International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Energy aware Multi Agent Deep Queue Optimization for efficient Resource Allocation and Task Offloading in Cloud Edge

Vidya S[1], Gopalakrishnan R[2], Satheesh Kumar[3]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract*:* IoT devices finds its application in almost all the fields leading to streaming of enormous amount of big and small data. This leads to many operational problems in load balancing, energy management, latency in processing and storage methods. Edge and cloud computing is leveraged as a potential solution to resolve these problems through its resourceful architectures and on demand services. However, achieving optimal energy efficiency and latency in time critical medical applications is still an open-ended research topic, that draws the attention of the researchers. This work proposes Energy aware Multi Agent Deep Queue Optimisation (E-MADQ) technique that classifies the priorities of the medical tasks using Ensemble Empirical Mode Decomposition (EEMD) smoothened with Extreme Gradient Boost (XGBoost) by extracting Intrinsic Mode Functions and averaging their spectrum characteristics. The offloading decision is made using the multi-agent deep queue optimization where the rewards are calculated based on the energy level of the IoT devices, which is very crucial parameter. By this, the tasks that demands high attention will not be left in starvation and proper resources allocation is done for time critical tasks. The experimental simulation of the proposed methodology shows that a good improvement in service parameters such as mean delivery time, communication and computing delay, execution time and energy level can be attained. In future, the classification of task priorities can be done with powerful deep learning techniques with more focus on dynamism.

Keywords: *Edge Computing, task offloading, multi agent queue, Intrinsic mode functions, Empirical mode decomposition, XGBoost*

1. Introduction

The rapid advancement of mobile communication technology, Cloud Computing (CC), and Internet of Things (IoT) has led to their widespread adoption across various domains. These technologies, alongside others, offer convenient services to end-users. The primary aim of IoT is to efficiently manage resources based on user demands and transform vast amounts of heterogeneous data from IoT devices into usable information.

However, IoT devices face increased computing and energy loads due to their time-critical nature and resource consumption. Computational offloading, where intensive tasks are transferred to other systems for processing, is a viable solution. Mobile Cloud Computing (MCC) enables IoT devices to offload computing tasks to cloud servers, thus reducing power consumption and extending battery life. However, offloading tasks to cloud servers may result in higher transmission delays and communication overheads. Mobile Edge Computing (MEC) emerges as a promising paradigm to address these challenges by processing tasks closer to end devices, reducing latency and improving efficiency.

Nevertheless, issues arise in distributing and handling sub-tasks within edge and main clouds. Efficient offloading algorithms are crucial for coordinating Edge Clouds (EC) and end devices, considering factors like device energy, bandwidth, connectivity, application latency, and workload distribution. Resolving offloading challenges is critical for MEC's success, emphasizing the need for more efficient approaches and thoughtful design considerations during EC setup.

1.1.objective

The " Energy aware Multi Agent Deep Queue Optimization for efficient Resource Allocation and Task Offloading in Cloud Edge" project is to tackle the operational hurdles encountered by Internet of Things (IoT) devices across diverse sectors, with a particular emphasis on addressing challenges within time-sensitive medical applications. The proposed solution, the Energy-aware Multi-Agent Deep Queue Optimization (E-MADQO) technique, is devised to enhance energy efficiency and diminish latency in processing critical medical tasks, critical for healthcare professionals striving for optimized patient care.

The core of the proposed model lies in its ability to leverage advanced techniques such as Ensemble Empirical Mode Decomposition (EEMD) coupled with Extreme Gradient Boost (XGBoost) to classify medical tasks based on their priority levels. By considering factors such as the criticality

1 Full Time Research Scholar Department of Electrical and Electronics Engineering K.S. Rangasamy College of Technology
Tiruchengode, Tamil Nadu, India.
Mail ID: svidyaeeeksr@gmail.com
2 Professor and Principal Department of Electrical and
Electronics Engineering K.S. Rangasamy college of Technology
Tiruchengode, Tamil Nadu, India.
Mail ID: gopsengr@gmail.com
3 Assistant Professor Department of Electrical and Electronics Engineering
Sona College of Technology Salem, Tamil Nadu, India.
Mail ID: indsatkr@gmail.com

and computational intensity of each task, the model can effectively discern between urgent medical procedures and less time-sensitive operations. This prioritization enables the system to allocate resources dynamically, ensuring that high-priority tasks receive the necessary attention while mitigating latency and optimizing power consumption.

