This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

Cybertwin-driven Resource Provisioning for IoE Applications at 6G-enabled Edge Networks

Mainak Adhikari, *Member, IEEE*, Ambigavathi Munusamy, *Member, IEEE*, Neeraj Kumar *Senior Member, IEEE*, and Satish Narayana Srirama, *Senior Member, IEEE*

***Abstract*—Cybertwin leverages the capabilities of networks and serves in multiple functionalities, by identifying digital records of activities of humans and things, from the Internet of Everything (IoE) applications. Cybertwin emerges as a promising solution along with next-generation communication networks, i.e., 6G technology, however, it increases additional challenges at the edge networks. Motivated by the above-mentioned perspectives, in this paper, we introduce a new cybertwin-driven edge framework using 6G-enabled technology with an intelligent service provisioning strategy, for supporting a massive scale of IoE applications. The proposed strategy distributes the incoming tasks from IoE applications using the Deep Reinforcement Learning technique based on their dynamic service requirements. Besides that, an Artificial Intelligence-driven technique, i.e., the Support Vector Machines (SVM) classifier model is applied at the edge network to analyze the data and achieve high accuracy. The simulation results over the real-time financial datasets demonstrate the effectiveness of the proposed service provisioning strategy and SVM model over the baseline algorithms in terms of various performance metrics. The proposed strategy reduces the energy consumption by 15% over the baseline algorithms, while increasing the prediction accuracy by 12% over the classification models.**

***Index Terms*—Cybertwin, Edge Computing, Internet-of-Everything, Resource provisioning, 6G networks, Data analytics.**

I. Introduction

In recent times, the Internet of Everything (IoE) plays a key role by enabling massive data-intensive applications including smart healthcare, VMR-based gaming, smart industry, etc., which requires technological advancement and the evolution of communication channels beyond the fifth-generation (5G) networks [1]. As per the estimation of the International Telecommunication Union, within 2030 real-time IoE applications will generate 4395 EB data traffic, where 5G networks would be unable to provide support to most of the next-generation IoE applications [2]. Therefore, the invention of 6G technology with terahertz communication media is expected to enhance the capabilities of 5G technology, where millions of IoE applications can transmit real-time data seamlessly to the remote computing device for further processing with minimum latency and high data rate [3].

Nowadays, the 6G technology plays a major role by supporting massive interconnectivity between the centralized cloud servers and IoE applications with highly diverse service requirements [4]. However, the centralized architecture of cloud computing increases the congestion of the network while increasing latency. Thus, one of the critical challenging tasks for the IoE application is to analyze the sensory data at the edge of the network with minimum latency and energy usage. In 2012, CISCO has invented a new technology, namely fog/edge computing that can provide the cloud servers at the edge of the networks, while reducing the latency and meeting the user's dynamic resource requirements [5]. Besides that, the collaborative edge-cloud framework with 6G technology can support various data-intensive and computation-sensitive IoE applications, with an efficient resource provisioning strategy.

Meanwhile, Cybertwin emerges as a promising technology for the future communication network, i.e., 6G-enabled Cybertwin technology goes beyond the concept of the digital twin, which is a virtual representation that performs the real-time digital replica of physical things or process. It serves as a contact hub and digital record of activities of the IoE applications in cyberspace at the edge networks. The current edge-cloud framework cannot address the dramatically increasing demands of the IoE application, which leads to the scalability issue in the network. Cybertwin leverages the capabilities of multi-access edge computing and provides multiple functionalities at the edge of the network while meeting the scalability requirements and Quality-of-Service (QoS) objectives of the 6G-enabled edge networks [2].

A. Related Studies

In recent times, the researchers have been focused to enhance service provisioning and edge intelligence by incorporating 6G network and Artificial Intelligence (AI) techniques for analyzing the massive amount of data at the edge of the network with low latency [6]. In [7], the authors have designed a 6G-enabled fog federation framework for processing IoE applications with an efficient service deployment strategy while reducing latency and energy usage. Similarly, Lin et al. have designed a collaborative AI-driven resource allocation strategy at 6G-enabled IoT networks [8]. In [9], the authors have designed a distributed probabilistic approach for IoE applications at 6G-enabled edge networks. Sanguanpuak et

Corresponding author: Satish Narayana Srirama.

M. Adhikari is with the Computer Science & Engineering, IIIT Lucknow, India. M. Adhikari is also associted with University of Tartu, Estonia. (e-mail: mainak.ism@gmail.com).

A Munusamy is with Anna University, Chennai, India, India. (e-mail: ambigaindhu8@gmail.com).

N. Kumar is with the Computer Science & Engineering, Thapar Institute of Engineering & Technology, Patiala. N.Kumar is also with School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand and Department of Computer Science and Information Engineering, Asia University, Taiwan. (e-mail: neeraj.kumar@thapar.edu).

S. N. Srirama is with the School of Computer and Information Sciences, University of Hyderabad, India (e-mail: satish.srirama@uohyd.ac.in).

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

al. have developed a resource sharing and caching technique with Stackelberg game technique at edge networks [10]. In [3], the authors have designed a new scheme at 6G-enabled edge networks with digital twin for IoE applications. Further, in [11]–[14], the authors have discussed various resource provisioning and management strategies for 6G-enabled IoE applications along with their research outcomes. A network architecture for 6G-enabled massive IoE applications with various AI techniques for enhancing edge intelligence is presented in [15]. An automatic and self-learning 6G-enabled edge network architecture is introduced in [6] to identify and classify unknown services.

In [16]–[18], the authors have developed various types of resource provisioning strategies for real-time applications at fog/edge networks while meeting various QoS objectives. Besides that, some of the important resource provisioning strategies along with their research outcomes for processing and analyzing real-time applications at fog/edge networks are highlighted in [19]–[21]. Nowadays, cybertwin is a new era for handling IoE applications with next-generation network architecture. In [22], the authors have designed a novel cybertwin-enabled network architecture with 6G technology for processing the IoE application efficiently using a collaborative edge-cloud framework. Further, Liang et al. have designed a resource trade mechanism with barter and combinatorial auction mechanism through cybertwin network for handling the network services efficiently with minimum cost [23]. The existing literature reviews state that most of the mechanisms apply the 6G and cybertwin technology for analyzing data at the edge of the networks. Thus, one of the important research aspects in the fields of IoE and edge computing is to handle the network resources efficiently while reducing the energy consumption of the network.

