Design and Analysis of Energy Consumption using Machine Learning Resource Allocation on Mobile Gadgets

Mrs. I. Varalakshmi[1], M. Krisha Santhoshi[2], D. Souvedha[3], S.Swetha[4], M. Ashmitha[5],

[1]Assistant Professor, Department of Computer Science and Engineering

[2,3,4,] UG Scholar, B.Tech, Department of Computer Science and Engineering,

[1,2,3,4,]Manakula Vinayagar Institute of Technology, Puducherry, India.

varalakshmicse@mvit.edu.in

Abstract

Energy consumption analysis and resource allocation (RA) for mobile devices need the efficient distribution of computational resources and the detailed analysis of power usage patterns among devices. Using techniques such as predictive modelling, monitoring, energy consumption patterns, and data collection are examined, enabling informed decisions based on the RA namely network bandwidth, CPU, and memory. By enhancing RA techniques based on device workload and real-time energy demands, this method focuses on enhancing energy effectiveness, extending battery lifetime, and improving overall system performance in mobile computing environment. This study introduces an innovative approach to monitoring the energy consumption of mobile devices interconnected to the Raspberry Pi via the web application interface. Particularly, the focus is on Android mobiles that are wirelessly connected to the Raspberry Pi through the WiFi network connection. This allows real-time monitoring of key energy metrics, such as overall energy consumption, CPU usage, and battery levels, which facilitates informed decision-making based on the RA. Moreover, the Raspberry Pi applies an XGBoost classifier to efficiently define allocate resources and the idle state of connected devices based on their usage patterns. The integrated system optimizes energy efficacy and improves resource utilization, thus contributing to the performance and sustainability of mobile devices. The system can intelligently allocate resources and predict device usage based on real-time energy demands through data collection and analysis, combined with machine learning techniques like XGBoost. The architecture intends to improve energy efficacy, extend battery lifetime, and improve overall system performance by enhancing RA, thus contributing to resilient and sustainable mobile computing environments.

Keywords: Energy Consumption; Raspberry Pi; Machine Learning; Resource Allocation; XGBoost; Internet of Things

1. Introduction

Recent developments in mobile technology have aided novel smart-city classes and applications of 5G networks like smart households and real condition examines [1]. This kind of application contains assorted desires like high data rates, lower potentials, substantial quantities of calculating and storage resources, and acquiring Internet of Things (IoT) devices [2]. The rate of electrical bills is increasing, which causes the users with low profits to struggle to create these expenses [3]. The varying weather situations and the assumption of novel domestic devices are dual main causes of increasing electricity bills. The IoT is one of the novel and effective technologies which authorizes consumers with intellect [4]. The IoT has massive possibility for usage in an extensive variety of states such as agriculture, healthcare, and industry, among others. Many researchers have performed in the energy sector [5]. Recently, study has been directed on the features of observing power consumption utilizing IoT. Due to the reduction of IoT actuators and sensors, IoT methods can be inserted in a diversity of modules that relate to networks of energy like domestic electricity meters, home electronic applications, power value service wires, etc [6]. The deliberated IoT method permits remote observation of power utilization in households.

Massive amounts of data are being taken to attain a superior vision of commercial methods, processes, products, consumers, etc [7]. The evolution of technologies and the constant group of data have presented special tasks, particularly to the data mining (DM) community. These tasks have inspired business analysts and researchers to frequently progress novel tools and models for enhancing the use of numerous machine learning (ML) methods [8]. The foremost objective is to recognize patterns, construct recommendations methods and, predictive techniques which will finally support the decision-making procedure within a group [9]. The ML methods application is wide and extends across dissimilar research fields [10].

The power consumption of mobile device offers the quality parameter to achieve effective functioning of mobile apps [11]. These quality parameter of the mobile applications and devices ensure process at both hardware and software levels. Quality parameters are power consumption and cost that enhance task scheduling (TS) with ML [12]. The TS processing of data permits mobile apps to enhance processing power and quality parameters [13]. Several recent research studies define the constraints of ML solution, for sample, certain application condition for mobile devices and offline model alteration for the system design [14]. Eventually, the power utilized by the mobile devices could not in agreement with provided case scenarios and improved the learning knowledge. The basic method was implemented utilizing these preceding systems, allowing the functionality of only a some solutions across all mobile applications. Dynamic adaptation and completely optimized solution are presented [15].

Enabling mobile data mining is an important benefit for nomadic users and organizations that essential to execute analysis of data created both mobile device and remote sources [16]. Mobile data mining may contain distinct conditions in which a mobile device can role the play of data producer, client of remote data miners, data analyzer, or combination of them. Therefore, an enhancing the count of smartphone and PDA-based data intensive applications are recently established [17]. Instances comprise smartphone-based systems for body-health monitoring, wireless security systems, and vehicle monitoring. New support for data analysis and mining was essential for these applications [18]. A basic feature that needs attention to allow effectual and reliable data mining under mobile devices is guaranteeing energy efficiency, as most commercially accessible mobile devices are battery power that would last only a few hours. Then, the next generation of mobile apps for such mobile devices are design to diminish the energy consumption. Therefore, an enhancing need to recognize the bottlenecks connected with the implementation of these applications in recent mobile-based structural design [19]. In recent years, there has been notable research focused on decreasing the computational complexity of data mining algorithms. Unfortunately, very little was done to ensure that data mining methods are complete functioning in mobile environment. According to our perception only inadequate studies are devoted to examine energy characterization of data mining methods on mobile devices [20].

This study introduces an innovative approach to monitoring the energy consumption of mobile devices interconnected to the Raspberry Pi via the web application interface. Particularly, the focus is on Android mobiles that are wirelessly connected to the Raspberry Pi through the WiFi network connection. This allows real-time monitoring of key energy metrics, such as overall energy consumption, CPU usage, and battery levels, which facilitates informed decision-making based on the RA. Moreover, the Raspberry Pi applies an XGBoost classifier to define and efficiently allocate resources and the idle state of connected devices based on their usage patterns. The architecture intends to improve energy efficacy, extend battery lifetime, and improve overall system performance by enhancing RA, thus contributing to resilient and sustainable mobile computing environments.

2. Related Works

Szabó and Pető [21] developed an advanced model for substituting wired communication in an effective agile manufacturing cell with wireless communication. The technique uses Reinforcement Learning (RL) to enhance the plan and decrease the vital amount of access points. The technique also projected an AI that simplifies supportive communication among access points and cameras, enhancing camera stream operation and radio convergence. Singh et al. [22] presented a resource distribution model for SDN-enabled fog calculating with Collaborative ML (CML) technique. This method was combined with the resource distribution method for the SDN-enabled fog-calculating atmosphere. The iFogSim and FogBus were used to assess the outcomes of the developed model utilizing numerous performance assessment metrics. In [23], a deadline-aware data offload system is presented utilizing DRL and dynamic voltage and frequency scaling (DVFS) in an EC atmosphere to decrease the energy utilization of IoT devices. The projected system absorbs the optimum data spreading strategies and local calculation DVFS frequency scaling by relating with the method atmosphere and learning the conduct of the edge servers, network and device.

