This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

Service Deployment Strategy for Predictive Analysis of FinTech IoT Applications in Edge Networks

Ambigavathi Munusamy, *Member, IEEE*, Mainak Adhikari, *Member, IEEE*, Venki Balasubramanian, *Member IEEE*, Mohammad Ayoub Khan, *Member, IEEE*, Varun G Menon, *Senior Member, IEEE*, Danda Rawat, *Senior Member, IEEE* and Satish Narayana Srirama, *Senior Member, IEEE*

***Abstract*—The seamless integration of sensors and smart communication technologies has led to the development of various supporting systems for Financial Technology (FinTech). The emergence of the Next-Generation Internet of Things (Nx-IoT) for FinTech applications enhances the customer satisfaction ratio. The main research challenge for FinTech applications is to analyse the incoming tasks at the edge of the networks with minimum delay and power consumption while increasing the prediction accuracy. Motivated by the above-mentioned challenge, in this paper, we develop a ranked-based service deployment strategy and an Artificial Intelligence technique for financial data analysis at edge networks. Initially, a risk-based task classification strategy has been developed for classifying the incoming financial tasks and providing the importance to the risk-based task for meeting users' satisfaction ratio. Besides that, an efficient service deployment strategy is developed using $Hall's$ theorem to assign the ranked-based financial data to the suitable edge or cloud servers with minimum delay and power consumption. Finally, the standard support vector machines (SVM) algorithm is used at edge networks for analysing the financial data with higher accuracy. The experimental results demonstrate the effectiveness of the proposed strategy and SVM model at edge networks over the baseline algorithms and classification models, respectively.**

***Index Terms*—IoT; FinTech applications; Task classification; Service deployment; Support Vector Machines; Edge networks.**

I. INTRODUCTION

The Internet of Things (IoT) is a promising and emerging technology in the Industrial domain that connects an enormous amount of smart devices including sensors and actuators to the network [1]. The smart devices and advanced sensors collect the environmental parameters and transfer the data to remote computing devices for analysis, and take appropriate action [2]. In recent times, IoT-enabled technology has been applied in many real-time applications including smart transportation, smart industry, smart grid, smart city, etc., in which smart Financial Technology (FinTech) application has received more attention by leveraging IoT technology [3], [4]. The emerging phenomenon of next-generation IoT (Nx-IoT) for FinTech application is going to reveal one of the most significant moves towards smart worldwide economic diaspora. Using a smart FinTech framework, the Banks and financial institutions can provide quality services to the customers using personalized virtual supervision by optimising the financial services with advanced Artificial Intelligence (AI) technology [5]. In such a scenario, the computations and communications become more vulnerable for analysing the large volume of financial data at remote computing devices by meeting various Quality-of-Service (QoS) parameters. [6]–[8].

Nowadays, FinTech applications such as various Banking services, i.e. ATMs, Bank APPs, etc. are relying on Nx-IoT to interface with their customers and require reliable remote computing services for analysing large-scale financial data. In the past decades, centralized cloud servers provided a plethora of resources for analysing financial data with advanced AI technologies. However, the major bottleneck faced by the cloud infrastructure is their limited scalability and centralized architecture that increases the latency and drops the overall performance of FinTech applications [9]. The advancement of a new paradigm in the industrial environment such as edge computing plays an important role in FinTech applications by bringing the resources closer to the customers and provides low latency and energy usage as compared to the centralized cloud servers [10]. In practice, Banks use the local edge devices for satisfying personalized customer experience by processing the latency-sensitive applications locally with minimum delay [11], [12]. For example, virtual tellers or facial recognition technology was difficult to analyze in the centralized cloud servers due to the high latency and low transmission speed. In recent times, due to the edge-centric framework of FinTech applications, the customers' faces can be recognized instantly, receive relevant loan offer information, delivering information to the Banking staff, etc. with minimum delay.

A. Motivation

The main focus of the Bank and FinTech institutes is to process or analyze the financial data, mainly the latency-sensitive applications, namely virtual tellers or facial recognition technology at the edge of the network with minimum

Corresponding author: Satish Narayana Srirama.

A. Munusamy is with the Department of ECE, CEG Campus, Anna University, Chennai, India. (e-mail: ambigaindhu8@gmail.com).

M. Adhikari is with the Mobile & Cloud Lab, Institute of Computer Science, University of Tartu, Estonia (e-mail: mainak.ism@gmail.com).

V. Balasubramaniam is with the School of Science, Engineering and Information Technology, Federation University, Mount Helen VIC 3350, Australia (e-mail: v.balasubramanian@federation.edu.au).

M. A. Khan is with the College of Computing and Information Technology, University of Bisha, Saudi Arabia (e-mail: ayoub.khan@ieee.org).

V. G. Menon is with the SCMS School of Engineering and Technology, India (e-mail: varunmenon@scmsgroup.org).

D Rawat is with the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA (e-mail: db.rawat@ieee.org).

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 Internet of Things Journal. 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/JIOT.2021.3078148

delay. Besides that, due to the limited resource capacity of the local edge devices, the computation-intensive financial data need to be transferred to the centralized cloud servers for analysis. Thus, two main research questions for developing an efficient edge-centric framework for FinTech applications are- *1. how to classify the mixed financial data as per their importance, so that the latency-sensitive risk-based data are analysed at local edge devices? 2. how to provide the services for the classified data, so that the risk-based data are analyzed at local edge devices with minimum delay and energy usage?* Besides that, *3. finding a suitable classification model to analyze the financial data at the edge of the networks with the minimum set of data with higher accuracy is another important research challenge?* Nowadays, FinTech applications generate a huge volume of financial data at an exponential rate from the next-generation IoT devices, customers, Banks, and insurance sectors, etc. One of the major critical tasks in financial industries is to predict the credit risks of legal clients and detect and prevent fraudulent activities. The traditional risk assessment techniques used in the financial sectors are costly and time-consuming to process labor-intensive tasks and cannot handle the large volume of financial data.

B. Related Work

To tackle the aforementioned issues, several research works have focused on service deployment and resource provisioning in edge networks. To provide a network service across multiple domains, a chain-based network deployment strategy has been introduced in [13]. This strategy aims to reduce the cost and latency using the virtual network function. Similarly, in [14], a collaborative service deployment and assignment scheme has been proposed in edge networks. The integrated resource provisioning model has been designed to seamlessly provide services across the edge servers and cloud server in [15]. This method effectively considered various service demands from the users and dynamically schedules the incoming tasks to achieve efficient service deployment. In [16], an energy-efficient task allocation scheme for a mobile cloud system has been designed to minimize the power consumption of the computing servers while meeting the deadline.

Authors in [6] have developed a 6G-aware fog federation model to effectively schedule the resources in fog networks using a non-cooperative Stackelberg game theory with minimum service costs while maximizing the users' satisfaction ratio. To balance the power consumption and delay tradeoff between the mobile devices and computing servers, three queuing models have been applied in [17] that find the optimal uploading probability and transmit power for each server. The energy-efficient multitasking strategy has been proposed at multi-access mobile edge computing networks in [18] that minimized the total power consumption of the computing devices with a suitable scheduling order. Further, a joint optimization problem has been formulated in [19] to minimize the power consumption and delay of the incoming tasks using a weighted function.