Moreover, the model incorporates insights from medical professionals, tapping into their expertise to label applications and prioritize treatments accurately. By integrating domain knowledge into the classification process, the model enhances the precision of task classification, thereby improving the overall efficiency of the system..

To validate the effectiveness of the proposed approach, the project employs experimental simulations to assess various service parameters. These parameters include mean delivery time, communication and computing delay, execution time, and energy levels. Through rigorous testing and analysis, the project aims to demonstrate tangible improvements in these metrics, illustrating the efficacy of the E-MADQO technique in enhancing the performance of IoT devices in medical settings.

By addressing the operational challenges faced by IoT devices in time-critical medical applications, the project contributes to the advancement of more efficient and responsive healthcare systems. The reduction in latency and enhancement of energy efficiency not only improve patient care but also have broader implications for healthcare delivery, enabling healthcare professionals to make more informed decisions and optimize resource utilization.

Overall, the project represents a significant step forward in harnessing advanced computational techniques to address real-world challenges in healthcare. Through its innovative approach and rigorous validation, the project aims to pave the way for the adoption of cutting-edge technologies in medical settings, ultimately leading to improved patient outcomes and more efficient healthcare delivery.

2.Problem Statement

The problem addressed in this project revolves around the operational challenges encountered by Internet of Things (IoT) devices, particularly in time-critical medical applications. As IoT devices are increasingly deployed across various fields, they generate vast amounts of data, leading to issues such as load balancing, energy management, and latency in processing and storage. While edge and cloud computing have been proposed as potential solutions, achieving optimal energy efficiency and latency in medical applications remains a significant research gap.

The core issue lies in the computational and energy burdens placed on IoT devices, especially in scenarios where timely processing of medical data is critical for patient care. Current approaches often lack efficiency in resource allocation and task prioritization, leading to suboptimal performance in terms of both energy consumption and latency. Traditional methods do not adequately address the dynamic nature of medical tasks, where certain procedures require immediate attention while others can afford some delay.

To tackle this problem, the project proposes the Energy-aware Multi-Agent Deep Queue Optimization (E-MADQO) technique. This novel approach aims to classify medical tasks based on priority using Ensemble Empirical Mode Decomposition (EEMD) smoothed with Extreme Gradient Boost (XGBoost). By extracting intrinsic features from medical signals, the model can discern critical tasks from less urgent ones, allowing for more efficient resource allocation.

3.Existing System

Existing systems for addressing operational challenges faced by IoT devices, particularly in time-critical medical applications, often rely on traditional cloud computing and basic edge computing solutions. Cloud computing involves centralized data processing and storage in remote servers, which can lead to latency issues and high communication overheads, especially for real-time applications like healthcare. On the other hand, basic edge computing solutions involve offloading tasks to nearby edge devices, which may not always be efficient in terms of resource allocation and load balancing.

In traditional cloud computing, IoT devices typically send all data to remote cloud servers for processing and storage. While this approach centralizes computational resources, it often results in high latency due to the distance between the IoT devices and the cloud servers. Moreover, the transmission of large amounts of data over the network can lead to increased communication overhead and may not be suitable for time-critical applications like medical diagnostics.

Basic edge computing solutions involve offloading tasks to nearby edge devices, such as routers or gateways, for processing. While this approach can reduce latency by processing data closer to the source, it may not always be efficient in terms of resource allocation and load balancing. Moreover, edge devices may have limited computational capabilities and storage capacity, which can affect the performance of time-critical applications.

4.Literature Survey

Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset, (2022), G. Karatas, O. Demir, and O. K. Sahingoz, This paper surveys the state-of-the-art in programmable networks with an emphasis on SDN. We provide a historic perspective of programmable networks from early ideas to recent developments. Then we present the SDN architecture and the Open Flow standard in particular, discuss current alternatives for implementation and testing of SDN-based protocols and services, examine current and future SDN applications, explore promising research directions based on the SDN paradigm. [1].

The work discusses the pros and cons of differ genres of approaches such as mathematical models, heuristic algorithms, game theory, graph theory, Lyapunov optimization, Reinforcement learning and Markov Decision Process. A novel Lyapunov optimization-based dynamic offloading algorithm is proposed by Kumaran et al. which integrates the offloading decision and the CPU-computation cycles for the execution of mobile applications [2].