Rahman et al. [24] presented a process management structure which is construed as an ML and cloud-based data-driven numeral for smart greenhouses. The developed structure contains 3 layers such as fog, cloud and physical. The physical greenery house dimensions were observed utilizing an extremely real 3D environment and immersive cloud-based. An instance structure has been projected utilizing business-related cloud and open-source devices to prove the evidence of model. Moreover, dissimilar ML methods are exploited to forecast the functioning necessities for smart greenhouses. Wu et al. [25] proposed a method separating and resource allocation technique to define the enhanced task of computing resources for DNN tasks unloaded from manifold devices against the edge server. The projected method initially uses the separating instructions to get an initial decision on model partitioning and presents a greedy-based approach to define the absolute decision on the partitioning facts of DNN frameworks and the quantity of computing resources allocated for task implementation.

Li et al. [26] merge graded federated learning (FL) with UAV-aided mobile edge computing (MEC) atmosphere to construct a UAV-aided MEC method structure for FL. This research paper presented incentive devices and Stackelberg game in FL. The communication among UAV, consumer devices, and the base station was demonstrated as a Stackelberg game to define the highest quantity of data needed for training. In [27], a containerized edge intelligence framework (CEIF) is proposed for a mobile-wearable IoT method. CEIF allows dynamic cause of the inference services of AI techniques and contains edge computing device

(ECD) to run the container abstraction model. Then, the model suggests a DL technique, where the container group plan absorbs the fluctuating user workload at the position of every ECD.

Minhaj et al. [28] present a novel model of utilizing 2 independent learning systems to allocate spreading factor (SF) and transmission power to devices utilizing a group of decentralized and centralized system. Lin et al. [29] proposal a distributed deep RL (DRL) based solution to enhance the task satisfaction ratio through equally optimizing the task offloading decision and the sub-channel transfer to assist the binary computing offloading strategy. In [30], a DL structure for optimize of the RA in multi-channel cellular methods with device-to-device (D2D) communication was presented. So, the channel assignment and discrete transfer power levels of D2D users are both integer variables, can optimized for growth of entire spectral efficacy whilst preserving the QoS of cellular users.

3. The Proposed Model

This study introduces an innovative approach to monitoring the energy consumption of mobile devices interconnected to the Raspberry Pi via the web application interface. Fig. 1 demonstrates the entire procedure of the proposed methodology.

3.1. Integration with Raspberry Pi

In this work, the Android devices are connected to the Raspberry Pi module. The devices are connected via Wifi links. Besides, the Raspberry Pi is integrated into web link for analysis. Raspberry Pi Foundation in UK in 2012 introduced Raspberry Pi, a reasonable, credit card-sized single-board computer to encourage teaching of basic computer skills in schools [31]. RPi became the fast-selling British computer around 5 million devices were vended within 3 years. RPi derives in Models A and B that vary with respect to technical specifications such as network connection, RAM and USB ports. Like desktop computer, it is a computer designed that do almost everything including video streaming, web surfing, computer programming, playing games and word processing. The RPi derives the energy it needs for the operation from three dissimilar sources. Initially, the device is interconnected through the 5V micro USB mains adaptor with the 1200mA current. Likewise, the Raspberry Pi is power-driven through USB-based portable battery compatible with smartphones. Another technique is to apply the Mobile Pi Power (MoPi) - a power regulator that provides the capability to switch electricity supply without interruption and provides various inputs (viz., standard battery, solar cells or car power sockets). Similarly, users can power the RPi through the battery box that runs with more the six AA batteries. RPi consumes power during its operation like other varieties of computers. It includes power necessary to run software or perform tasks on the platform and to function the hardware component. After calculating the power states, a power consumption system of Raspberry Pi, called PowerPi, was recommended in recent times. The diverse functionalities of power consumption existing within the platform weren’t examined even though the PowerPi models and measures RPi power consumption from different elements such as USB WiFi dongle, CPU and Ethernet mainly.