In [20], authors have evaluated the series of Machine Learning (ML) models over credit card fraud detection datasets to find the best classification model concerning the type of frauds. The various ML classification models have been investigated over different financial datasets in [21] to resolve the issue of the data imbalance. Authors in [22] have studied the ML classification models in various financial institutions include credit rating, bankruptcy prediction, and fraud detection. In [23], authors have developed an automated insurance prediction system to reduce human interaction, secure the insurance activities, notify risky customers, and detect fraudulent claims. In [24], the authors have revealed the classification models ineffectively only when the financial data are highly imbalanced. Authors in [25] have considered the Random Forest algorithm to classify the churned customers using two datasets with higher prediction accuracy. Therefore, the critical challenge for analysing the FinTech applications at the edge level is to distribute the incoming tasks on the local edge devices or centralized cloud servers as per their importance through an efficient service deployment and prediction strategy with higher accuracy. Considering these challenges as a motivation, we design an efficient ranked-based service deployment strategy for predictive analysis of FinTech applications with SVM algorithm at edge networks for achieving higher prediction accuracy and minimum delay.

C. Contributions

Our main contributions of the Ranked-Based Service Deployment (RBSD) strategy for predictive analysis of the FinTech applications at edge networks are summarized as follows.

- Design a new ranked-based strategy for classifying the incoming financial tasks at the edge of the network such as risk-based and nonrisk-based tasks as per their priority. Such a classification aids for analysing the risk-based financial data at the distributed edge devices with minimum delay and higher accuracy.
- Devise a service deployment strategy with a perfect matching theorem in Graph theory, i.e., $Hall's$ theorem for distributing the ranked-based tasks to the remote computing devices. $Hall's$ theorem is used to find a perfect matching between the ranked-based tasks and the active set of computing devices for minimizing power consumption at networks.
- Introduce a standard SVM classification model for analysing the ranked-based tasks at the edge networks using a real dataset with higher accuracy and precision. The SVM model uses a small-scale dataset for risk prediction at the edge level, whereas a large-scale dataset is used for prediction at the cloud level with minimum error.
- Extensive simulation results demonstrate the effectiveness of the proposed RBSD strategy at edge networks for FinTech applications in terms of average delay and power consumption. Besides that, the standard SVM technique demonstrates the effectiveness of analysing financial tasks with real datasets at edge networks over standard classification models in terms of accuracy and precision.

The remaining sections of the paper are organized as follows. Section II highlights the system model followed by the prob-

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

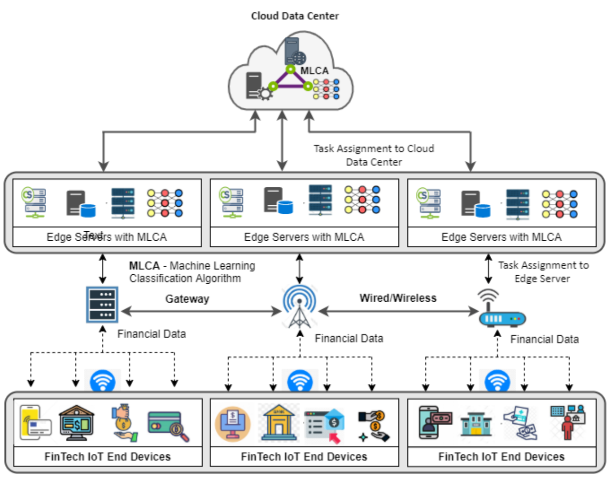


Fig. 1. Edge Framework for Predictive Analysis of FinTech Applications

lem formulation of edge networks for FinTech applications. The proposed service deployment strategy for predictive analysis of FinTech applications is discussed in Section III. The empirical evaluations of the proposed methodology over the existing ones are elaborated in Section IV. Finally, Section V concludes the work and highlights future directions.

II. System Model and Problem Formulation

This section describes the proposed edge-centric service deployment framework for predictive analysis of FinTech IoT applications followed by the problem formulation.

A. System Model

The proposed edge-centric service deployment framework for FinTech applications is depicted in Fig. 1. This network is constructed with a set of edge servers $\mathcal{S} = \{S_1, S_2, S_3, \ldots, S_d\}$ and finite number of remote cloud servers $\mathcal{R} = \{R_1, R_2, R_3, \ldots, R_o\}$. The computing servers are highly capable to process the large amount of financial data, collected from the set of FinTech IoT devices $\mathcal{D} = \{D_1, D_2, D_3, \ldots, D_f\}$. These devices seamlessly generate the financial tasks $\mathcal{T} = \{T_1, T_2, \ldots, T_f\}$ with various degrees of importance including risk-based (R) and non-risk (NR) financial tasks, i.e., $(\mathcal{T} \in (R \cup NR))$. Further, the financial tasks are processed either locally or transmitted to the remote computing servers for further predictions through a set of gateway devices $\mathcal{G}$, denoted as $\mathcal{G} = \{G_1, G_2, \ldots, G_d\}$. The local gateway devices are responsible for task ranking and service deployment decisions over the received data. Due to inefficient processing capacity (τ_{end}^{CPU}) and power consumption (P_{end}^{CPU}), the efficiency of these two metrics for IoT devices are always less than the edge and cloud servers. Likewise, the CPU capacity and power consumption of an edge device (τ_{edge}^{CPU}, P_{edge}^{CPU}) should be less than the remote cloud server (τ_{cloud}^{CPU}, P_{cloud}^{CPU}).

In this network, the set of local edge devices and remote cloud servers are represented as $\mathcal{SR} = (\mathcal{S} \cup \mathcal{R})$. The edge-centric network cogitates that the ith risk-based financial task, referred to as T_i^R, is assigned to the local edge devices. Similarly, the nonrisk-based financial task, referred to as T_i^{NR}, is deployed to the remote cloud servers. The input and output size of each task are denoted as T_i^{in} and T_i^{out}, respectively. For instance, the task assignment probability $X(i,j)$ is stated that the assignment of a financial task i to the jth computing device, $\forall j \in (\mathcal{D} \cup \mathcal{SR})$. In this scenario, the value of task assignment probability $X(i,j)$ is 1, if ith task is assigned to the jth computing device, where $\forall j \in (\mathcal{D} \cup \mathcal{SR})$, otherwise $X(i,j)$ is 0. Therefore, this work mainly focuses to investigate the impact of both power consumption and delay of financial tasks in three different operational modes including financial task uploading, downloading, and processing.