The chief contribution of this approach is that the offloading decision rely only on current system state without any information distribution. However, this algorithm has been validated only on mobile device, which lacks genericity. Dinh et al. deployed a multi-user and multi-edge-node offloading problem, which is formulated as non-cooperative exact potential game [3].

The MU maximises its processing capacity in selfish manner in static channels to achieve Nash equilibrium by employing best response-based offloading method. Though the method shows superior performance, the user mobility was not considered in this work. Ke Zhang et al. formulated the offloading of computations and file transfers as an optimization problem, with mitigated energy consumption [4].

This energy efficient system incorporates the multi-access characteristics of 5G network during radio resource. But this method does not consider the offloading schemes below a certain threshold. To achieve maximum system utility by establishing a trade-off between throughput and fairness, Ran Bi et al. designed an computationally economical scheme [5].

This works on the basis of Karush-Kuhn-Tucker condition that deploys gradient-based approach. A holistic survey of MEC technology that augments the motivation and evolution of remote computing technologies is done by Mohammed Maray et al [6].

This work presents an up-to-date status quo of the concepts used in offloading mechanisms, its granularities, along with the techniques. A near-end network solution for offloading in MEC is proposed by Saranya et al [7].

The work primarily focus on reducing the latency and energy consumption and hence other QOS parameters takes a back seat. Another important work that deploys reinforcement learning is proposed by Miaojiang Chen et al. [8]

Sensing-based spectrum sharing in cognitive radio networks, 2022, X. Kang etal, This spectrum sharing model can achieve a higher capacity of SU link and improve the spectrum utilization. Also achieved the ergodic capacity of the SU link considering both transmit and interference power constraints [9]

Optimal wideband spectrum sensing framework for cognitive radio systems, 2022, P. Paysarvi-Hoseini and N. C. Beaulieu, Provided secondary transmission opportunities over multiple non overlapping narrowband channels is presented. An efficient iterative algorithm which solves the optimization problem with much lower complexity [10].

Power, sensing time, and throughput tradeoffs in cognitive radio systems: A cross-layer approach, 2018, K. Hamdi and K. B. Letaief, A cross-layer optimization problem to design the sensing time and optimize the transmit power in order to maximize the cognitive system throughput while keeping the interference to the primary user under a threshold constraint [11]**.**

5.Proposed System

The model proposed in this work mitigates latency at reduced power consumption of the edge IoT device. These devices are generally healthcare sensors, body parameter monitoring devices, scanning equipment, or any other applications associated with health domain, which demands minimum delay in data processing. The MEC servers plays two major roles: performs local computation and offloads to the core cloud for computation.

The decision of performing the computations locally or on the core cloud is done based on the offloading decision, which greatly relies on the type of task. As the application is focussed on medical domain, prioritised treatments are of primary importance to the healthcare professionals. Hence, the work uses the Ensemble Empirical Mode Decomposition (EEMD) which is further smoothened by XGBoost to classify the tasks with higher priority based on the previous experience.

The critical, computationally intensive and time sensitive medical equipment are given higher priority than the others. The labelling of the applications is done using the medical expertise of the professionals. After classifying the tasks based on its critical nature, the offloading decision is made using the novel based on the two criteria namely latency and power consumption is made by the novel Energy aware Multi Agent Deep Queue Optimisation

(E-MADQO).

Empirical Mode Decomposition (EMD) a data-adaptive technique that decomposes the signal into meaningful components. They can analyse and processing non-linear, non-stationary signals by isolating them into physical components with different resolutions This property of the of EMD make is suitable for mission sensitive applications like biomedical data analysis, bearing fault detection, seismic signal analysis and power signal analysis.

In EMD, the physical signal x(t) is decomposed into number of Intrinsic Mode Functions (IMFs) where the signal is expected to meet two criteria: firstly, the total number of extrema must be same as the number of zero crossing or the difference can be at most 1 and secondly the average of the upper and lower envelope of the signal must be is zero at all points. This empirical EMD signifies true IMF as the average of the corresponding IMF that is obtained by EMD through ensemble of trials with varied realizations of white noise added to the signal x(t).

The XGBoost algorithm is integrated with the EEMD of the non-stationary signal to prioritize the tasks based on their nature. The energy spectrum characteristics of the medical signals are decomposed into individual IMF components. An extended feature vector is formed by EEMD. The XGBoost feature selection is done by reducing the dimensions of the heterogeneous physical signals.