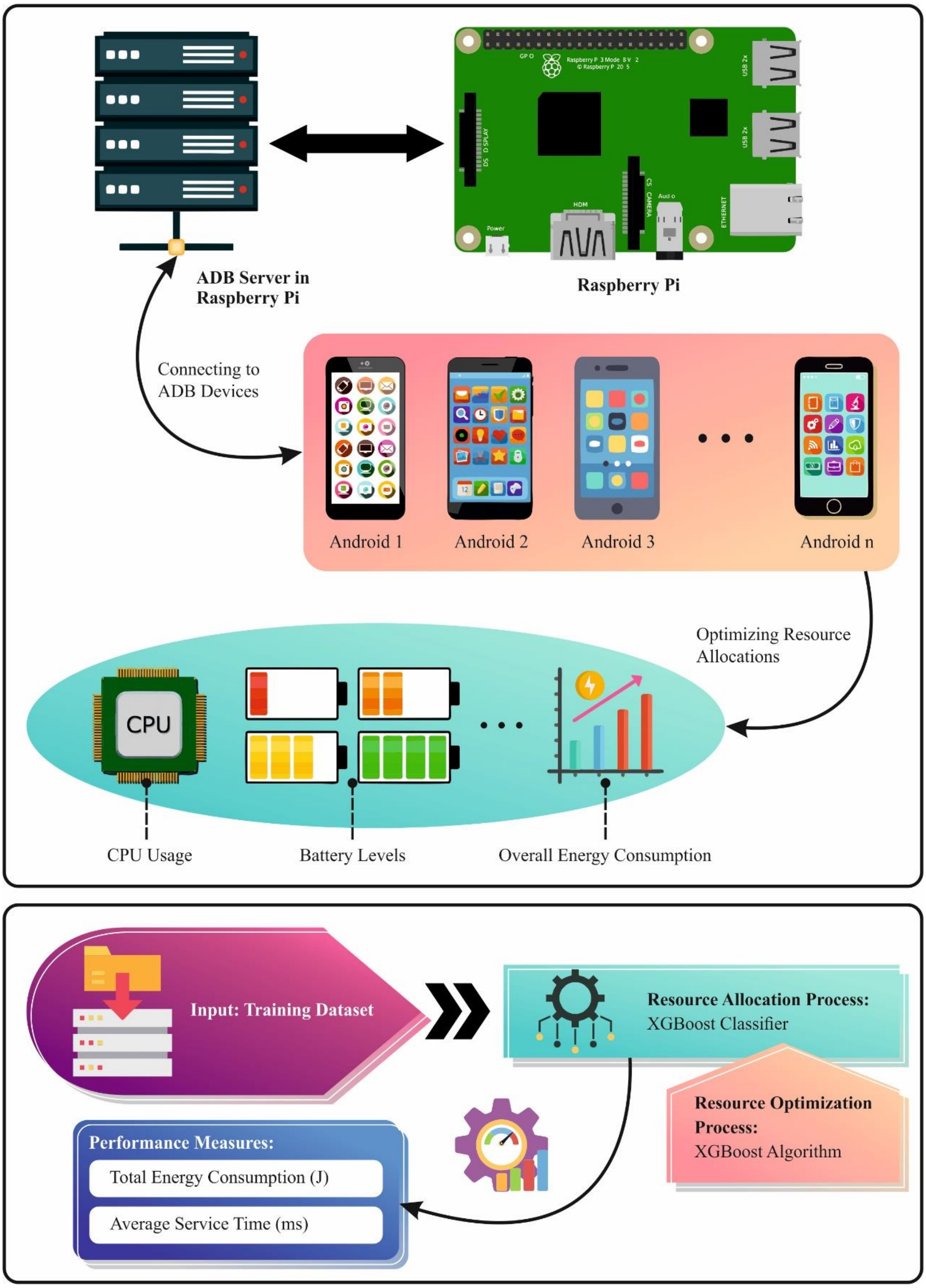

Fig. 1. Overall process of proposed model

3.2. Energy Consumption Monitoring and Analysis

The Raspberry Pi acts as a central monitoring hub for connected mobile devices, which oversees different metrics crucial for RA and energy consumption analysis. It continuously tracks the idle and active states of the mobile devices, which discerns whether they are dormant or used, which is essential for improving RA. Furthermore, the Raspberry Pi is used to monitor the usage of CPU, which provides insight into the computation workload of all the devices. This data assists in dynamically allocating CPU resources based on the demand, which ensures effective use. Additionally, the Raspberry Pi monitors on battery level, allowing proactive management of power resources to avoid early device shutdown. Additionally, energy consumption is accurately traced, permitting energy-intensive tasks or application detection for optimization. Finally, resource monitoring includes storage usage and monitoring CPU through connected devices, empowering effective storage capacity and computational resource distribution to meet different requirements. This wide-ranging monitoring scheme assisted by the Raspberry Pi allows dynamic RA and energy consumption analysis for enriched efficiency and performance in mobile computing environment.

- Idle or Active State: The Raspberry Pi continuously track the connected mobile devices that are in idle or active state. This is critical for RA decision, as active device might need more resources than idle one. For example, an active device may be performing tasks or running applications that demand higher usage of CPU, while idle device requires less resources.
- CPU Usage: Monitoring CPU usage offers insights into the computation job of the connected mobile device. The Raspberry Pi can dynamically adjust RA to ensure optimum performance by constantly tracing CPU usage. For instance, if the device's CPU usage spikes, representing heavy computation task, the Raspberry Pi could assign further CPU resources to that device to delay processing or avoid slowdown.
- Battery Monitor: Battery monitoring includes trailing the battery level of connected mobile devices in real time. These metrics are crucial to prevent unexpected shutdowns and effectively manage power resources owing to low battery levels. The Raspberry Pi can alert users or implement power-saving measures when batteries need charging by monitoring battery levels.
- Energy Consumption: The Raspberry Pi is used to measure the energy consumption of connected mobile device, which provide insight into power usage patterns. This assists in identifying energy-intensive applications or tasks that might quickly drain battery lifetime. The Raspberry Pi can implement energy-saving strategies or optimizations by analyzing energy consumption patterns, to enhance overall efficiency and extend battery life.

- Resource Monitoring (CPU, Storage): Resource monitoring involves tracing storage utilization and CPU usage through connected mobile devices. Monitoring CPU usage aids in effectively allocating computation resources, while monitoring storage usage ensures optimum use of storage capacity. For example, if the device is running lower on storage space, the Raspberry Pi prompts user to perform storage management techniques or offload data to prevent degradation performance and free up space.

The Raspberry Pi can effectively analyze energy consumption patterns and dynamically allocate resources to connected mobile gadgets by monitoring the metrics in detail, thus enhancing efficiency and performance in mobile computing environment.

3.3. Machine Learning-based Resource Allocation of Idle Devices

Implementing ML-based RA for idle mobile devices is critical for maximizing efficiency and enhancing energy consumption. By leveraging complex techniques, namely neural networks or reinforcement learning, on platforms such as Raspberry Pi, idle mobile device is intelligently identified and assigned resources according to the predicted usage pattern. This technique ensures that resource is efficiently distributed, minimalizing energy consumption while concurrently improving the responsiveness and system performance. In this work, the XGBoost model is applied for the allocation of resources to idle devices. The XGBoost technique is an effectual gradient boosting decision tree (GBDT) method that is enhanced from the GBDT [32]. However, the forward additional process is most common approach to boosting processes. This method iteratively creates a novel tree by fitting residuals and developing a classifier with superior accuracy and stronger generalization proficiency. The fundamental regression tree (RT) method employed from the XGBoost approach is defined as:

$$y_i = \sum_{t=1}^{K} f_k\,(x_i), f_k(xi) \in R \tag{1}$$

whereas K represents the tree counts; f_k denotes the function under the function space R, y_i implies the predicting rate of the RT; x_i stands for the i^{th} data input and R indicates the set of every probable RT method. The main function of the XGBoost technique can expressed in Eq. (2).

$$X_{obj} = \sum_{i=1}^{n} l\,(y, y) + \sum_{k=1}^{K} \Omega\,(f_k) \tag{2}$$

In which, $l(y, \hat{y})$ denotes the difference between the predicting rate of the model and actual rate, and $\Omega(f_k)$ represents the regular term of the scalar function. Fig. 2 illustrates the infrastructure of XGBoost model.