B. Local Execution Mode

The local FinTech IoT devices have limited power, and CPU frequency (τ_i^{CPU}). For instance, the ith task can process locally when the required CPU frequency of the incoming task is less than or equal to the available CPU capacity of the local IoT device. The total time required to execute the ith task in jth IoT device is expressed as follows.

$$P_{R_{ij}} = X(i,j) \times \frac{T_i^{in}}{\tau_i^{CPU}} : \forall i \in \mathcal{T}, j \in \mathcal{D} \tag{1}$$

Processing the task at local IoT devices depends on CPU frequency instead of the communication delay. Let us consider that the required power to process a 1-bit task at jth IoT device is defined as P_j^{CPU}. Thus, the overall power consumed by the task i at jth IoT device is computed as follows.

$$P_{ij}^{proc} = X(i,j) \times \frac{T_i^{in}}{\tau_i^{CPU}} \times P_j^{CPU} : \forall i \in \mathcal{T}, j \in \mathcal{D} \tag{2}$$

C. Remote Execution Mode

Due to the limited processing and storage capacity of the FinTech IoT devices, the large volume of financial tasks $\mathcal{T}$ is directly uploaded to the remote edge or cloud servers for further predictions. Therefore, the total time required to process the financial tasks at remote computing devices depends on the uploading, downloading, and processing time. For instance, if a task i is assigned to the jth computing device, i.e., $\forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R})$, then, the transmission rate of the ith task to jth computing device is defined as $\gamma_{ij}^{up} = \mathcal{W}_{ij}^{in} \log(1 + P_j^{up} \times \frac{\delta_i^{power}}{\alpha_i^2})$. Here, $\mathcal{W}_{ij}^{in}$ indicates the channel utilization factor between the ith IoT device and jth computing device. α_i^2 and P_j^{UP} represent the Additive White Gaussian Noise of the local IoT device and the transmission power to offload the task to the jth computing device, respectively. Thus, the total transmission time required to upload the task to the remote computing device can be formulated as follows.

$$T_{ij}^{up} = X(i,j) \times \frac{T_i^{in}}{\gamma_{ij}^{up}} : \forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R}) \tag{3}$$

Consequently, the uploading power consumption (P_{ij}^{up}) of ith financial task to jth remote computing device is expressed as follows.

$$P_{ij}^{up} = T_{ij}^{up} \times P_j^{up} : \forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R}) \tag{4}$$

The total time required to execute a task i $\forall i \in (\mathcal{T}_i^R, \mathcal{T}_i^{NR})$ on the jth remote computing device $\forall j \in (\mathcal{S}, \mathcal{R})$ is defined as follows.

$$P_{ij} = \begin{cases} \mu_{kj}^R \times X(i,j) \times \frac{T_i^{in}}{T_i^{CPU}} & \text{if}, T_i \in \mathcal{T}_i^R, j \in \mathcal{S} \\ \mu_{kj}^{NR} \times X(i,j) \times \frac{T_i^{in}}{T_i^{CPU}} & \text{if}, T_i \in \mathcal{T}_i^{NR}, j \in \mathcal{S} \\ (1-\mu_{kj}^R) \times X(i,j) \times \frac{T_i^{in}}{T_i^{CPU}} & \text{if}, T_i \in \mathcal{T}_i^R, j \in \mathcal{R} \\ (1-\mu_{kj}^{NR}) \times X(i,j) \times \frac{T_i^{in}}{T_i^{CPU}} & \text{if}, T_i \in \mathcal{T}_i^{NR}, j \in \mathcal{R} \end{cases} \tag{5}$$

The arrival rate of the financial task on the remote edge and cloud servers are represented as λ_i^{edge} and λ_i^{cloud}, respectively. Further, the waiting time l_{ij} of ith task before assigning to the jth computing device is defined as follows.

$$l_{ij} = \lambda_i^{edge} \frac{T_i^{in2}}{\tau_i^{CPU}} (\tau_i^{CPU} - \lambda_i^{edge} \times \mathcal{T}_i^{in}) : j \in (\mathcal{S}, \mathcal{R}) \tag{6}$$

The total execution delay of the ith task on jth computing device at time t is expressed as $l(t) = \sum_{i=1}^{q} l_{ij}$. Let P_j^{CPU} represents the processing power to process 1-bit data at remote computing device. Thus, the total consumed power to process ith task on the jth remote computing device is measured as follows.

$$P_{ij}^{proc} = P_{ij} \times P_j^{CPU} : \forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R}) \tag{7}$$

Let σ_j^{power} represents the channel power gain of jth computing device. W_{ij}^{out} and δ_j^{power} denote the channel utilization between remote jth computing device to ith IoT device and required transmission power of jth remote computing device. Thus, the power consumption of ith task during the downloading process (γ_{ji}^{down}) is defined as follows.

$$\gamma_{ji}^{down} = \mathcal{W}_{ij}^{out} \log(1 + P_j^{down} \times \frac{\delta_j^{power}}{\alpha_j^2}) : \forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R}) \tag{8}$$

Where α_j^2 denotes the Gaussian noise ratio on the jth remote computing device. The downloading time T_{ji}^{down} from jth computing device to the ith IoT device is defined as follows.

$$T_{ij}^{down} = X(j,i) \times \frac{T_i^{out}}{\gamma_{ji}^{down}} : \forall i \in \mathcal{T}, j \in (\mathcal{S}, \mathcal{R}) \tag{9}$$

Subsequently, the downloading power consumption of ith financial task is computed as follows.

$$P_{ij}^{down} = X(i,j) \times T_i^{out} \times \frac{P_j^{down}}{W_{ij}^{out} \times \log(1 + P_j^{down} \times \frac{\delta_j^{power}}{\alpha_j^2})} \tag{10}$$

The total power consumption of a financial task i during computation at jth remote computing device is measured as follows.

$$P_{ij}^{total} = (P_{ij}^{up} + P_{ij}^{proc} + P_{ji}^{down}) \tag{11}$$

Therefore, the total power consumption $(P_{ij}^{total}(t))$ of a financial task i during uploading, processing, and downloading to the jth computing device at time t is expressed as follows.

$$P_{ij}^{total}(t) = (P_{ij}^{up}(t) + P_{ij}^{proc}(t) + P_{ij}^{down}(t)) \tag{12}$$

D. Problem Formulation

The main goal of this work is to minimize the power consumption and delay of the financial tasks in three different modes such as uploading, processing and downloading phase. If a financial task is assigned to the local IoT device $\mathcal{D}$, then the total power consumed (i.e. P_{ij}^{total}) by the ith financial task is equal to the processing power (P_{ij}^{proc}) in the local IoT device. However, if the i is assied to the local edge or remote cloud server j, then the total power consumption (P_{ij}^{total}) by the task i depends on the uploading power P_{ij}^{up}, downloading power P_{ij}^{proc} and processing power P_{ij}^{proc}, i.e., $P_{ij}^{total} = (P_{ij}^{up} + P_{ij}^{proc} + P_{ij}^{down})$. The objective function of the work with necessary constraints are formulated as follows.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} P_{ij}^{total}(t) && (13a) \\ \text{subject to} \quad & P_{ij}^{total}(t) \leq \eta_j^{max}, j \in (\mathcal{S} \cup \mathcal{R}) \,, && (13b) \\ & l_{ij}(t) \leq l_j^{max}, j \in (\mathcal{S} \cup \mathcal{R}); , && (13c) \\ & \tau_i^{CPU}(t) \leq \tau_i^{max}, j \in (\mathcal{S} \cup \mathcal{R}); , && (13d) \\ & \sum_{i=1}^{(|T|)} \sum_{j=1}^{|SR|} X(i,j) \leq |\mathcal{S} \cup \mathcal{R}|; , && (13e) \\ & \sum_{j=1}^{(|T|)} X(i,j) = 1; && (13f) \end{aligned}$$

