer on topics related to 5G and Network Softwarization. His current research topics are in the field of architectural enhancements to mobile core networks, mobile cloud networking, network functions virtualization, and SDN. He received the Best Paper Award from IEEE WCNC 2018, IEEE IWCMC 2016, IEEE ICC 2012, ACM MSWiM 2005, and the IEEE Fred W. Ellersik Prize for the Best IEEE Communications Magazine for 2017.

Lyes Khoukhi received the Ph.D. degree in electrical and computer engineering from the University of Sherbrooke, Canada, in 2006. From 2007 to 2008, he was a Post-Doctoral Researcher with the Department of Computer Science and Operations Research, University of Montreal. Currently, he is Full Professor with the ENSICAEN, Normandie University, GREYC CNRS. His current research topics are in the field of cybersecurity, attacks detection and performance evaluation in advanced networks like cloud networking, 5G/SDN, IoT/V2X and CPS.

Mohsen Guizani [s'85, M'89, Sm'99, F'09] (mguizani@ieee.org) received his B.S., M.S., and Ph.D. from Syracuse University. He is the Associate Provost for Faculty Affairs and Institutional Advancement at Mohamed Bin Zayed University of Artificial Intelligence. Previously, he worked at different institutions: the University of Idaho, Qatar University, Western Michigan University, the University of West Florida, the University of Missou-ri-Kansas City, the University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, applied machine learning, cloud computing, and security and its application to healthcare systems. He has more than 800 publications in these areas. He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019 and 2020. He has won several research awards, including the 2015 IEEE Communications Society Best Survey Paper Award as well as four Best Paper Awards from IEEE ICC and IEEE GLOBECOM conferences.