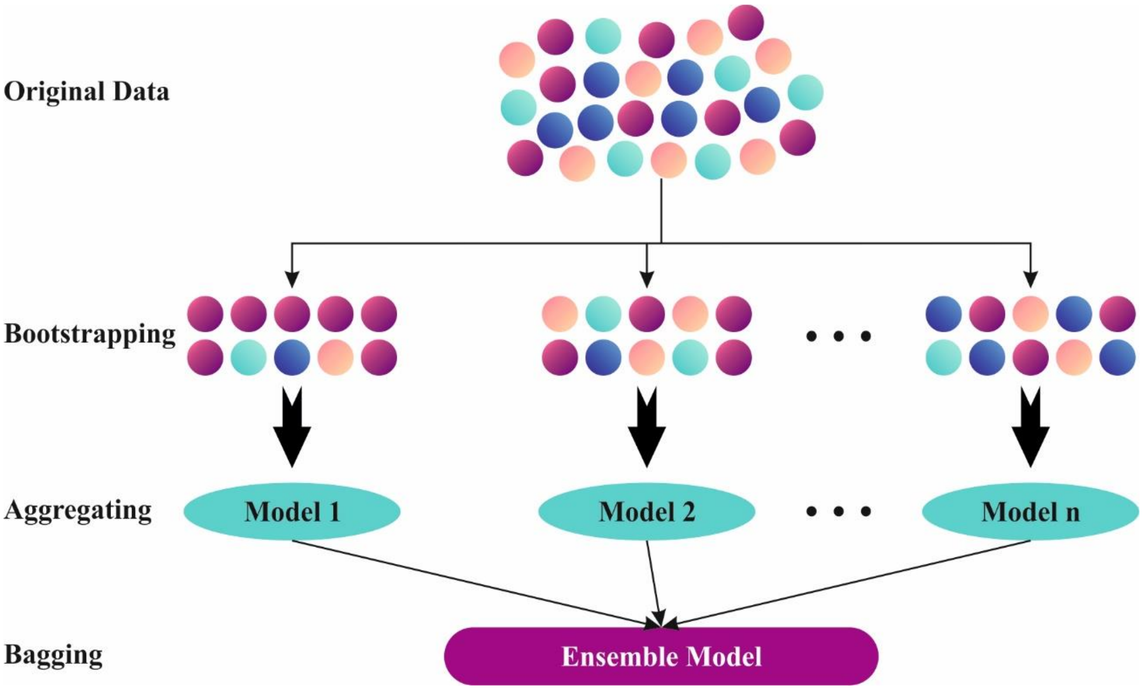

Fig. 2. Structure of XGBoost Model

The regularized penalty function has been employed to avoid overfitting the model, as defined in Eq. (3).

$$\Omega(f_k) = \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} \omega_j^2 \tag{3}$$

whereas T stands for the amount of leaf nodes; γ represents the penalty function coefficient; ω refers to the score of leaf nodes, and λ implies the regularized penalty coefficient.

4. Result analysis

This section inspects the performance of the proposed technique under distinct measures. The proposed model is simulated using Raspberry Pi 3 with Android Debugging Tool, React JS Device Monitoring, and Python JSON Server.

Table 1 and Fig. 3 show the total energy consumption (TEC) results of the proposed system [33, 34]. The results reported that the proposed approach reached effective results with least TEC values. With 200kHz bandwidth, the proposed model gains decreased TEC value of

0.8216J while LC, MC, BHGC, and JTOREAH approaches attain increased TEC values of 1.8305J, 1.3565J, 1.0782J, and 0.9390J, correspondingly. Besides, with 400kHz bandwidth, the proposed model acquires reduced TEC value of 0.4867J while LC, MC, BHGC, and JTOREAH techniques achieve enlarged TEC values of 1.8653J, 0.8738J, 0.6737J, and 0.5737J, respectively. Also, with 600kHz bandwidth, the proposed system gains reduced TEC value of 0.4171J whereas LC, MC, BHGC, and JTOREAH methodologies get enlarged TEC values of 1.8740J, 0.7650J, 0.5606J, and 0.4736J, respectively.

Table 1 TEC analysis of proposed model with recent approaches under various bandwidth

Total Energy Consumption (J)					
Bandwidth (KHz)	**LC Algorithm**	**MC Algorithm**	**BHGC Algorithm**	**JTORAEH Algorithm**	**Proposed Method**
200	1.8305	1.3565	1.0782	0.9390	0.8216
300	1.8784	0.9868	0.7911	0.6694	0.5476
400	1.8653	0.8738	0.6737	0.5737	0.4867
500	1.8784	0.8216	0.6346	0.5171	0.4476
600	1.8740	0.7650	0.5606	0.4736	0.4171

a

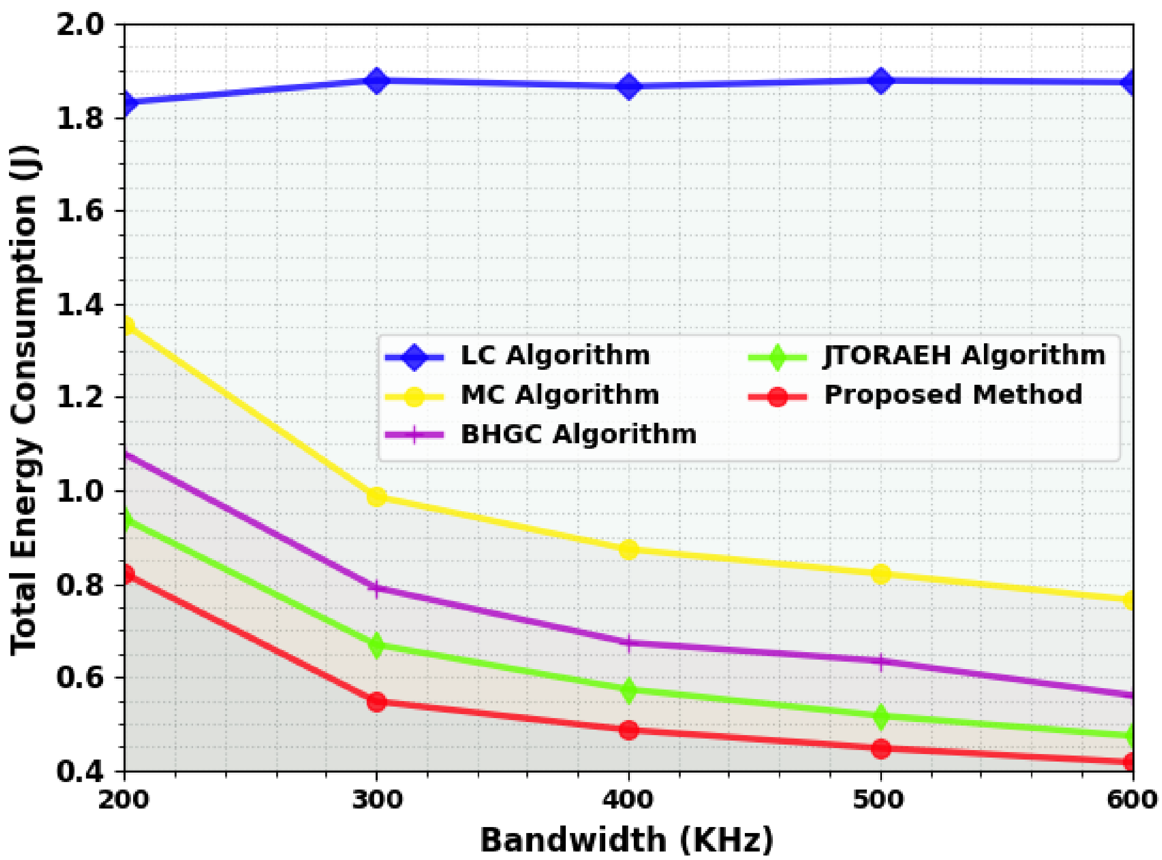

Fig. 3. TEC analysis of proposed model under various bandwidth

Table 2 and Fig. 4 show the TEC outcomes of the proposed technique below numerous CPU-cycle frequencies. The results stated that the proposed methodology got effectual outcomes with minimum TEC values. With 200GHz CPU-cycle frequency, the proposed approach gains declined TEC value of 0.7329J while LC, MC, BHGC, and JTOREAH methods attain enlarged TEC values of 1.8691J, 1.3501J, 1.0789J, and 0.9433J, respectively.