From the above problem formulation, constraints (13b) and (13c) state the total power consumption and delay of a financial task i should be less than or equal to the maximum power consumption η_j^{max} and delay l_{ij}^{max}, respectively. According to the constraint (13d), the required CPU frequency of ith financial task should be less than or equal to the selected computing device j. (13e) represents the active number of remote computing devices in the network. Finally, constraints (13f) states that each financial task should be assigned at most one computing device at time t.

III. Ranked-based Service Deployment Strategy

This section presents an effective Ranked-Based Service Deployment (RBSD) strategy for FinTech IoT applications at edge networks. Initially, the incoming tasks from various FinTech IoT devices are ranked according to their importance and priorities. Then, the ranked financial tasks are assigned to the suitable computing devices for further analysis.

A. Ranked-based Task Classification

In the ranked-based classification model, the incoming financial tasks from the IoT devices are classified based on their degrees of importance and service requirements. Subsequently,

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

the ranked tasks are placed into the buffers of a local gateway device for making further decisions. To get instant response from the local edge devices, the rank index (η) factor is introduced to identify the importance of the financial tasks and locate them according to the non-decreasing order. We consider η is a priority threshold value to classify the severity of incoming financial tasks. With the help of (η) value, the financial tasks are effectively categorized into two types, risk-based (R) and non-risk-based (NR) tasks, represented as T_i^R and T_i^{NR}, respectively. The value 0 and 1 indicates the types of the incoming task, i.e., 0 represents risk-based task T_i^R and 1 represents the non-risk-based task T_i^{NR}.

In this way, the proposed RBSD strategy satisfies the following two constraints: (i) a task T_i is called a risk-based task if $\eta(T_i) \geq 0.5$ or (ii) a non-risk-based task if $\eta(T_i) < 0.5$. Based on the ranking orders, the risk-based tasks are placed into the risk-based buffer $\omega_i^R(t)$, if $T_i \in T_i^R$ or to the non-risk based buffer $\omega_i^{NR}(t)$, if $T_i \in T_i^{NR}$. The systematic workflow of the ranked-based classification model is illustrated in Fig. 2. In this model, the arrival rate of financial tasks are symbolically represented using a Poisson process with the density function $f(t) = \lambda_i^e - \lambda_i^t$. The parameters λ_i and ϕ_{jk} denote the financial task arrival rate and the task uploading probability from j^{th} IoT device to the k^{th} gateway device, respectively. The offloading decisions at the k^{th} gateway device is defined as $\lambda_{jk}^{rem} = \phi_{jk} \times \lambda_i, \forall j \in D$. Thus, the arrival rate of the ith task for processing locally on the jth IoT device is formulated as follows.

$$\lambda_{ij}^{local} = (1 - \phi_{jk}) \times \lambda_i \tag{14}$$

The arrival rate of the set of financial tasks (σ_{jk}) under a risk-based buffer of kth local gateway device is defined as follows.

$$\lambda_{jk}^R = \sigma_{jk} \times \lambda_{jk}^{rem} \tag{15}$$

Similarly, the remaining set of financial tasks that arrive under a non-risk-based buffer of kth gateway device is expressed as follows.

$$\lambda_{jk}^{NR} = (1 - \sigma_{jk}) \times \lambda_{jk}^{rem} \tag{16}$$

The probabilities of assigning risk-based and non-risk-based financial tasks to the jth computing device are expressed as μ_{kj}^R and μ_{kj}^{NR}, respectively. Thus, the arrival rate of the ith task from the kth gateway device to the jth edge device, $\forall j \in \mathcal{S}$ is expressed as follows.

$$\lambda_i^{edge} = \mu_{kj}^R \times \lambda_{jk}^R + \mu_{kj}^{NR} \times \lambda_{jk}^{NR} \tag{17}$$

$$= \mu_{kj}^R \times \sigma_{jk} \times \lambda_{jk}^{rem} + \mu_{kj}^{NR} \times (1 - \sigma_{jk}) \times \lambda_{jk}^{rem} \tag{18}$$

Similarly, the task arrival rate of the ith task to the jth remote cloud server, $\forall j \in \mathcal{R}$ from kth gateway device is represented as follows.

$$\lambda_i^{cloud} = (1 - \mu_{kj}^R) \times \lambda_{jk}^R + (1 - \mu_{kj}^{NR}) \times \lambda_{jk}^{NR} \tag{19}$$

$$= (1 - \mu_{kj}^R) \times \sigma_{jk} \times \lambda_{jk}^{rem} + (1 - \mu_{kj}^{NR}) \times (1 - \sigma_{jk}) \times \lambda_{jk}^{rem} \tag{20}$$

The total arrival rate of risk-based $(i.e. \sum_{(j \in I)} \omega_i^R(t) \lambda_i^R)$ and non-risk-based financial tasks $(i.e. \sum_{(j \in J)} \omega_j^{NR}(t) \lambda_j^{NR})$, and service rate (μ_{ij}) at the local buffer of the gateway device

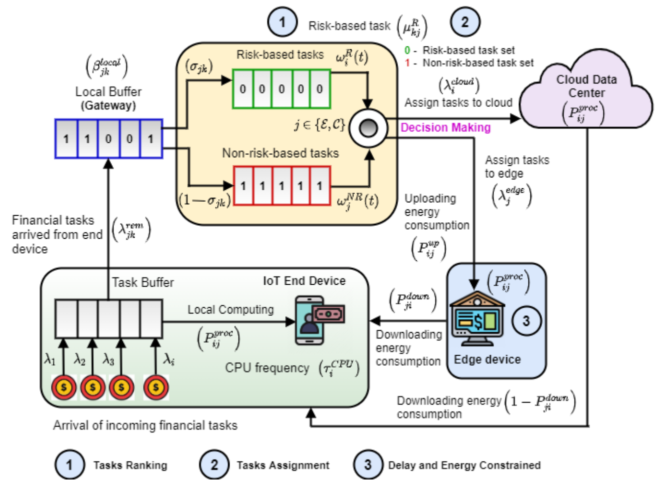


Fig. 2. Workflow of Ranked-based Task Classification

do not create much impact on financial tasks uploading and downloading decisions at time t. Further, the power-efficient task uploading decisions can be achieved using the following function.

$$\beta_{T_i}^{out}(t) = minimize \sum_{j \in S,R} \frac{(T_i^{in} \times P_j^{up})}{W_{ij}^{in}} + \frac{P_j^{CPU} \times T_i^{in}}{\tau_i^{CPU}} + \frac{(T_i^{out} \times P_j^{down})}{W_{ji}^{out}} + \sum_i \in I \omega_i^R(t) \times \mu_i(t) - \sum_j \in J \omega_i^N R(t) \times \mu_j(t)$$

Based on the above formulation it is proved that the ranked-based classification model satisfies the power consumption and delay constraints (from (13a)-(13h)) in the edge networks. Next, the classified tasks are assigned to the suitable remote computing devices for further analysis using a perfect matching algorithm.