6.Software Components

6.1 Keras

Keras, a popularly known python library, operates atop either TensorFlow or Theano. While alternative high-level Python neural networks libraries like TF-Slim can be applied aboveTensorFlow, they are less developed. Keras simplifies TensorFlow code by utilizing a more concise code base, ensuring reduced code length and smoother processing. Keras is used for a graphical representation of the models which helps to understand the structure of the model. Auto Keras, a library based on keras, has also gained popularity and can be used to make it quicker to get results.

6.2 Pandas

Pandas is a powerful open-source data manipulation and analysis library in Python, widely used for handling structured data. It provides high-performance, easy-to-use data structures and tools for working with structured data, making it an essential tool for data scientists, analysts, and developers. The primary data structures in Pandas are Series and DataFrame. Series is a one-dimensional labeled array capable of holding data of any type, while DataFrame is a two-dimensional labeled data structure resembling a spreadsheet or SQL table. Pandas offers a wide range of functions and methods for data manipulation, including filtering, sorting, grouping, merging, and reshaping data. It also supports handling missing data and time-series data efficiently. Moreover, Pandas integrates seamlessly with other Python libraries, such as NumPy, Matplotlib, and scikit-learn, making it a valuable tool for data analysis, visualization, and machine learning tasks. Overall, Pandas simplifies the process of data manipulation and analysis in Python, enabling users to perform complex data tasks with ease and efficiency.

6.3 NumPy

NumPy is a fundamental library in Python for numerical computing that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It is an essential tool for data scientists, engineers, and researchers working with numerical data and mathematical computations. NumPy's main object is the ndarray (N-dimensional array), which is a flexible container for homogeneous data, allowing for fast operations on large datasets. The ndarray enables vectorized operations, which perform mathematical operations on entire arrays without the need for explicit looping, making computations faster and more concise.

6.2.1 Flow of system

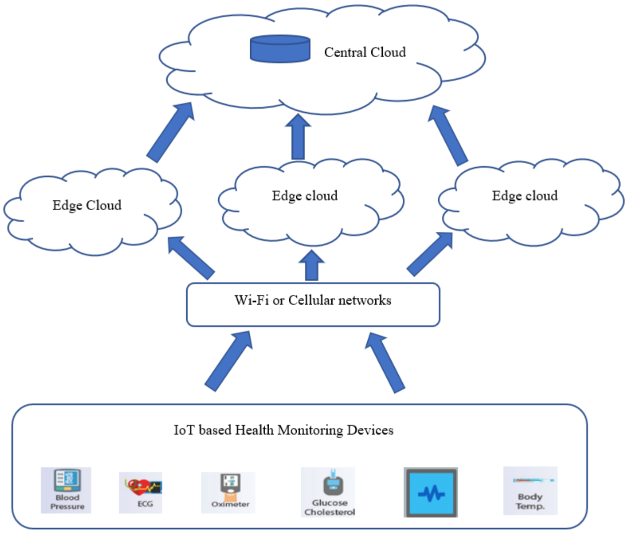


Fig 1. Flow Diagram

6.2.2 Block Diagram

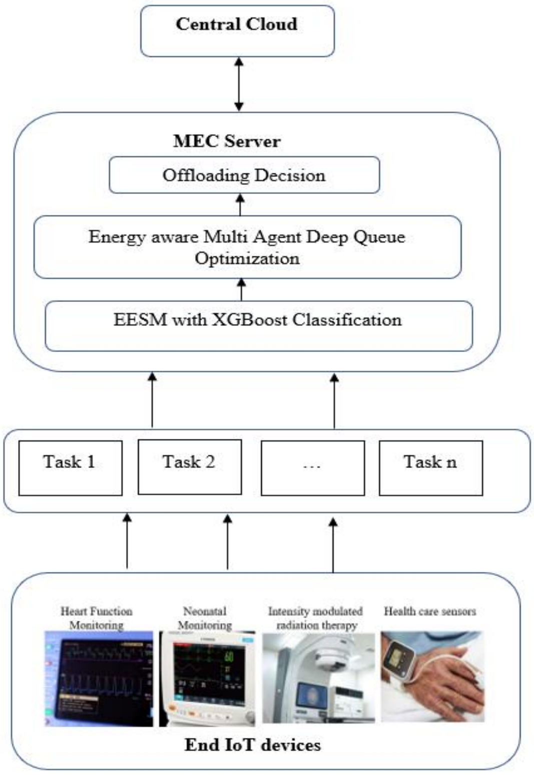

Central Cloud
MEC Server
Offloading Decision
Energy aware Multi Agent Deep Queue Optimization
EESM with XGBoost Classification
Task 1
Task 2
...
Task n
Heart Function Monitoring
Neonatal Monitoring
Intensity modulated radiation therapy
Health care sensors
End IoT devices