Table 2 TEC analysis of proposed model with recent approaches under various CPU-cycle frequency

Total Energy Consumption (J)					
CPU-cycle Frequency (GHz)	**LC Algorithm**	**MC Algorithm**	**BHGC Algorithm**	**JTORAEH Algorithm**	**Proposed Method**
200	1.8691	1.3501	1.0789	0.9433	0.7329
300	1.8551	0.9106	0.7236	0.5973	0.4150
400	1.8925	0.7142	0.5506	0.4570	0.3261
500	1.8598	0.5693	0.4103	0.3308	0.2513
600	1.8644	0.5038	0.3682	0.2934	0.2420

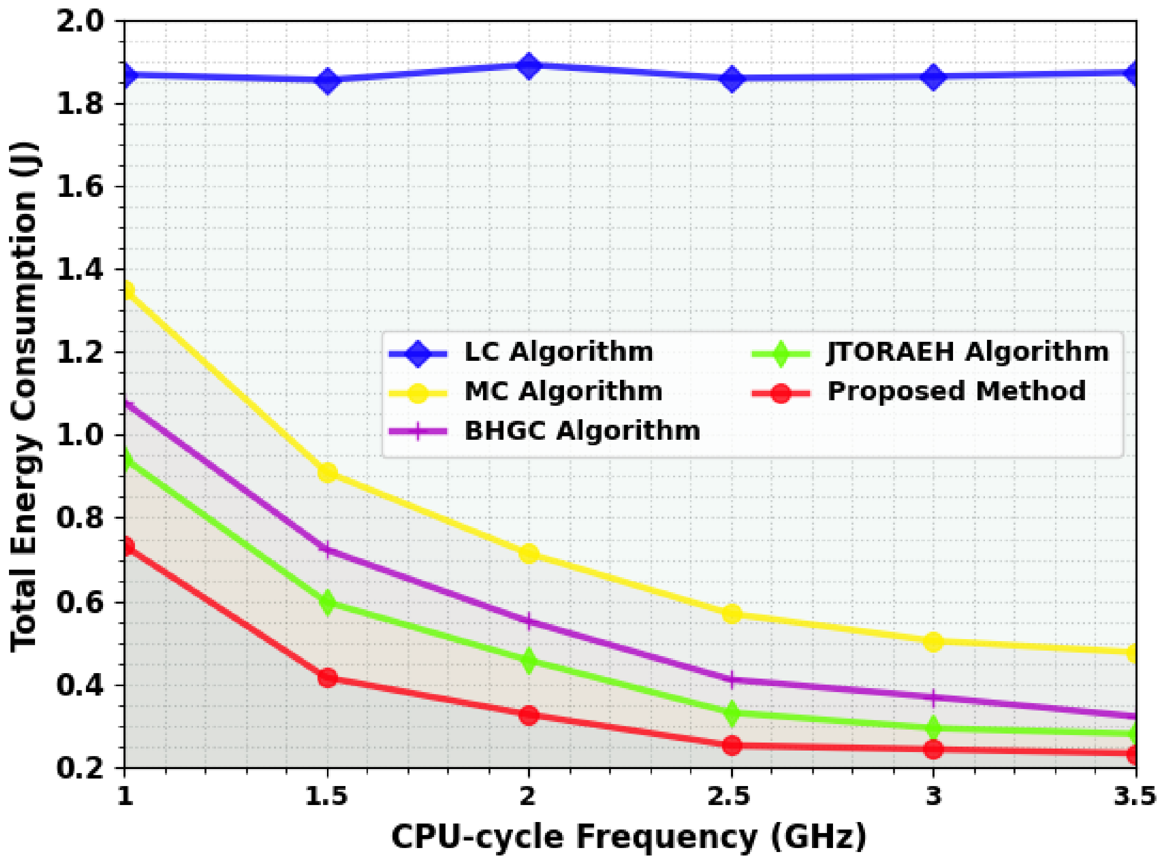

Fig. 4. TEC analysis of proposed model under various CPU-cycle frequency

Besides, with 400GHz CPU-cycle frequency, the proposed system gets reduced TEC value of 0.3261J whereas LC, MC, BHGC, and JTOREAH techniques achieve enlarged TEC values of 1.8925J, 0.7142J, 0.5506J, and 0.4570J, respectively. Also, with 600GHz CPU-cycle frequency, the proposed method obtains diminished TEC value of 0.2420J while LC, MC, BHGC, and JTOREAH models attain increased TEC values of 1.8644J, 0.5038J, 0.3682J, and 0.2934J, respectively.

Table 3 and Fig. 5 show the TEC results of the proposed model under numerous input data. The results described that the proposed system got effectual outcomes with minimum TEC values. With 200Mbits input data, the proposed technique gains declined TEC value of 0.9151J while LC, MC, BHGC, and JTOREAH methods achieve enlarged TEC values of 1.8572J, 1.3509J, 1.0814J, and 0.9327J, respectively. Moreover, with 400Mbits input data, the proposed model obtains decreased TEC value of 0.9781J while LC, MC, BHGC, and JTOREAH approaches attain amplified TEC values of 1.8799J, 1.4718J, 1.1821J, and 1.0335J, correspondingly. Also, with 600Mbits input data, the proposed model gains decreased TEC value of 1.1015J while LC, MC, BHGC, and JTOREAH methodologies reach enlarged TEC values of 1.8698J, 1.6758J, 1.4113J, and 1.2526J, respectively.

Table 3 TEC analysis of proposed model with recent approaches under various input data

Total Energy Consumption (J)					
Input Data (Mbits)	**LC Algorithm**	**MC Algorithm**	**BHGC Algorithm**	**JTORAEH Algorithm**	**Proposed Method**
200	1.8572	1.3509	1.0814	0.9327	0.9151
300	1.8496	1.3962	1.1015	0.9680	0.9302
400	1.8799	1.4718	1.1821	1.0335	0.9781
500	1.8496	1.5700	1.2829	1.1267	1.0310
600	1.8698	1.6758	1.4113	1.2526	1.1015