B. Motivations

From the theoretical and analytical point of view, the following key research questions need to be addressed while designing an intelligent resource provisioning strategy at cybertwin-driven edge networks: 1. how to develop an intelligent edge-centric framework by supporting cybertwin and 6G technology while reducing latency and energy consumption of the IoE applications? 2. how to design an intelligent resource provisioning strategy with an efficient AI technique at edge networks while meeting user's dynamic resource requirements? 3. how to analyze the IoE applications at edge networks using a standard classification model with collaborative edge-cloud computing servers?

The advanced cybertwin technique at edge-centric framework brings the computing and storage resources closer to the end-users by providing services at the edge of the networks. Besides that, the 6G technology supports terahertz communication for transferring the real-time data from IoE applications to remote computing devices through cybertwin server with minimum latency. Therefore, the critical and yet unsolved challenge is to design an intelligent resource provisioning strategy at cybertwin-driven edge networks to distribute the incoming applications to the local edge devices or centralized cloud servers through 6G-enabled communication links while reducing energy consumption. Moreover, analyzing the real-time data with higher accuracy with a feature selection strategy and AI technique is another important research aspect at the edge of the network.

C. Contributions

Motivated by the above-mentioned challenges, in this paper, we propose a new cybertwin-driven resource provisioning strategy for analyzing the IoE applications at the 6G-enabled edge networks. The major contributions of the work are summarized as follows.

- Develop an intelligent edge-centric framework with a cybertwin server as a communication assistant that can allocate computing and communication services coordinately at 6G-enabled networks with an AI-assisted resource provisioning strategy for handling user's dynamic service requests. The new and unique features of the cybertwin and 6G technology make the proposed edge-centric framework flexible, scalable while reducing delay and energy consumption for IoE applications.
- Design an intelligent resource provisioning strategy at the cybertwin server using the Deep Reinforcement Learning (DRL) technique that can provide the services at the edge of the network or centralized cloud servers as per the user's requirements. The proposed resource provisioning strategy meets the user's satisfaction ratio while utilizing the computing resources of the edge networks efficiently.
- Design an AI-enabled technique, namely Support Vector Machines (SVM) with a feature extraction strategy, i.e., Intonation Groups (IGs) for analyzing the IoE applications at the edge of the networks. The IGs strategy is used to extract the important features of the IoE applications for noise reduction and reduce the complexity of the incoming tasks for further data analysis. The proposed SVM technique is implemented at local edge devices for analyzing the selected features of the tasks while increasing the accuracy of the applications.
- Extensive simulation results and performance analysis demonstrate the effectiveness of the proposed resource provisioning strategy over the baseline algorithms in terms of various performance metrics. Besides that, the proposed SVM technique is evaluated with real-time financial datasets over the standard classification model in terms of accuracy and error.

The remaining sections of the paper are organized as follows. Section II demonstrates the system model followed by the problem formulation of the work. The proposed cybertwin-driven resource provisioning strategy at a 6G-enabled edge network is elaborated in Section III. The performance analysis of the proposed strategy is demonstrated in Section IV. Finally, Section V concludes the work.

II. System Model and Problem Formulation

Considering a cybertwin network with a set of K IoE devices, where $\mathcal{K} = \{1, 2, \ldots, K\}$, $\forall k \in K$, are taking services from the collaborative edge-cloud networks, which

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

is equipped with a set of local edge devices and centralized cloud servers [24], as depicted in Fig. 1. Let us consider that the requirements of the incoming tasks of the IoE application are represented by two tuples $(\mathcal{J}_k, \mathcal{L}_k)$, where $\mathcal{J}_k$ denotes the input size and $\mathcal{L}_k$ represents the application dependent delay requirement of the IoE device k, respectively. Considering an interference-free network with a set of distributed edge devices N, where $\mathcal{N} = \{1, 2, \dots, N\}$, $\forall n \in N$ and the transmission bandwidth is exclusively distributed among the edge devices in the network. Let $\gamma_{k,n}$ denotes a binary allocator of edge network, where

$$\gamma_{k,n} = \begin{cases} 1 & \text{if IoE application } k \text{ is assigned to edge device } N \\ 0 & \text{otherwise.} \end{cases}$$

Further, the requested computation frequency for device k can be modelled as $\mathcal{A}_k \mathcal{J}_k$, where $\mathcal{A}_k > 0$ indicates the number of CPU cycles that needed to complete a one bit computation. Let $\mathcal{X}_k (0 \leq \mathcal{X}_k \leq 1)$ be the amount partitioned input data executed by the IoE device k and the remaining $(1 - \mathcal{X}_k)$ tasks are offloaded to the remote computing devices for further processing. Therefore, the executable data on the IoE device is $\mathcal{X}_k \mathcal{J}_k$ and the edge device is $(1 - \mathcal{X}_k)\mathcal{J}_k$. In the following sub-sections, we discuss the delay and energy usage of the IoE applications at the edge of the network during local and remote execution.

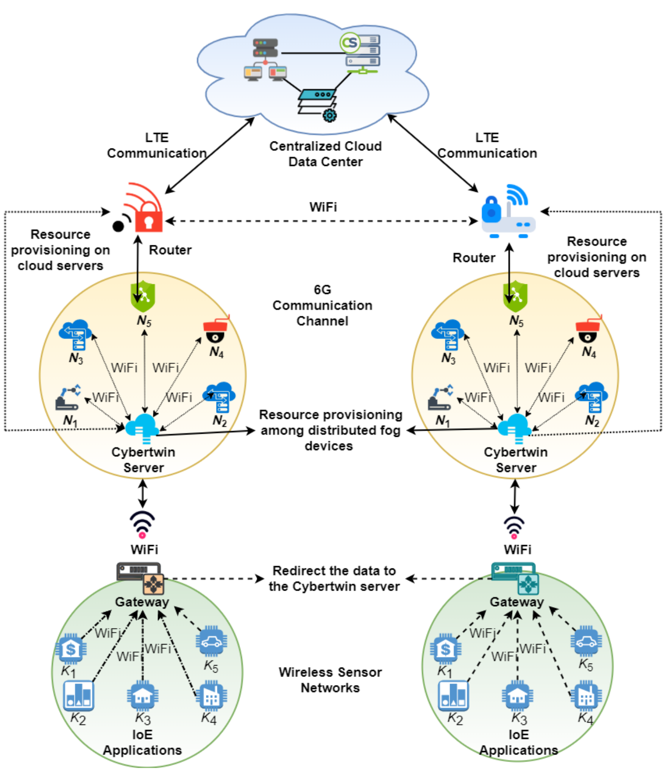

Fig. 1. Cybertwin-driven 6G-enabled edge networks.