Fig 2. Block Diagram

7. Implementation Methodology

7.1 Ensemble Empirical Mode Decomposition smoothened

Empirical Mode Decomposition (EMD) a data-adaptive technique that decomposes the signal into meaningful components. They can analyze and processing non-linear, non-stationary signals by isolating them into physical components with different resolutions. This property of the of EMD make is suitable for mission sensitive applications like biomedical data analysis, bearing fault detection, seismic signal analysis and power signal analysis. In EMD, the physical signal x(t) is decomposed into number of Intrinsic Mode Functions (IMFs) where the signal is expected to meet two criteria: firstly, the total number of extrema must be same as the number of zero crossing or the difference can be at most 1 and secondly the average of the upper and lower envelope of the signal must be is zero at all points.

7.2 EEMD smoothened with XGBoost for prioritizing the tasks

Input: Signals from various medical devices (x(t))

Output: Prioritized tasks

EEMD

Generate white noise $x^i(t) = x(t) + w^i(t)$.

For each signal extract their IMFs for k number of modes

The mean IMF is estimated as

$$\underline{IMF_k}[t] = \frac{\sum_{i=1}^{I} IMF_i^i[t]}{I}$$

For i=1 to N do:

Train a weak base learner b_t : X□R, where X is the $\underline{IMF_k}[t]$

initialized as D_i (t)= (1/k), where k is the number of modes.

Determine the weight α_i of b_t.

Model is trained by using by Z_t is the normalization factor :

$$D_{(t+1)}(i) = \frac{D_t(i)e^{-\alpha_t y_i b_t (x_i)}}{Z_t}$$

End for

Task priority (f(x)) = $\sum_{t=0}^{T} \alpha_t h_t$

The XGBoost algorithm is integrated with the EEMD of the non-stationary signal to prioritize the tasks based on their nature. The energy spectrum characteristics of the medical signals are decomposed into individual IMF components. An extended feature vector is formed by EEMD. The XGBoost feature selection is done by reducing the dimensions of the heterogeneous physical signals. The input to the XGBoost classifier determines the various types of tasks from the medical devices. The primary objective of the EEMD-XGBoost is used to schedule the offloading tasks in edge cloud system. In this, the resource manager aids the process of scheduling the offloading tasks to decrease the entire service period. In addition to this, this improves the edge-cloud resource efficiency.

7.3 Energy aware Multi agent deep Queue Optimisation

The Energy-aware Multi-agent Deep Queue Optimization (E-MADQ) algorithm introduces a reinforcement learning (RL) approach to optimize offloading decisions in IoT environments. It employs multiple agents placed in the environment, each making decisions based on its observed energy level without prior initialization of dynamics. The Markov Decision Process (MDP) framework guides the agents through states and actions, with rewards determined by energy levels and offloading decisions. The algorithm evaluates the energy level of each device and compares it to operational thresholds, determining whether to offload tasks or not. This decision-making process iterates for each agent, ensuring that no device remains underutilized. By considering energy levels and task requirements, E-MADQ efficiently allocates tasks, maximizing system performance in energy-constrained IoT environments.

Input: Task t, Energy level of the device of task t en(t), l_e, f_e

```
Output: Offloading decision 0-do not offload, 1-offload
t□current task
en(t) □energy level of the tasks
if en(t) $\le$ $l_e$ then
  set $R_t\,(en(t)\,,a_t) = 1$
  return 0
break
else if $l_e \le\ \ en(t) \square \le \frac{f_e - l_e}{f_e + l_e}$ or t is not high priority
  set $R_t\,(en(t)\,,a_t) = 2$
  return1
break
else
  return1
break
```

This algorithm is done repeatedly for all the agents in the state space. As multiple agents spawn in the state space, they will not be left starving. Thus the proposed E-MADQ algorithm is employed to make final offloading decisions by considering the energy levels and energy requirements of the offloading task.