B. Service Deployment Strategy with Perfect Matching

This section discusses the proposed service deployment strategy with a perfect matching theorem for assigning the ranked-based tasks of the FinTech IoT applications to suitable remote computing devices for further prediction while minimizing the power consumption and delay. To map the ranked financial tasks with the active set of computing servers, a well-known perfect matching theorem in Graph theory, namely $Hall's$ theorem is considered in edge networks. Mathematically, the perfect mapping function is expressed as P: $T_i \to \mathcal{C}$ between the ranked task set $\mathcal{T}$ and the computing devices c using a link weight function $F : Q \to R^+ \cup \infty$. In this model, the weight function F_{ij} between the ranked task T_i and computing server C_j always depends on the total power consumption (P_{ij}^{total}). Constantly, the gateway device produces a new set of ranked financial tasks concerning the availability of the active set of computing devices.

The $Hall's$ perfect matching theorem for FinTech IoT applications at the local gateway device is depicted in Fig. 3. The decision making graph is constructed using hall's

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

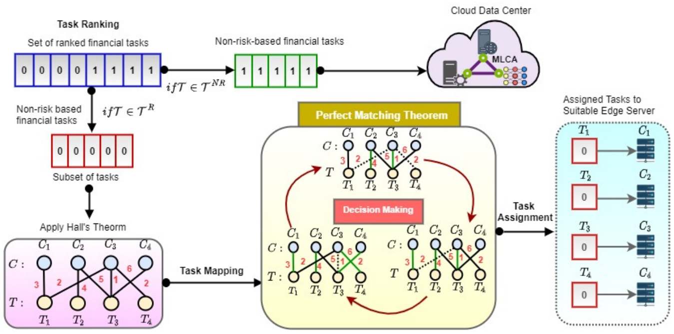


Fig. 3. Service Deployment with Perfect Matching Theorem

complete bipartite graph G (M, N), which consists of set of vertices and dummy edges with a positive link weight ∞ in the form of power consumption. In this graph, the ranked-based task assignment starts with a dual matching solution such that $D_j = 0, \forall j \in C$ and $D_i = \text{MIN}(F_{ij}) : N_{ij} \in K(i), \forall i \in \mathcal{T}$. This condition states that the tight edges N' has at least one perfect matching in subgraph G', defined as $F_{ij} = D_i + D_j$. If there is no matching N', then the dual value of corresponding $Hall's$ financial tasks set is modified by adding a constant value K to T_i and subtracting the value K from C_j, referred as $D_i = D_i + K$ and $D_j = D_j - K$, respectively.

In a given task assignment graph $G = (M, N)$ with bi-partition (T, C), where $M = (T \cup C)$ and a perfect matching function $P : T \rightarrow C$ such that G assigns set of all ranked-based tasks T in each time frame if and only if $|X| \geq |B(X)|$, where $X \subseteq T$ and $B(X) = (h \in C | C = (S \cup R), (T, C) \in Q,$ and $\forall T \in X)$. Let us consider that $X = (T_1, T_2, T_3, T_4), X \subseteq T$, then $B(X) = B(T_1) \cup B(T_2) \cup B(T_3) \cup B(T_4) = (C_1, C_2, C_3, C_4)$. Hence, the $Hall's$ condition is satisfied with $|X| \leq |B(X)|$, where X is the set of all possible combination of tasks in the financial task set T. The condition $|X| \leq |B(X)|$ denotes that all the subsets of T are mapped when there exists a mapping from financial tasks to the corresponding computing devices. Therefore, $Hall's$ condition is satisfied and the graph $\mathcal{G}$ has saturated matching from task $\mathcal{T}$ to the edge device $\mathcal{S}$.

As shown in Fig. 3, the financial task T_2 is perfectly matched with C_2, and T_3 is matched with C_3. However, for task T_3, there is no tight matching in the set C, which indicates that among the tight edges in N' both the task T_2 and T_3 have a perfect matching. Furthermore, for a task T_1, there is a $Hall's$ set, i.e., $T_1 \cup T_3$. Accordingly, the ranked-based task assignment graph needs to be modified using the dual value, so the sub-graph G' extends with untight edges until a perfect match is found. For this purpose, the sub-graph G' is modified by adding the value of K in the financial task set $\mathcal{T}$ and removing K from the set C. Based on the perfect matching theorem, each ranked task T_i is assigned or mapped to at most one remote computing device C_j, which ensures the financial task assignment constraint (13d). Finally, all the ranked-based financial tasks are assigned to the suitable edge devices based on their perfect matching order. Furthermore, the proposed service deployment strategy decreases the computation and communication overhead of the network by assigning the nonrisk-based tasks to the remote cloud servers while finding a maximum matching between the ranked-based tasks and local edge devices. The systematic procedures of the ranked-based service deployment strategy are depicted in Algorithm 1.

Algorithm 1: Ranked-based Service Deployment

INPUT: Rank index factor: η, Incoming tasks: $\mathcal{T}_i$, Set of computing servers:$C \leftarrow (S \cup R)$, Risk based buffer: ω_i^R
OUTPUT: Classify and assign the incoming tasks to the suitable computing servers using η
for i:1 to n **do**
  Assign rank index factor η to the incoming tasks
  if A task $\omega_i^{NR} \leftarrow T_i^{NR} \leq \eta$ **then**
    Assign a $T_i^N R$ to non-risk-based buffer ω_i^{NR}
  end if
  if A task $\omega_i^R \leftarrow T_i^R \geq \eta$ **then**
    Assign a T_i^R to risk-based buffer ω_i^R
  end if
  Assign ranked tasks to the suitable C using Perfect matching
  if $|X| \leq |B(X)|)$ **then**
    Graph has a saturated matching of $\mathcal{T}_i$
  end if
  if $|X| \geq |B(X)|)$ **then**
    Find matching from N' Where $(F_{ij} = D_i - D_j)$;
    Modify $D_i = D_i + k, \forall i \in \mathcal{T}$
    Modify $D_j = D_j + k, \forall i \in \mathcal{T}$
    Update the value of tight edges N' based the matching function F
  end if
  Assign risk based financial tasks T_i^R to the edge server S_j
end for
for All ranked tasks $T_{ij} \in \omega_j^{NR}$ **do**
  Assign non-risk based financial tasks T_i^{NR} to the remote cloud server R_j
end for
Return a perfect mapping function

C. Predictive Analysis at Edge Networks

The huge volume of data, collected from various FinTech applications through Nx-IoT demands instant decisions and service requirements from the banking or financial sectors. However, most of the financial industries still process customer-related information using traditional or manual screening and analytic tools. Due to the digital transformation of financial data using Nx-IoT, the instant prediction and identification of cybercriminals and frauds are challenging tasks in financial industries. Thus, the financial industries must require an intelligent predictive and analytical model to deal with them. Besides that, the transmission of mixed types of financial data from FinTech IoT devices to the remote cloud server increases the delay and power consumption of the customer service requirements. In such cases, instigating predictive analytic models at the local edge devices helps to

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

analyze, and identify the huge volume of risk-based financial data and provide instant services closer to the customers with minimum delays and errors.