A. Local Execution Model

Let $\mathcal{F}_k$ be the processing frequency of the IoE device k, $\forall k \in K$. The execution delay $\mathbb{T}_k^{user}$ on IoE device $k \in K$ for processing $\mathcal{X}_k \mathcal{J}_k$ amount of task can be expressed as follows.

$$\mathbb{T}_k^{user} = \frac{\mathcal{A}_k \mathcal{X}_k \mathcal{J}_k}{\mathcal{F}_k} \tag{1}$$

The overall power dissipation is represented as $\mathscr{P} = \varepsilon \mathcal{F}^3$, where ε denotes the chip coefficient depending on IoE devices $\mathcal{K}$ [24]. The corresponding energy consumption $\mathbb{E}_k^{user}$ on the local IoE device $k \in K$ can be expressed as follows.

$$\mathbb{E}_k^{user} = \mathcal{A}_k \mathcal{X}_k \mathcal{J}_k \varepsilon \mathcal{F}_k^2 \tag{2}$$

B. Resource Provisioning Model

Ideally, IoE devices have limited resource capacity and can process a limited scale of data locally, which are generated by the sensor or IoE devices. However, the inadequacy of computation capability $\mathcal{F}_k$ on the IoE devices creates a barrier for local execution and distributes the tasks on the remote edge devices through a cybertwin server. The local cybertwin server brings the centralized cloud servers on the edge of the network for the IoE applications and distributes the tasks through an intelligent resource provisioning strategy while minimizing delay and energy consumption. Let $\mathscr{D}_k^{up}$ and $h_{k,n}^{up}$ be the transmission bandwidth and instantaneous channel power gain of the IoE device $k \in K$ [24]. The data transmission rate $\mathscr{R}_{k,n}$ on the remote edge device $n \in N$ is represented as follows.

$$\mathscr{R}_{k,n} = \mathscr{D}_k^{up} log_2 \left(1 + \frac{\mathscr{P}_{k,n} h_{k,n}^{up}}{\mathscr{D}_k^{up} \sigma^2}\right) \tag{3}$$

where σ^2 and $\mathscr{P}_{k,n}$ denotes the noise spectral density and transmission power of the IoE device $k \in \mathcal{K}$. Thus, the uplink transmission rate $\mathscr{R}_k$ and corresponding energy consumption $\mathscr{P}_k$ by IoE device k are represented as follows.

$$\mathscr{R}_k = \sum_{n=1}^{N} \gamma_{k,n} \mathscr{R}_{k,n} \tag{4}$$

$$\mathscr{P}_k = \mathscr{P}_0 + \delta \sum_{n=1}^{N} \gamma_{k,n} \mathscr{P}_{k,n} \tag{5}$$

where $\mathscr{P}_0$ and δ denotes the static energy consumption and inverse of the power amplifier efficiency, respectively. The uploading transmission delay $\mathbb{T}_k^{upload}$ and energy consumption $\mathbb{E}_k^{upload}$ with data size $(1 - \mathcal{X}_k)\mathcal{J}_k$ can be expressed as follows.

$$\mathbb{T}_k^{upload} = \frac{\beta_k (1 - \mathcal{X}_k) \mathcal{J}_k}{\mathscr{R}_k} \tag{6}$$

$$\mathbb{E}_k^{upload} = \mathscr{P}_k \mathbb{T}_k^{upload} \tag{7}$$

where β_k denotes the additional transmission overhead over the channel. $\mathcal{F}_k$ represents the allocated computational frequency by the remote edge device. The execution delay $\mathbb{T}_k^{server}$ and energy consumption $\mathbb{E}_k^{server}$ on the remote edge device can be represented as follows.

$$\mathbb{T}_k^{server} = \frac{\mathcal{A}_k (1 - \mathcal{X}_k) \mathcal{J}_k}{\mathcal{F}_k} \tag{8}$$

$$\mathbb{E}_k^{server} = \mathcal{A}_k (1 - \mathcal{X}_k) \mathcal{J}_k \varepsilon \mathcal{F}_k^2 \tag{9}$$

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

Based on the above formulations of uploading, and processing, the overall delay and energy consumption of the IoE applications on the remote edge devices are represented as follows.

$$\mathbb{T}_k^{offload} = \mathbb{T}_k^{upload} + \mathbb{T}_k^{server} \tag{10}$$

$$\mathbb{E}_k^{offload} = \mathbb{E}_k^{upload} + \mathbb{E}_k^{server} \tag{11}$$

C. Problem Formulation

The key objective of the cybertwin-driven edge networks is to derive the sum of weighted energy consumption $w_k\mathbb{E}_k(.)$ of all the IoE users K, which can be obtained by considering CPU frequency of the IoE device $\mathcal{F}_l = (\mathcal{F}_k)_{\forall k}$, CPU frequency of the remote edge devices $\mathcal{F}_c = (\mathcal{F}_k)_{\forall k}$, edge device allocation $\gamma = (\gamma_{k,n})_{\forall k,n}$, transmission power $\mathscr{P} = (\mathscr{P}_{k,n})_{\forall k,n}$, and offloading ration $\mathcal{X} = (\mathcal{X}_k)_{\forall k}$. Therefore, the mathematical formulation of the objective function with corresponding constraints for developing an efficient resource provision strategy is formulated as follows.