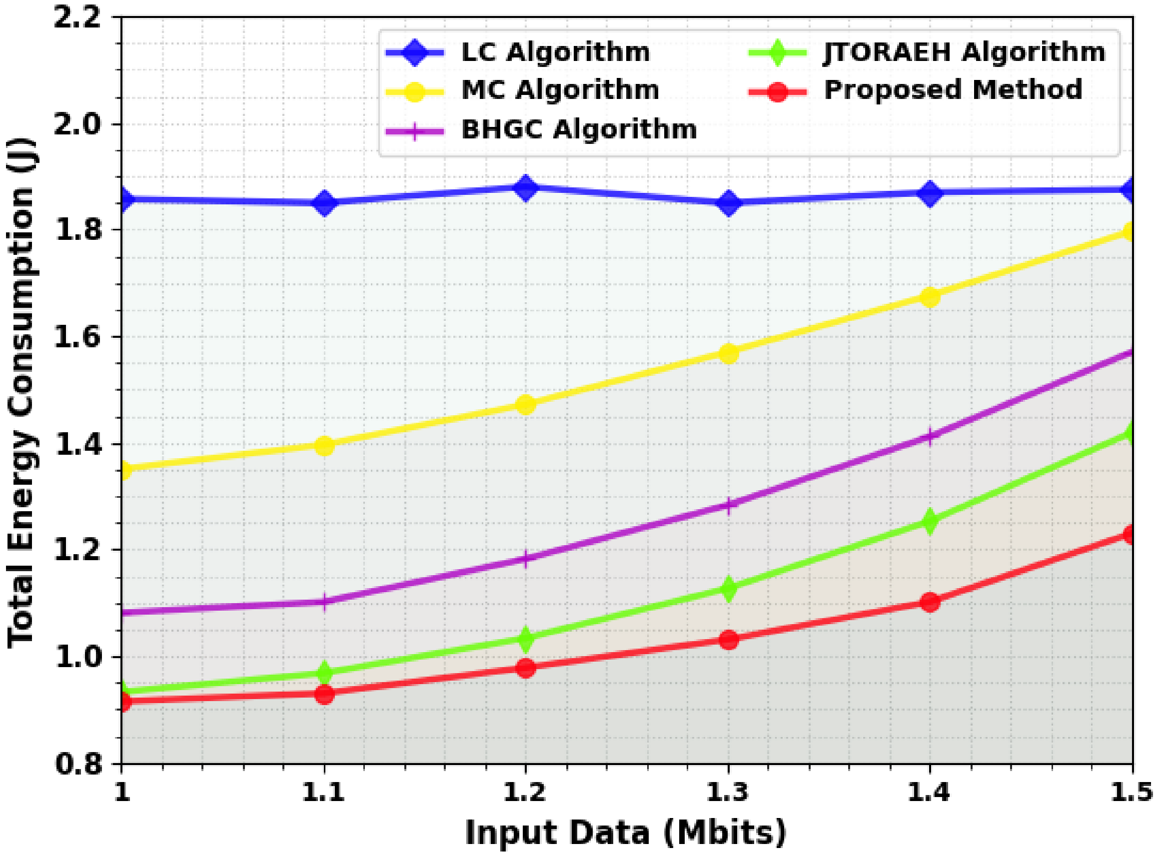

Fig. 5. TEC analysis of proposed model under various input data

Table 4 and Fig. 6 display the average service time (AST) outcomes of the proposed model under different computing requirements of applications (CROA). The results described that the proposed method got effective outcomes with least AST values. With 50 megacycles of CROA, the proposed model gets reduced AST value of 38.74ms whereas OSPF and DRLRA techniques get enlarged AST values of 68.12ms and 58.77ms, correspondingly. Also, with 250 megacycles of CROA, the proposed technique gains reduced AST value of 149.60ms while OSPF and DRLRA methods reach enlarged AST values of 252.44ms and 207.03ms, respectively. Also, with 400 megacycles of CROA, the proposed model gains declined AST value of 272.48ms whereas OSPF and DRLRA methods achieve improved AST values of 410.05ms and 336.59ms, respectively.

Table 4 AST analysis of proposed model with recent approaches under various CROA

Average Service Time (ms)			
Computing Requirement of Applications (Megacycles)	**OSPF**	**DRLRA**	**Proposed Method**
50	68.12	58.77	38.74

100	136.24	102.85	60.11
150	156.28	117.54	78.81
200	221.72	193.68	137.58
250	252.44	207.03	149.60
300	273.82	220.39	185.66
350	352.62	247.10	200.35
400	410.05	336.59	272.48

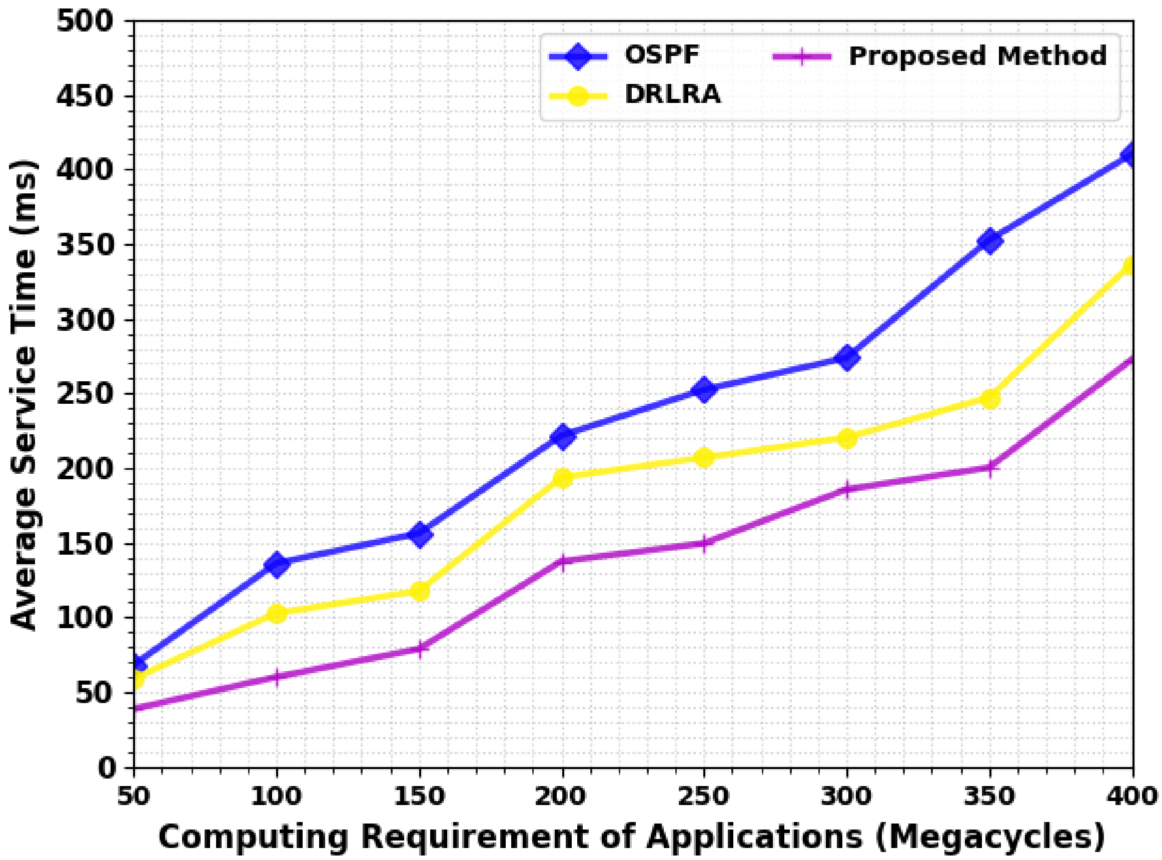

Fig. 6. Computing Requirement of applications on average service time

Table 5 and Fig. 7 demonstrate the AST results of the proposed method below several data routing capacity (DRC). The outcomes described that the proposed technique grabbed effectual outcomes with least AST values. With 250Mbps DRC, the proposed system acquires decreased AST value of 100.85ms whereas OSPF and DRLRA models collect increased AST values of 142.14ms and 128.84ms, respectively. Besides, with 750Mbps DRC, the proposed model gains diminished AST value of 81.06ms while OSPF and DRLRA methodologies attain amplified AST values of 114.84ms and 95.74ms, correspondingly. Also, with 1000Mbps DRC, the proposed approach obtains reduced AST value of 64.00ms while OSPF and DRLRA systems get increased AST values of 91.30ms and 73.21ms, respectively.