Based on these perceptions, various machine learning classification models such as Logistic Regression (LR), Decision Trees (DT), Support Vector Machine (SVM), and Random Forest (RF) have been studied and validated using different real-time financial datasets. However, the proposed edge-centric predictive analysis considers the SVM model as the baseline model to effectively analyze and estimate the banking crises with higher accuracy over other classification models. The reason behind selecting the SVM classification model is that the SVM model is capable to handle high-dimensional financial data and improves significant accuracy with less computation power [26], [27]. Further, to estimate the decision function with minimum error, the SVM model uses a linear model with a non-linear boundaries class based on support vectors. In the proposed strategy, with the help of the SVM classification model, the ranked-based tasks are analyzed and predicted at the resource-constraints edge devices to get an instant response and enhance the service requirements of the customers. Similarly, the non-risk-based tasks are analysed at the remote cloud server for future predictions.

IV. Empirical Evaluation

This section briefly discusses the empirical evaluation of the proposed ranked-based classification model and service deployment strategy in edge networks. The proposed edge-centric FinTech framework is quantified and validated concerning average delay and power consumption. To verify the ability of the edge-centric framework, we compare the proposed framework with two baseline schemes such as CoISDA [14] and OSP [15]. Further, the predictive classification model, i.e., the SVM technique is applied over the financial tasks at both edge and cloud server to prove the superiority of the proposed framework and the results are compared with the state-of-the-art models including LR [28], DT [29], and RF, [30]. Further, different validation metrics including accuracy, precision, recall, and F1 score are considered to find the effectiveness of the SVM classification models for financial risk predictions.

A. Experimental Setup and Dataset

The proposed strategy has been implemented on Intel Core i7-8550U Quad-Core CPU with 12GB RAM using Ubuntu LTS operating system. The simulation test parameters are summarized in Table I. The edge network consists of 500 FinTech IoT devices that generate 500 tasks/sec in each timestamp. Here, the maximum data transmission rate is fixed to 2.5 Mb/s, the range of input task size is T_i^{in} is [50kb-10Mb], and the financial task arrival rate on the edge devices λ_i^{edge} is 0.125 and the remote cloud server λ_i^{cloud} is 0.25. Here, the ranked-based financial tasks are analyzed using real datasets such as credit card fraud prediction (D1)[1], credit card risk prediction (D2)[2], Customer Churn Prediction (D3)[3], and Insurance Claim Prediction (D4)[4]. Table II contains the summary of FinTech datasets and their properties for edge-cloud level analysis.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Number of IoT devices ($\mathcal{D}$)	500
Number of Edge devices ($\mathcal{S}$)	20
Number of cloud servers ($\mathcal{R}$)	2
Number of gateway devices ($\mathcal{G}$)	2
Average number of incoming data (λ_i)	500 [tasks/sec]
Maximum channel bandwidth (W)	20 MHz
CPU frequency of IoT devices (τ_i^{CPU})	10 × 105 [cycles/sec]
CPU frequency of edge devices (τ_e^{CPU})	20 × 110 [cycles/sec]
CPU frequency of cloud servers (τ_c^{CPU})	30 × 120 [cycles/sec]
CPU processing power usage (P^{CPU})	0.5 Joules
Transmission power of IoT devices (T^I)	1 mW

B. Simulation Results

The simulation results of the proposed service deployment strategy are evaluated in two different phases such as communication and computation, respectively. In the first phase, the delay and power consumption of the incoming financial tasks have been analyzed in edge networks. Likewise, the prediction accuracy of the classification models has been tested and validated in the computation phase. The quantitative results of the proposed strategy are concisely described in the following subsections.

1) Analysis of Delay: Fig. 4 shows the impact of task assignment over the delay in edge networks. The delay of the financial task depends on the processing, uploading, and downloading time while assigning to the remote computing devices. The delay variation of the risk-based tasks is 29.6 ms, which is lower than the non-risk-based tasks (41.2 ms), as depicted in Fig. 4(a). Moreover, the rank index factor η is introduced to classify the incoming financial tasks based on the different order of severity. Fig. 4(b) presents the comparative analysis of the average delay of the proposed RBSD with the baseline schemes. From the analysis, it is noticed that the average delay of the baseline schemes, i.e., CoISDA (37.2 ms) and OSP (46.9 ms) is increased while varying task arrival rate, which is higher than the proposed RBSD strategy (19.4 ms). The main reason behind that the existing schemes do not consider any ranking model to classify the incoming financial tasks based on their importance and assign them to suitable computing devices. However, the proposed RBSD method used a ranked-based classification model and an efficient service deployment strategy for analysing the FinTech tasks at the edge of the networks, which reduces the delay. The proposed RBSD strategy has minimized the delay by 17.8% and 27.5% over CoISDA and OSP, respectively.

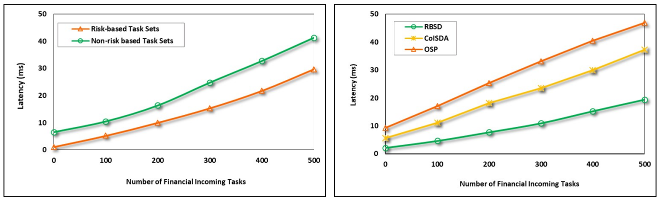
2) Analysis of Power Consumption : The impact of power consumption during the financial task assignment from the IoT devices to the remote computing devices through a local gateway is shown in Fig. 5. From Fig. 5(a), it is noted that the total required power of the IoT device (24.53 mW) is

[1]https://www.kaggle.com/nandini1999/credit-card-fraud-detection
[2]https://www.kaggle.com/kabure/predicting-credit-risk-model-pipeline
[3]https://www.kaggle.com/kmalit/bank-customer-churn-prediction
[4]https://www.kaggle.com/saikrishna223/insuranceclaimprediction