$$\underset{\mathcal{F}_l, \mathcal{F}_c, \gamma, \mathscr{P}, \mathcal{X}}{\text{minimize}} \quad \sum_{k=1}^{K} w_k \mathbb{E}_k(\mathcal{F}_l, \mathcal{F}_c, \gamma_{k,n}, \mathscr{P}_{k,n}, \mathcal{X}_k) \tag{12a}$$

$$\text{subject to} \quad \sum_{n=1}^{N} \gamma_{k,n} \mathscr{P}_{k,n} \leq P_{T_k}, \tag{12b}$$

$$\sum_{k=1}^{K} \mathcal{F}_k \leq F_c, \tag{12c}$$

$$\sum_{k=1}^{K} \gamma_{k,n} \leq 1, \tag{12d}$$

$$\max \ \{\mathbb{T}_k, \mathbb{T}_k\} \leq \mathcal{L}_k, \tag{12e}$$

$$\mathcal{F}_k \geq 0; \mathscr{P}_{k,n} \geq 0 \tag{12f}$$

The objective of the resource provisioning at cybertwin-driven edge networks for minimizing the overall energy consumption for the IoE applications is presented in (12a), subject to satisfying constraints (12b)-(12h). Where constraints (12b) represents the maximum transmission power over the remote edge device n. Constraint (12c) denotes the allocated CPU frequency must be limited by the capacity of the edge device. Constraint (12d) denotes the maximum edge device assigned for an IoE device is 1. The latency bound for each IoE application presented in constraint (12e). Finally, constraint (12f) represents the non-negative feature of computation resource and transmission power of the remote edge device.

III. Cybertwin-driven Resource Provisioning

In this section, we design an intelligent resource provisioning strategy using the DRL technique for distributing the incoming applications of IoE applications on the local edge devices through a cybertwin server while minimizing energy consumption and delay. Besides that, a feature selection technique, namely IGs technique is incorporated in the local edge devices for extracting the important features of the incoming data of IoE applications. Finally, the SVM technique analyzed the extracted features for further prediction with higher accuracy.

A. Intelligent Resource Provisioning Strategy

To design an efficient resource provisioning strategy, the DRL technique is introduced in the cybertwin server using state (S), action (A), and reward (R) for distributing the incoming tasks on the remote edge devices. The key aim to use this formulation is to obtain maximum reward from the environment. In each state S, the local cybertwin server takes action A and reaches the next state S' and obtains immediate reward R(S'/SA) using a stationary policy π, where $\pi(S) = A$. However, to obtain optimal policy $\pi*$, the agent has to solve the state-action pair using Bellman's equation, defined as follows.

$$\pi^*(S) = arg \max_{A \in \mathbb{A}} Q(S, A) \tag{13}$$

$$Q^*(S) = \max_{A \in \mathbb{A}} Q(S, A) \tag{14}$$

and the function Q(S,A) can be defined as follows

$$Q(S, A) = \sum_{S' \in \mathbb{S}} P(S'|S, A) \left[R(S'|S, A) + \mathcal{X} Q^*(S)\right] \tag{15}$$

where $\mathcal{X}$ is a discount factor and is defined as $\mathcal{X} \in (0, 1)$. One simple approach to solve this Bellman equation is dynamic programming. Although dynamic programming requires state transition probability in each step, which is uncertain due to the dynamicity of the edge devices and network parameters. Q-learning is a well-known algorithm popularly used for approximating state transition probability, which is defined as follows.

$$\begin{aligned} Q(S_t, A_t) = Q(S_t, A_t) + \\ \mu_t \Big[R_{t+1} + \mathcal{X} \max_A Q(S_{t+1}, A) - Q(S_t, A_t)\Big] \end{aligned} \tag{16}$$

where S_t is the current state, S_{t+1} is the next state and R_{t+1} is the next state reward. However, the complexity of the Q-learning equation increases with the increase on state action pair. A promising approach to solve Q-values in the Q-Learning equation is the DQN. A DQN strategy mainly contain three components such as state, action and reward, which are defined as follows.

- *State:* A state S represents the information about the resource provisioning decision variable on the cybertwin networks. The states in the cybertwin networks can be defined as $S = s = \{\gamma_{kn}\}$.
- *Action:* The action A in the proposed network represents the workload distribution decision for any given state. The action space can be derived as $A = a = \{\gamma'_{kn}\}$.
- *Reward:* The key intuition of the DRL strategy is to maximize the utility over the edge network while minimizing the energy consumption. The reward of the edge network can be defined as the difference between two consecutive energy consumption functions as $R = \mathbb{E}_k(.) - \mathbb{E}'_k(.)$.

DQN is a Reinforcement Learning (RL) approach, where Deep Learning (DL) models are developed to determine the operations of an agent at each time stamp. In DQN-based learning, the cybertwin server makes offloading action a_t from state s_t in such a way that maximizes the future reward in the long run. The DQN function can be derived as follows [25].

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

Algorithm 1: DRL-based resource provisioning

INPUT: $\mathcal{J}_k$: Input size and $\mathcal{L}_k$: Deadline requirement
OUTPUT: $\pi^*(s_t)$: optimal decisions
Initialize s_t, a_t, $\mathcal{K}$ and Θ_t
for $p = 1$ to **do**
  On state s_t execute action a_t
  Observe s_{t+1}, r_t
  calculate $\mathbb{E}_k(\mathcal{F}_l, \mathcal{F}_c, \gamma_{k,n}, \mathscr{P}_{k,n}, \mathcal{X}_k)$
  Store (s_t, a_t, r_t, s_{t+1}) into the memory
  Select mini-batch examples from memory
  Calculate $\mathcal{Y}_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \Theta_{t-1})$
  Update loss function using $\mathcal{Y}_t$ and $Q(s_t, a_t)$
  $\mathcal{L}_t(\Theta_t) = \mathbb{E}\left[(\mathcal{Y}_t - Q(s_t, a_t; \Theta_t))^2\right]$
  Update network weight Θ
  Take optimal decision $\pi^*(s_t)$
end for

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \sum_{k=1}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a, \pi\right] \tag{17}$$

where γ is the discount factor. Moreover, the Q-network is trained with weight w to minimize the loss $L(\theta_t)$ at timestamp t, which is defined as follows.

$$L_t(\theta_t) = \mathbb{E}\left[\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta_{t-1}) - Q(s_t, a_t; \theta)\right)^2\right] \tag{18}$$

The steps for the proposed DQN-based resource provisioning strategy are presented in Algorithm 1. It is worth noting that the proposed DRL strategy initially starts with random action. However, after a few iterations, DRL starts making optimal decisions based on obtaining experience. This process gradually optimizes the decision-making time and helps to achieve minimal energy emission from the industrial environment.