Table 5 AST analysis of proposed model with recent approaches under various DRC

Average Service Time (ms)			
Data Routing Capacity (Mbps)	**OSPF**	**DRLRA**	**Proposed Method**
250	142.14	128.84	100.85
500	130.20	103.58	87.20
750	114.84	95.74	81.06
1000	91.30	73.21	64.00

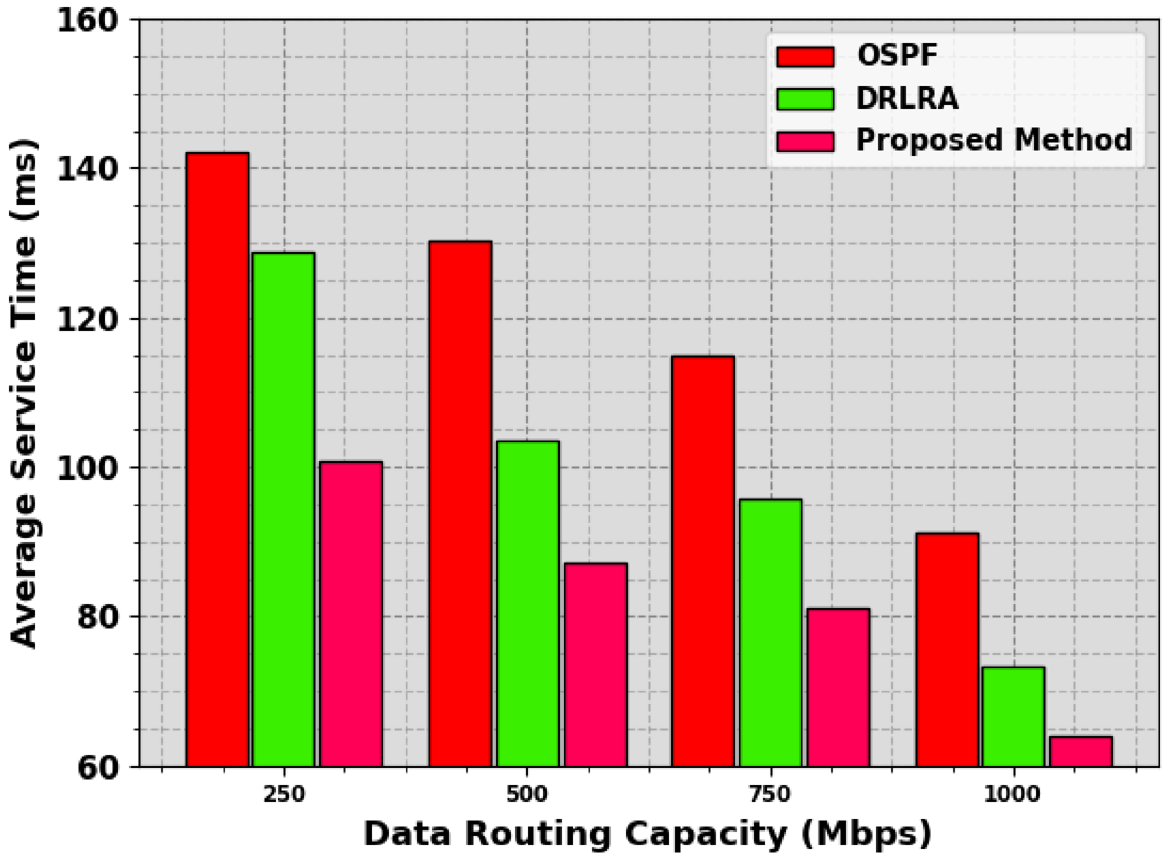

Fig. 7. AST analysis of proposed model under different data routing capacities

5. Conclusion

In this study, we have introduced an innovative approach to monitoring the energy consumption of mobile devices interconnected to the Raspberry Pi via the web application interface. Particularly, the focus is on Android mobiles that are wirelessly connected to the Raspberry Pi through the WiFi network connection. This allows real-time monitoring of key energy metrics, such as overall energy consumption, CPU usage, and battery levels, which facilitates informed decision-making based on the RA. Moreover, the Raspberry Pi applies an XGBoost classifier to define and efficiently allocate resources and the idle state of connected devices based on their usage patterns. The system can intelligently allocate resources and predict device usage based on real-time energy demands through data collection and analysis, combined with ML techniques like XGBoost. The architecture intends to improve energy efficacy, extend battery lifetime, and improve overall system performance by enhancing RA, thus contributing to resilient and sustainable mobile computing environments.

References

[1] Routis, G., Michailidis, M. and Roussaki, I., 2024. Plant Disease Identification Using Machine Learning Algorithms on Single-Board Computers in IoT Environments. *Electronics*, *13*(6), p.1010.

[2] Zhang, C., Peng, C., Lin, M., Du, Z. and Wu, C., 2023, October. Double DQN Reinforcement Learning-Based Computational Offloading and Resource Allocation for MEC. In *International Conference on Mobile Networks and Management* (pp. 240-253). Cham: Springer Nature Switzerland.

[3] Singh, D., Singh, A.K., Goyal, M., Singh, A. and Sinha, D., 2022. Intelligent model for managing energy efficiency of public sector by using machine learning and IoT system.

[4] Islam, A. and Ghose, M., 2024, January. ELITE: Energy and Latency-Optimized Task Offloading for DVFS-Enabled Resource-Constrained Devices in MEC. In *International Conference on Distributed Computing and Intelligent Technology* (pp. 50-67). Cham: Springer Nature Switzerland.

[5] Makina, H. and Ben Letaifa, A., 2023. Bringing intelligence to Edge/Fog in Internet of Things-based healthcare applications: Machine learning/deep learning-based use cases. *International Journal of Communication Systems*, *36*(9), p.e5484.

[6] Sami, T.M.G., Zeebaree, S.R. and Ahmed, S.H., 2024. Designing a New Hashing Algorithm for Enhancing IoT Devices Security and Energy Management. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(4s), pp.202-215.

[7] I. Varalakshmi, M. Thenmozhi and R. Sasi, "Detection of Distributed Denial of Service Attack in an Internet of Things Environment -A Review," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 2021, pp. 1-6, doi: 10.1109/ICSCAN53069.2021.9526378.