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2021.3078148

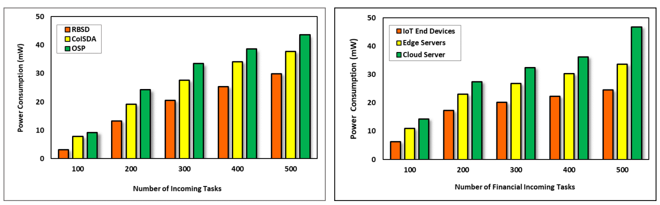
TABLE II
SUMMARY OF FINTECH DATASETS AND THEIR PROPERTIES FOR EDGE-CLOUD LEVEL ANALYSIS

Level of Analysis	Dataset(s)	No of Instances	No of Attributes	Purpose
Edge Server	D1	284808	31	Credit Card Fraud Detection
	D2	1000	20	Credit Card Risk Prediction
Cloud Server	D3	1000	14	Customer Churn Prediction
	D4	1338	8	Insurance Claim Prediction



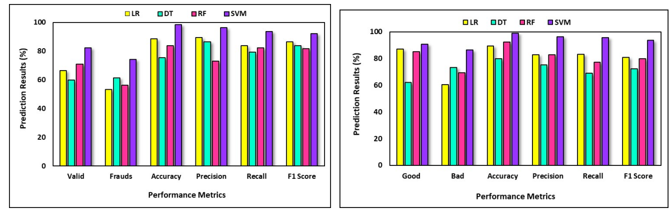
(a) (b)

Fig. 4. The impact of task assignment over delay (a) Various financial tasks (b) Comparative analysis with baseline schemes



(a) (b)

Fig. 5. The impact of task assignment over power consumption (a) Various financial tasks (b) Comparative analysis with baseline schemes



(a) (b)

Fig. 6. Edge Level Analysis using MLCAs (a) Prediction Results of D1 (b) Prediction Results of D2

less than the distributed edge devices (33.67 mW) or remote cloud servers (46.82 mW) while task analysis. However, the total power consumption of the financial tasks depends on the uploading, downloading, and processing power. Besides that, the long communication distance between the IoT devices and remote computing devices can increase the uploading and downloading time of the financial tasks, which further increases the total power consumption. The proposed RBSD strategy distributes the ranked-based tasks on the local edge devices (mainly risk-based tasks), which causes communication distance and required power consumption of the FinTech tasks. Fig. 5(b) presents the comparative analysis of average power consumption of the proposed strategy with baseline schemes. From the analysis, it is observed that the proposed strategy consumes low power (29.93 mW), while the existing schemes CoISDA and OSP consume 37.71 mW and 43.59 mW, respectively. Moreover, the quantitative analysis results show that RBSD outperforms over CoISDA and OSP schemes, which reduces the power consumption by 7.7% and 13.6%, respectively.

3) Predictive Analysis at edge level: The predictive analysis results of various classification models at the edge devices are listed in Table III. After uploading the risk-based financial tasks to the local edge devices, the standard classification models have been applied over the risk-based datasets. In the edge-based analysis, two different types of risk-based financial datasets (i.e. D1 and D2) are considered to validate and test the classification models. The prediction results of standard classification models with respect to the various performance metrics over D1 and D2 are shown in Fig. 6(a) and Fig. 6(b), respectively. From the analysis, it is evident that the SVM model provides better accuracy over the standard classification models such as LR, DT, and RF models. The SVM model achieves 98.49% accuracy while predicting the valid and fraud customers using the D1 dataset. However, the accuracy result of this model is different when considering the D2 dataset to predict the good and bad credit risk assessments. In this case, the accuracy rate of the SVM classifier achieves 99.02%, which is much higher than other standard classification models. Thus, SVM yields a minimum mean absolute error of 0.27 at edge level, which is less than the standard baseline models. This is achieved by ranking and selecting more critical features from the dataset before training the models at edge networks.

4) Predictive Analysis at cloud level: The predictive analysis results of various classification models at the cloud server are summarized in Table IV. The proposed service deployment strategy is deployed the non-risk-based financial tasks to the cloud server and the standard classification models have been applied over the non-risk-based financial datasets for further analysis. In the cloud-based analysis, two different types of non-risk-based financial datasets (i.e. D3 and D4) are considered to validate and test the classification models. The prediction results of the standards classification models over D3 and D4 are shown in Fig. 7(a) and Fig. 7(b), respectively. From the analysis, it is observed that the accuracy of the SVM classification model is greatly increased than the other standard classification models. The SVM classification model achieves 99.64% accuracy while predicting the churned and retained banking customers using the D3 dataset. However, the accuracy of the same model for the D4 dataset is improved by 99.26%, which predicts the status of claimed and unclaimed insurance of the customers, which is higher than the standard classification models. Thus, SVM yields a minimum mean absolute error of 0.36 at cloud level, which is less than the standard baseline models.

Also, it is noticed that the values of precision, recall, F1

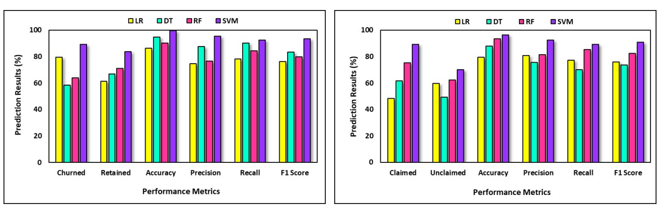
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3078148

TABLE III
PREDICTION ACCURACY OF VARIOUS CLASSIFICATION MODELS IN EDGE SERVER

Edge Level Analysis							
Dataset	**MLCA Models**	**Fraud Detection**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Valid	**Frauds**				
D1	LR	0.6645	0.5331	0.8867	0.8959	0.8362	0.8636
	DT	0.5993	0.6148	0.7532	0.8642	0.7925	0.8386
	RF	0.7076	0.5637	0.8387	0.7306	0.8254	0.8159
	SVM	0.8228	0.7406	**0.9849**	0.9639	0.9356	0.9211
Dataset	**MLCA Models**	**Risk Prediction**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Good	**Bad**				
D2	LR	0.8711	0.6039	0.8946	0.8273	0.8306	0.8093
	DT	0.6203	0.7321	0.7997	0.7527	0.6914	0.7236
	RF	0.8511	0.6939	0.9246	0.8273	0.7706	0.7993
	SVM	0.9062	0.8657	**0.9902**	0.9615	0.9558	0.9381

TABLE IV
PREDICTION ACCURACY OF VARIOUS CLASSIFICATION MODELS IN CLOUD SERVER

Cloud Level Analysis							
Dataset	**MLCA Models**	**Churn Prediction**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Churned	**Retained**				
D3	LR	0.7939	0.6133	0.8618	0.7457	0.7822	0.7635
	DT	0.5846	0.6674	0.9465	0.8769	0.9031	0.8328
	RF	0.6382	0.7092	0.9013	0.7643	0.8429	0.7976
	SVM	0.8915	0.8365	**0.9964**	0.9523	0.9241	0.9354
Dataset	**MLCA Models**	**Insurance Prediction**		**Accuracy**	**Precision**	**Recall**	**F1 Score**
		Claimed	**Unclaimed**				
D4	LR	0.4835	0.5960	0.7953	0.8067	0.7714	0.7602
	DT	0.6167	0.4928	0.8802	0.7561	0.6992	0.7353
	RF	0.7522	0.6239	0.9350	0.8134	0.8519	0.8225
	SVM	0.8908	0.7014	**0.9626**	0.9257	0.8911	0.9076



(a) (b)

Fig. 7. Cloud Level Analysis using MLCAs (a) Prediction Results of D3 (b) Prediction Results of D4

score for all the datasets (i.e., D1-D4) show higher variations in the SVM model, whereas other classification models yield fewer variations for the same set of performance metrics. Thus, the proposed ranked-based service deployment strategy along with the SVM classification model improves the risk prediction accuracy of the financial tasks and power consumption of the edge networks.