B. Feature Extraction Strategy

The massive amount of data, collected from IoE devices is difficult to analyze and store at the local edge device. Most of the data generated from these devices contain redundant and irrelevant features, which imposes processing overhead on the edge network, reduces the prediction accuracy and feature dimensionality. Therefore, the inclusion of feature extraction techniques at the edge networks can greatly minimize the processing overhead, training time, and help to improve the prediction accuracy of the AI models. Stimulated by this idea, we introduce IGs feature extraction technique to precisely extract the data features from IoE applications. Let $\mathcal{U}$ and $\mathcal{V}$ be the difference between two feature variables. Then, IGs selects a distinct feature as given below.

$$IG(\mathcal{U}|\mathcal{V}) = H(\mathcal{U}) - H(\mathcal{U}|\mathcal{V}) \tag{19}$$

where $\mathcal{U}$ and $\mathcal{V}$ denote the discrete random variables. The prior entropy of feature $\mathcal{U}_i$ is defined as follows.

$$H(\mathcal{U}) = -\sum_i P(\mathcal{U}_i) log_2 \mathcal{P}(\mathcal{V}_i) \tag{20}$$

where $\mathcal{P}(\mathcal{U}_i)$ represent the prior probability of $\mathcal{U}_i$. Let $\mathcal{U}$ be the conditional entropy after substituting the post entropy of feature $\mathcal{V}$, formulated as follows.

$$H(\mathcal{U}|\mathcal{V}) = \begin{cases} -\sum_i \mathcal{P}(\mathcal{U}_j) H(\mathcal{U}|\mathcal{V}_i) \\ -\sum_i \mathcal{P}(\mathcal{U}_j) \sum_i \Big(\mathcal{P}(\mathcal{U}_i|\mathcal{V}_j) log_2 \mathcal{P}(\mathcal{U}_i|\mathcal{V}_j)\Big) \end{cases} \tag{21}$$

By substituting the terms of Equations (20) and (21) into Equation (19), we can obtain the final value of IGs as follows.

$$IG(\mathcal{U}|\mathcal{V}) = -\sum_i \mathcal{P}(\mathcal{U}_i) log_2 \mathcal{P}(\mathcal{U}_i) - \left(-\sum_i \mathcal{P}(\mathcal{U}_j) \sum_i \Big(\mathcal{P}(\mathcal{U}_i|\mathcal{V}_j) log_2 \mathcal{P}(\mathcal{U}_i|\mathcal{V}_j)\Big)\right) \tag{22}$$

The IGs technique performs well with a minimum number of feature sets. Thus, it is very difficult to select the most significant features from the large volume of IoE data. To address this issue, we assign feature weight to each feature to efficiently remove the irrelevant and redundant features. Let $\mathcal{C}$ be the set of feature variables of a particular class $\mathcal{C} = \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \ldots, \mathcal{C}_k$, where $\forall c_k \in \mathcal{C}$. Then, an instance of a particular class with weight is represented as $\mathcal{W} = \mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3, \ldots, \mathcal{W}_k$, where $\forall w_k \in \mathcal{W}$. Thus, the weight of a particular instance can be expressed as follows.

$$\Omega_{(c_k, w_k)} = -\sum_k \mathcal{P}(\mathcal{Q}|w_k) log \frac{\mathcal{P}(\mathcal{Q}|w_k)}{\mathcal{P}(\mathcal{Q})} \tag{23}$$

where $\mathcal{Q}$ represents the weight of a specific feature set w_k of the class $\mathcal{Q}$. The values of $\mathcal{W}_{(c_k, u_k)}$ should be in the range of [0, 1], which is related to feature variable c_k of the class $\mathcal{Q}$. The overall procedures of feature extraction and predictive analytic model are described in Algorithm 2.

C. Edge-cloud Collaborative Data Analytics

The volume of data traffic on IoE applications is constantly increasing and highly demanding instant predictions and identifications of digital records of human activities and things, and better service requirements. However, handling a huge volume of irrelevant or redundant data is one of the most crucial issues in edge-centric IoE applications. Further, the transmission of a huge volume of data from IoE devices to the centralized cloud server increases the processing overhead, power consumption, and delay. Such applications must require an edge-level analysis with an efficient feature extraction technique and predictive analysis model with higher accuracy.

However, due to less storage and processing capacity of local edge devices, only the critical features, extracted using the IGs technique are analysed at the resource-constraint edge devices and the rest of the data is stored and analyzed on the remote cloud server for future predictions. For this validation, we consider a standard SVM model as the baseline model to predict and classify various labels based on the extracted feature set. The main reason behind introducing

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

Algorithm 2: Feature Extraction and Data Analysis

1 **INPUT :** $\mathcal{W}$: Weight of an instance, $\mathcal{Q}$: Weight of a particular feature set, $\mathcal{C}$: Set of feature variables of a particular class, k_s: Number of iterations

2 **OUTPUT :** $\Omega_{(c_k,w_k)}$: Extracted feature set and classification accuracy

1: Select distinct feature variables
2: Calculate the conditional probability using Eq.no (21)
3: Obtain final value of IGs using Eq.no (22)
4: Compute weight of a particular instance using the formula $\Omega_{(c_k,w_k)} = -\sum_k \mathcal{P}(\mathcal{Q}|w_k) log \frac{\mathcal{P}(\mathcal{Q}|w_k)}{\mathcal{P}(\mathcal{Q})}$
5: Normalize the datasets
6: Apply SVM classification model over datasets
7: **for** $i = 1$ to m **do**
8:     Sort $\mathcal{W} = \mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3, \ldots, \mathcal{W}_k$
9:     Select $\mathcal{W}_k$ according to its $\mathcal{C}_k$
10:     Store newly generated feature set $\Omega_{(c_k,w_k)}$
11:     Sort S_r= (Sort $\mathcal{C} = \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k + m$), K_s)
12: **end for**

SVM technique at edge network is to its relative memory efficiency, support for high dimensional dataset, and increased prediction accuracy while analysing IoE applications [26].