[8] Varalakshmi, Mrs I., and M. Thenmozhi. "Mitigation of DDoS attack using machine learning algorithms in SDN_IoT environment." Design Engineering (2021): 4381-4390.

[9] Wang, Z., Goudarzi, M., Gong, M. and Buyya, R., 2024. Deep Reinforcement Learning-based scheduling for optimizing system load and response time in edge and fog computing environments. *Future Generation Computer Systems*, *152*, pp.55-69.

[10] Tayel, A.F.M., Proietti Mattia, G. and Beraldi, R., 2023, September. A Double-Decision Reinforcement Learning Based Algorithm for Online Scheduling in Edge and Fog Computing. In *International Symposium on Algorithmic Aspects of Cloud Computing* (pp. 197-210). Cham: Springer Nature Switzerland.

[11] Wang, S., Chen, M., Liu, X., Yin, C., Cui, S. and Poor, H.V., 2020. A machine learning approach for task and resource allocation in mobile-edge computing-based networks. *IEEE Internet of Things Journal*, *8*(3), pp.1358-1372.

[12] Vimal, S., Khari, M., Dey, N., Crespo, R.G. and Robinson, Y.H., 2020. Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT. *Computer Communications*, *151*, pp.355-364.

[13] Abdullaev, I., Prodanova, N., Bhaskar, K.A., Lydia, E.L., Kadry, S. and Kim, J., 2023. Task offloading and resource allocation in iot based mobile edge computing using deep learning. *Computers, Materials & Continua*, *76*(2), pp.1463-1477.

[14] Hassan, M.U., Al-Awady, A.A., Ali, A., Iqbal, M.M., Akram, M. and Jamil, H., 2024. Smart Resource Allocation in Mobile Cloud Next-Generation Network (NGN) Orchestration with Context-Aware Data and Machine Learning for the Cost Optimization of Microservice Applications. *Sensors*, *24*(3), p.865.

[15] Chua, T.J., Yu, W. and Zhao, J., 2022, October. Resource allocation for mobile metaverse with the Internet of Vehicles over 6G wireless communications: A deep reinforcement learning approach. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)* (pp. 1-7). IEEE.

[16] Zheng, J., Li, K., Mhaisen, N., Ni, W., Tovar, E. and Guizani, M., 2023, March. Federated learning for online resource allocation in mobile edge computing: A deep reinforcement learning approach. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1-6). IEEE.

[17] Li, M., Pei, P., Yu, F.R., Si, P., Li, Y., Sun, E. and Zhang, Y., 2022. Cloud–edge collaborative resource allocation for blockchain-enabled Internet of Things: A collective reinforcement learning approach. *IEEE Internet of Things Journal*, *9*(22), pp.23115-23129.

[18] Zhang, H., Zhang, H., Long, K. and Karagiannidis, G.K., 2020. Deep learning based radio resource management in NOMA networks: User association, subchannel and power

allocation. *IEEE Transactions on Network Science and Engineering*, *7*(4), pp.2406-2415.

[19] Consul, P., Budhiraja, I., Garg, D., Kumar, N., Singh, R. and Almogren, A.S., 2024. A Hybrid Task Offloading and Resource Allocation Approach For Digital Twin-Empowered UAV-Assisted MEC Network Using Federated Reinforcement Learning For Future Wireless Network. *IEEE Transactions on Consumer Electronics*.

[20] Ning, Z., Sun, S., Wang, X., Guo, L., Wang, G., Gao, X. and Kwok, R.Y., 2021. Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach. *Science China Information Sciences*, *64*(6), p.162303.

[21] Szabó, G. and Pető, J., 2024. Intelligent wireless resource management in industrial camera systems: Reinforcement Learning-based AI-extension for efficient network utilization. *Computer Communications*, *216*, pp.68-85.

[22] Singh, J., Singh, P., Hedabou, M. and Kumar, N., 2023. An efficient machine learning-based resource allocation scheme for sdn-enabled fog computing environment. *IEEE Transactions on Vehicular Technology*.

[23] Panda, S.K., Lin, M. and Zhou, T., 2022. Energy-efficient computation offloading with DVFS using deep reinforcement learning for time-critical IoT applications in edge computing. *IEEE Internet of Things Journal*, *10*(8), pp.6611-6621.

[24] Rahman, H., Shah, U.M., Riaz, S.M., Kifayat, K., Moqurrab, S.A. and Yoo, J., 2024. Digital twin framework for smart greenhouse management using next-gen mobile networks and machine learning. *Future Generation Computer Systems*.

[25] Wu, Q., Zhang, Y., Yang, C. and Sun, J., 2023, December. Joint Optimization of Model Partitioning and Resource Allocation for Edge Computing with Intermittently Operating Devices. In *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 2186-2193). IEEE.

[26] Li, C., Song, M. and Luo, Y., 2024. Federated learning based on Stackelberg game in unmanned-aerial-vehicle-enabled mobile edge computing. *Expert Systems with Applications*, *235*, p.121023.

[27] Nkenyereye, L., Baeg, K.J. and Chung, W., 2023. Deep Reinforcement Learning for Containerized Edge Intelligence Inference Request Processing in IoT Edge Computing. *IEEE Transactions on Services Computing*.

[28] Minhaj, S.U., Mahmood, A., Abedin, S.F., Hassan, S.A., Bhatti, M.T., Ali, S.H. and Gidlund, M., 2023. Intelligent resource allocation in LoRaWAN using machine learning techniques. *IEEE Access*, *11*, pp.10092-10106.

[29] Lin, L., Zhou, W.A., Yang, Z. and Liu, J., 2023. Deep reinforcement learning-based task scheduling and resource allocation for NOMA-MEC in Industrial Internet of Things. *Peer-to-Peer Networking and Applications*, *16*(1), pp.170-188.

[30] Lee, W. and Schober, R., 2022. Deep learning-based resource allocation for device-to-device communication. *IEEE Transactions on Wireless Communications*, *21*(7), pp.5235-5250.

[31] Bekaroo, G. and Santokhee, A., 2016, August. Power consumption of the Raspberry Pi: A comparative analysis. In *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)* (pp. 361-366). IEEE.

[32] Wang, H., Yan, S., Ju, D., Ma, N., Fang, J., Wang, S., Li, H., Zhang, T., Xie, Y. and Wang, J., 2023. Short-term photovoltaic power forecasting based on a feature rise-dimensional two-layer ensemble learning model. *Sustainability*, *15*(21), p.15594.

[33] Li, S., Zhang, N., Jiang, R., Zhou, Z., Zheng, F. and Yang, G., 2022. Joint task offloading and resource allocation in mobile edge computing with energy harvesting. *Journal of Cloud Computing*, *11*(1), p.17.

[34] Wang, J., Zhao, L., Liu, J. and Kato, N., 2019. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach. *IEEE Transactions on emerging topics in computing*, *9*(3), pp.1529-1541.