V. CONCLUSION

In this paper, we have proposed a ranked-based service deployment strategy for predictive financial data analysis at the edge networks. The main aim of this work is to analyze the risk-based financial task at the local edge devices with a standard SVM algorithm for minimizing the average delay and power consumption while maximizing the prediction accuracy. To achieve this, a ranked-based strategy has been designed for classifying the incoming financial tasks based on their priorities. Further, a service deployment strategy has been developed using a prefect matching theorem, i.e., $Hall$ theorem for assigning the classified task on the suitable remote computing devices as per their importance. Extensive simulation results exhibit the effectiveness of the proposed ranked-based service deployment strategy and the SVM algorithm at edge networks over baseline algorithms and standard classification models, respectively. The proposed strategy minimizes 17.8%-27.5% average delay and 7.7%-13.6% power consumption over the baseline algorithms. Further, the SVM algorithm achieves 98.49%, and 99.02% accuracy while analysing the data at the edge level of the network. In the future, we will enhance the proposed strategy for FinTech application by introducing various data aggregation and data fusion techniques at edge networks for minimizing network overhead and achieving higher prediction accuracy.

ACKNOWLEDGEMENT

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

REFERENCES

[1] M. Abbasi, H. Rezaei, V. G. Menon, L. Qi, and M. R. Khosravi, “Enhancing the performance of flow classification in sdn-based intelligent vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[2] S. N. Srirama, F. M. S. Dick, and M. Adhikari, “Akka framework based on the actor model for executing distributed fog computing applications,” *Future Generation Computer Systems*, vol. 117, pp. 439–452, 2021.

[3] A. Mukherjee, M. Li, P. Goswami, L. Yang, S. Garg, and M. J. Piran, “Hybrid nn-based green cognitive radio sensor networks for next-generation iot,” *Neural Computing and Applications*, pp. 1–9, Mar. 2021.

[4] S. Mostafi, F. Khan, A. Chakrabarty, D. Y. Suh, and M. J. Piran, "An algorithm for mapping a traffic domain into a complex network: A social internet of things approach," *IEEE Access*, vol. 7, pp. 40 925–40 940, Mar. 2019.

[5] 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. 1–9, 2021.

[6] 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. 1–9, 2020.

[7] M. J. Piran, Q.-V. Pham, S. R. Islam, S. Cho, B. Bae, D. Y. Suh, and Z. Han, "Multimedia communication over cognitive radio networks from QoS/QoE perspective: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 172, p. 102759, Dec. 2020.

[8] D. Thomas, R. Shankaran, Q. Z. Sheng, M. A. Orgun, M. Hitchens, M. Masud, W. Ni, S. C. Mukhopadhyay, and M. J. Piran, "Qos-aware energy management and node scheduling schemes for sensor network-based surveillance applications," *IEEE Access*, vol. 9, pp. 3065–3096, 2021.

[9] A. Mukherjee, P. Goswami, M. A. Khan, L. Manman, L. Yang, and P. Pillai, "Energy efficient resource allocation strategy in massive iot for industrial 6g applications," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5194–5201, 2021.

[10] S. R. Pokhrel, S. Verma, S. Garg, A. K. Sharma, and J. Choi, "An efficient clustering framework for massive sensor networking in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4917–4924, 2021.

[11] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9372–9382, 2020.

[12] S. Verma, S. Kaur, M. A. Khan, and P. S. Sehdev, "Toward green communication in 6g-enabled massive internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5408–5415, 2021.

[13] C. Zhang, X. Wang, Y. Zhao, A. Dong, F. Li, and M. Huang, "Cost efficient and low-latency network service chain deployment across multiple domains for sdn," *IEEE Access*, vol. 7, pp. 143 454–143 470, 2019.

[14] Y. Chen, Y. Sun, T. Feng, and S. Li, "A collaborative service deployment and application assignment method for regional edge computing enabled iot," *IEEE Access*, vol. 8, pp. 112 659–112 673, 2020.

[15] X. Cao, G. Tang, D. Guo, Y. Li, and W. Zhang, "Edge federation: Towards an integrated service provisioning model," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1116–1129, 2020.

[16] S. K. Mishra, D. Puthal, B. Sahoo, S. Sharma, Z. Xue, and A. Y. Zomaya, "Energy-efficient deployment of edge dataenters for mobile clouds in sustainable iot," *IEEE Access*, vol. 6, pp. 56 587–56 597, 2018.

[17] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2017.

[18] Y. Wu, B. Shi, L. P. Qian, F. Hou, J. Cai, and X. S. Shen, "Energy-efficient multi-task multi-access computation offloading via noma transmission for iots," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4811–4822, 2019.

[19] X. Wei, C. Tang, J. Fan, and S. Subramaniam, "Joint optimization of energy consumption and delay in cloud-to-thing continuum," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2325–2337, 2019.

[20] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, "Real-time credit card fraud detection using machine learning," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019, pp. 488–493.

[21] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201 173–201 198, 2020.

[22] A. Mashrur, W. Luo, N. A. Zaidi, and A. Robles-Kelly, "Machine learning for financial risk management: A survey," *IEEE Access*, vol. 8, pp. 203 203–203 223, 2020.

[23] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement," *IEEE Access*, vol. 8, pp. 58 546–58 558, 2020.

[24] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[25] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector," *IEEE Access*, vol. 7, pp. 60 134–60 149, 2019.

[26] S. García-Méndez, M. Fernández-Gavilanes, J. Juncal-Martínez, F. J. González-Castaño, and Ó. B. Seara, "Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus," *IEEE Access*, vol. 8, pp. 61 642–61 655, 2020.

[27] B. N. Pambudi, I. Hidayah, and S. Fauziati, "Improving money laundering detection using optimized support vector machine," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2019, pp. 273–278.

[28] Y. Li, "Credit risk prediction based on machine learning methods," in *2019 14th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2019, pp. 1011–1013.

[29] A. A. Khine and H. W. Khin, "Credit card fraud detection using online boosting with extremely fast decision tree," in *2020 IEEE Conference on Computer Applications (ICCA)*. IEEE, 2020, pp. 1–4.

[30] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2018, pp. 1–6.