IV. Numerical Analysis

In this section, we analyze the performance of the proposed DRL-based resource provisioning strategy in terms of delay, energy consumption, and reward. Moreover, to present the effectiveness of the proposed strategy, we compare the proposed strategy over three baseline algorithms such as Local Execution (`LE`), Random Execution (`RE`), and Server (`SE`) framework. A short description of these algorithms is explained below.

- *Local Execution* (`LE`) : In the `LE` strategy, IoE devices process all the computation locally without considering the server execution.
- *Random Execution* (`RE`) : In the `RE` strategy, IoE devices offload and process tasks on remote computing devices using a randomized selection strategy.
- *Server Execution* (`SE`) : In the `SE` strategy, tasks are processed on remote cloud server without considering any proper task assignment strategy.

Furthermore, we consider a standard SVM classification model at both edge server and cloud server to analyse and evaluate the most significant features extracted from IoE applications. To show the superiority of the proposed predictive model, we validate our SVM model with existing classification models such as Logical Regression (LR), Random Forest (RF), and Decision Tree (DT) under different performance metrics including accuracy, precision, recall, and F1 score, respectively.

A. Simulation Setup

For the simulation setup, we use an Intel i7 CPU@ 3.40 GHz system with 16GB RAM and Python platform for implementing this work. More specifically, we use GYM OpenAI, an open-source python library, which is popularly used for implementing DRL-based techniques. We consider the number of IoE devices = 500 and edge server = 10, uniformly distributed over the network. We set the clock frequency of the IoE device = 5 and edge server = 10, respectively. Furthermore, we consider transmission bandwidth = 5, transmission power = 5, η = 10^{-19}. Other simulation parameters of 6G-enabled edge networks are presented in Table I.

TABLE I
Parameters Used For Simulation

Parameters	Values	Parameters	Values
W_k	1	$\mathcal{N}$	[4,7]
$\mathscr{D}_k^{up}$	100GHz	δ	1
$\gamma_{k,n}$	[0,1]	$\mathcal{D}_i$	1900
$\mathcal{F}_i$	$5{\times}10^7$	$\mathcal{F}_J$	$50{\times}10^9$
λ_i	0.5	$\mathscr{P}_0$	500mW

- Dataset: To empirically validate the prediction accuracy of the proposed predictive model, we consider two distinct real-time datasets such as loan approval prediction dataset (D1)[1] and lending club loan dataset (D2) [2], respectively. The dataset D1 is used for edge-level analysis, which contains 615 instances and 13 attributes. Similarly, D2 is considered for cloud-level analysis that consists of 887379 instances and 74 attributes. For this validation, we split the dataset into 80:20 ratio, where 80 % of the data is assumed for training the model while 20% of the data is used for testing the model.

B. Average Energy Consumption

The energy consumption of the 6G-enabled edge networks depends on the computation ($\mathbb{E}_k^{user}$) and communication ($\mathbb{E}_k^{upload}$) energy while processing and transmitting the data of IoE applications on remote computing devices, respectively. The proposed DRL-driven resource provisioning strategy distributes the incoming tasks on the local edge devices through the cybertwin server while observing dynamic changes of the network parameters and input task parameters $(\mathcal{J}_k, \mathcal{L}_k)$. Fig. 2 represents the comparative analysis of the proposed DRL resource provisioning strategy over the baseline algorithms. The comparative analysis demonstrates that the proposed strategy outperforms the baseline algorithms in terms of energy consumption up to 15% while processing the incoming tasks on 6G-enabled edge networks.

C. Average Delay

The average delay of the IoE applications depends on the uploading time ($\mathbb{T}_k^{upload}$) of the incoming tasks on the remote computing devices before start processing on it. Thus, an intelligent resource provisioning strategy distributes the incoming tasks on the local edge devices based on the requirements of the incoming tasks $(\mathcal{J}_k, \mathcal{L}_k)$ and the resource availability of the edge devices. The proposed DRL-driven resource provisioning strategy distributes the incoming tasks intelligently on the remote edge devices through a local cybertwin server.

[1]https://www.kaggle.com/ajaymanwani/loan-approval-prediction
[2]https://www.kaggle.com/aamirsiddiqui/lending-club-loan-machine-learning

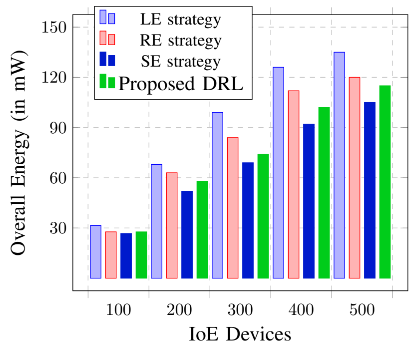

Fig. 2. Energy consumption of intelligent resource provisioning strategy.

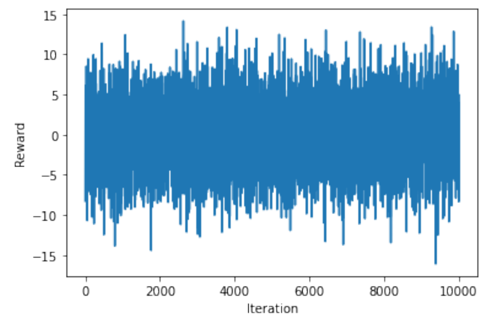

Fig. 4. Reward using random action strategy.

Besides that, THz communication of the 6G network helps to reduce the delay of the IoE applications while uploading the tasks on the remote edge devices. Fig. 3 represents the comparative analysis of the proposed DRL-driven resource provisioning strategy over the baseline algorithm. The comparative analysis depicts that the proposed strategy outperforms with 23% of delay minimization compared with the baseline algorithms due to intelligent tasks distribution using cybertwin server and THz communication of 6G networks.

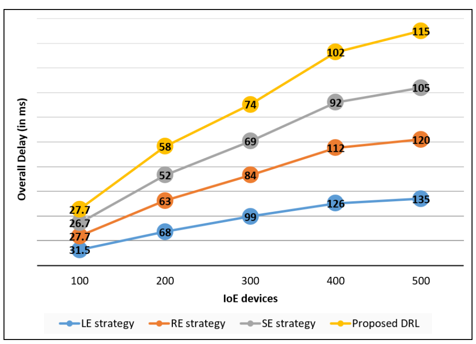

Fig. 3. Average delay of intelligent resource provisioning strategy.

D. Reward

To find the average reward for each iteration, we consider learning rate 0.001, batch size 32, decay rate 0.7, and the number of iterations 10000. We consider the mean absolute error as the loss function and the bellman equation as the policy. Further, we use memory size 5000 bytes and four different optimizers such as Adam, SGD, RMSprop, and Adamax. Fig. 4 represents the cumulative reward by using a random action strategy. From the figure, it is observed that the variation between maximum and minimum reward in each iteration is large. The reason behind that the random action strategy starts with randomly choosing the action state, and this continues in all the iterations rather than selecting a predicted action state. This causes lower reward calculation from the environment.

However, by using the proposed DRL-driven resource provisioning strategy, as presented in Fig. 5, the deviation was minimized to small rage. It can also be observed from the experiment that the Adam optimizer performs well as compared with other optimizer to reduce the overall energy consumption rate in the cybertwin networks.

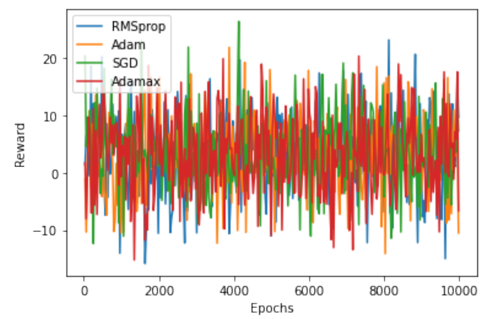

Fig. 5. Reward using proposed DRL strategy.

E. Feature Extraction and Analysis on Computing Devices

1) Predictive Analysis at Edge Server: Fig. 6 shows the edge-level analysis of various classification models under dataset (D1). From the analysis, it is noticed that the SVM classification model yields 87.23% accuracy while predicting the status of loan approval as compared to other classification models such as LR, DT, and RF. Besides that, after extracting the significant features from D1, the overall accuracy of SVM is greatly improved, i.e., 95.39 %, which is higher than the baseline classification models. Table II presents the prediction accuracy of various classification models in the distributed edge servers.

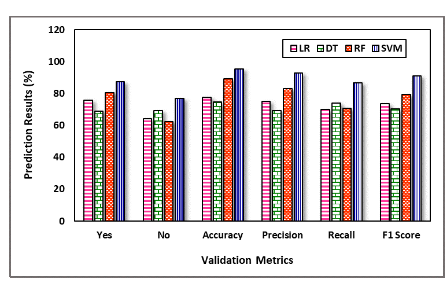

Fig. 6. Prediction Accuracy Results of D1 at distributed edge servers.

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3096672

TABLE II
PREDICTION ACCURACY OF VARIOUS CLASSIFICATION MODELS IN EDGE SERVER.

Edge Level Analysis							
Dataset	**Classification Models**	**Loan Approval**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Yes	**No**				
D1	LR	0.7586	0.6425	0.7754	0.7513	0.6987	0.7362
	DT	0.6895	0.6942	0.7462	0.6906	0.7381	0.7045
	RF	0.8051	0.6233	0.8921	0.8311	0.7054	0.7929
	SVM	0.8723	0.7688	**0.9539**	0.9268	0.8652	0.9087

2) Predictive Analysis at Cloud Server: Fig. 7 shows the cloud-level analysis of various classification models under data set (D2). From the analysis, it is also proved that the SVM classification model achieves 66.52 % accuracy while predicting the status of lending load whether the customer has fully paid or not. Further, the SVM model enhances the overall accuracy by 31.9% as compared with other baseline models based on the significant feature sets. However, other models such as LR, DT, and RF are improved only by 24.32%, 0.08%, and 28.65%. Table III shows the prediction accuracy of various classification models in the centralized cloud servers.

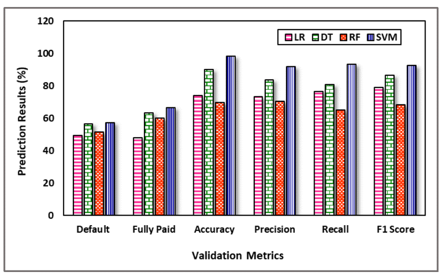

Fig. 7. Prediction Accuracy Results of D2 at centralized cloud servers.

3) Analysis of Training time: Fig. 8 presents the training time of the standard classification models. The training time of the SVM is less, i.e., 0.28 seconds after injecting the optimal features. However, the training time of LR and RF is much higher, i.e. 0.42 and 0.37, respectively. Thus, we conclude that the selection of the most critical set of features with the SEM model at the edge server can improve the overall prediction accuracy of the models.

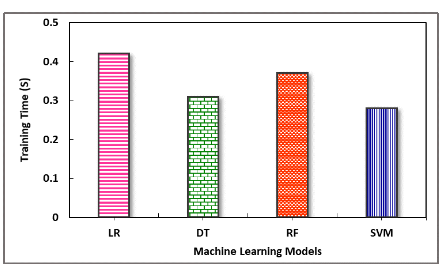

Fig. 8. Training time of edge-cloud collaboration

V. CONCLUSION

In this paper, we have developed a new cybertwin-driven edge-centric framework by supporting 6G technology for analyzing IoE applications efficiently. For handling the incoming tasks of the applications, an intelligent resource provisioning mechanism is designed the cybertwin server with a DRL model that can distribute the incoming tasks on the remote edge devices as per their requirements with minimum latency and energy consumption. Besides that, the SVM classification model is introduced at the remote computing devices at the edge network along with an IGs-enabled feature selection strategy for analyzing the tasks of the IoE application efficiently with critical features while increasing prediction accuracy. The experimental evaluation of the proposed resource provisioning and data analytics demonstrate the effectiveness of the proposed strategies over the baseline algorithms and standard classification models respectively. From the simulation results, it is observed that the proposed resource provisioning strategy reduces 10%-15% energy consumption over baseline algorithms and the proposed SVM model improves 10%-12% accuracy over the standard classification models. In the future, we will enhance the cybertwin-driven edge-centric framework by incorporating mobility and privacy techniques to improve its capability in a dynamic environment.

ACKNOWLEDGEMENT

We acknowledge financial support to UoH IoE by MHRD (F11/9/2019-U3(A)).

REFERENCES

[1] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based iot (feciot): Architecture, applications, and research issues," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4118–4149, 2019.
[2] Q. Yu, J. Ren, Y. Fu, Y. Li, and W. Zhang, "Cybertwin: An origin of next generation network architecture," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 111–117, 2019.
[3] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 240–12 251, 2020.
[4] Y. Sun, J. Liu, J. Wang, Y. Cao, and N. Kato, "When machine learning meets privacy in 6g: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2694–2724, 2020.
[5] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.
[6] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. V. Poor, "Toward self-learning edge intelligence in 6g," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 34–40, 2020.
[7] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Stackelberg game for service deployment of iot-enabled applications in 6g-aware fog networks," *IEEE Internet of Things Journal*, pp. 1–10, 2020.
[8] K. Lin, Y. Li, Q. Zhang, and G. Fortino, "Ai-driven collaborative resource allocation for task execution in 6g enabled massive iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5264–5273, 2021.

TABLE III
PREDICTION ACCURACY OF VARIOUS CLASSIFICATION MODELS IN CLOUD SERVER

Cloud Level Analysis							
Dataset	**Classification Models**	**Lending Loan**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Default	**Fully paid**				
D2	LR	0.4929	0.4783	0.7411	0.7326	0.7658	0.7901
	DT	0.5653	0.6316	0.8995	0.8354	0.8081	0.8665
	RF	0.5167	0.5995	0.6978	0.7065	0.6518	0.6847
	SVM	0.5731	0.6652	**0.9843**	0.9203	0.9347	0.9256

[9] Z. Liao, J. Peng, J. Huang, J. Wang, J. Wang, P. K. Sharma, and U. Ghosh, "Distributed probabilistic offloading in edge computing for 6g-enabled massive internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5298–5308, 2020.

[10] T. Sanguanpuak, D. Niyato, N. Rajatheva, and M. Latva-aho, "Radio resource sharing and edge caching with latency constraint for local 5g operator: Geometric programming meets stackelberg game," *IEEE Transactions on Mobile Computing*, vol. 2, no. 2, pp. 707–721, 2021.

[11] P.-C. Chen, Y.-C. Chen, W.-H. Huang, C.-W. Huang, and O. Tirkkonen, "Ddpg-based radio resource management for user interactive mobile edge networks," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.

[12] Z. Lv and N. Kumar, "Software defined solutions for sensors in 6g/ioe," *Computer Communications*, vol. 153, pp. 42–47, 2020.

[13] A. Shukla, R. Gupta, S. Tanwar, N. Kumar, and J. J. Rodrigues, "Block-ras: A p2p resource allocation scheme in 6g environment with public blockchains," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[14] B. Ji, Y. Wang, K. Song, C. Li, H. Wen, V. G. Menon, and S. Mumtaz, "A survey of computational intelligence for 6g: Key technologies, applications and trends," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2021.

[15] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive iot toward 6g: A comprehensive survey," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[16] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Joint computation offloading and scheduling optimization of iot applications in fog networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3266–3278, 2020.

[17] M. Adhikari, S. N. Srirama, and T. Amgoth, "Application offloading strategy for hierarchical fog environment through swarm optimization," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4317–4328, 2019.

[18] M. Adhikari, M. Mukherjee, and S. N. Srirama, "Dpto: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5773–5782, 2019.

[19] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 260–288, 2019.

[20] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management and evaluation of fog computing systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2494–2516, 2021.

[21] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial internet of things security: Requirements and fog computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[22] Q. Yu, J. Ren, H. Zhou, and W. Zhang, "A cybertwin based network architecture for 6g," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.

[23] H. Liang and W. Zhang, "A barter and combinatorial auction based hierarchical resource trade mechanism for cybertwin network," in *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2020, pp. 84–89.

[24] M. Sheng, Y. Wang, X. Wang, and J. Li, "Energy-Efficient Multiuser Partial Computation Offloading With Collaboration of Terminals, Radio Access Network, and Edge Server," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1524–1537, 2020.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[26] J. Wang, L. Wu, H. Wang, K.-K. R. Choo, and D. He, "An efficient and privacy-preserving outsourced support vector machine training for internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 458–473, 2021.

Mainak Adhikari is currently working as an Assistant Professor at IIIT Lucknow, India. He has completed Post Doctorate Research Fellowship from University of Tartu, Estonia in the year of 2021. He has obtained his Ph.D from IIT(ISM) Dhanbad, India in 2020. He has obtained his M.Tech. and B. Tech from Kalyani University in 2013 and West Bengal University of Technology in 2011, respectively. He is the Associate Editor of Cluster Computing and Physical Communication and Technical Committee member of Computer Communication.

Ambigavathi Munusamy obtained her B.E degree in CSE from Anna University, Chennai, Tamil Nadu, India in 2010 and her M.E degree in CSE in 2012 from Anna University, Tamil Nadu, India. Currently she is a Full Time Research Scholar in the Department of Electronics and Communication Engineering at Anna University, CEG Campus, Tamil Nadu, India. Her research interests include Mobile Ad hoc Networks, Wireless Body Sensor Networks and Internet of Things.

Neeraj Kumar received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra, India. He is now a Full Professor in the Department of CSE, Thapar University, Patiala, Punjab (India). His research is focused on mobile computing, parallel/distributed computing, multiagent systems, service oriented computing, routing and security issues in mobile ad hoc, sensor and mesh networks. He has more than 100 technical research papers in leading journals and conferences. His research is supported from DST, TCS and UGC. He has guided many students leading to M.E. and Ph.D.

Satish Narayana Srirama is an Associate Professor at School of Computer and Information Sciences, University of Hyderabad, India. He is also a Visiting Professor and the honorary head of the Mobile & Cloud Lab at the Institute of Computer Science, University of Tartu, Estonia, which he led as a Research Professor until June 2020. His current research focuses on cloud computing, mobile cloud, IoT, Fog computing, migrating scientific computing and large scale data analytics to the cloud. He is an Editor of Wiley Software: Practice and Experience journal.