8.Architecture

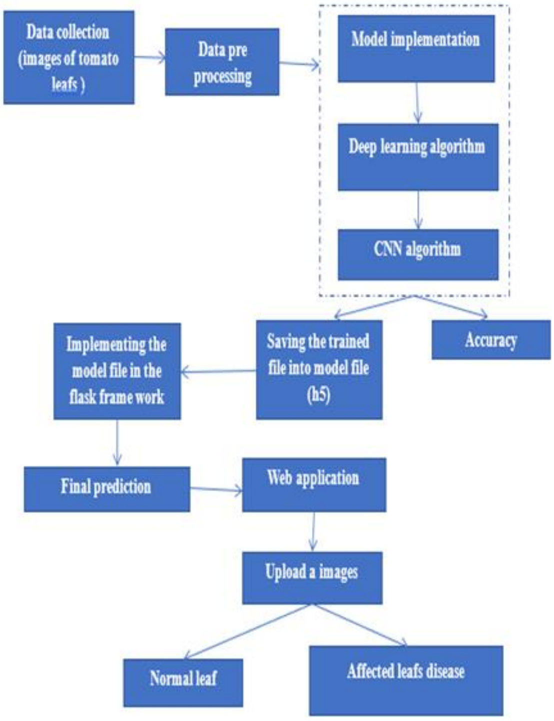


Fig 5. Architecture

9. Acquired Results

9.1 Parameter setup for the proposed methodology

Their environment has three edge network layers for cloud, terminal and heterogeneous edge layer. The experimental simulation is done with 1–4 wired edge servers along with 2–8 wireless edge servers all distributed in the edge layer. The parameter setup for the proposed methodology is given in Table 1.

Table 1: Parameter setup

Technique	Parameter	Range
EEMD	Number of iterations	10
Smoothened with XGBoost	Number of trees	100
	Number of cores	5
	Depth	4
	Number of bootstraps	50
	K for IMF	5
E-MADQ	Number of iterations	5000
	Learning rate for the agent	0.001
	Size of the minibatch	32
	Rate of reward attenuation	0.7

9.2 Setting up of simulation environment

The provided system specifications offer a comprehensive comparison between two distinct software options: CloudSim and Azure Stack Edge. CloudSim emerges as the frontrunner, boasting superior performance metrics across multiple parameters. With a MIPS rating of 1000, CloudSim sets a high standard for computational power, ensuring efficient execution of complex tasks. Moreover, its substantial RAM allocation of 2048 MB facilitates smooth multitasking and enhances overall system responsiveness. The generous storage capacity of 1,000,000 MB further solidifies CloudSim's position as a robust solution for data-intensive applications, allowing ample room for storing large datasets and files. However, CloudSim's limitation lies in its supported bandwidth, capped at 1000 MBPS. While sufficient for most typical workloads, this may pose a constraint for scenarios requiring ultra-high-speed data transfer. In contrast, Azure Stack Edge offers a more modest set of resource allocations. With a MIPS rating of 300 and

512 MB of RAM, it provides adequate computing power for basic tasks and applications. The storage capacity of 10,000 MB, though notably smaller than CloudSim's, still offers ample space for storing essential data and applications. Where Azure Stack Edge truly shines is in its supported bandwidth, which is significantly higher at 10,000 MBPS. This substantial bandwidth allocation ensures swift data transfer and seamless connectivity, making Azure Stack Edge an attractive option for applications that prioritize network speed and reliability.

Table 2: simulation environment

Software/ System	**Parameter**	**Range/ Value**
CloudSim	MIPS	1000
	RAM	2048 MB
	Storage	1,00,000 MB
	Supported bandwidth	1000 MBPS
Azure Stack Edge	MIPS	300
	RAM	512 MB
	Storage	10,000 MB
	Supported bandwidth	10,000 MB
System Specifications	OS	Windows 11
	RAM	16 GB
	Processor	Core i10
	Hard disk capacity	256 GB

10.Performance Metrics

The performance metrics for evaluating the efficiency of the proposed system are defined as follows:

- **True Positive (TP):** Legs correctly categorized as positive.
- **False Positive (FP):** Legs incorrectly categorized as positive.
- **False Negative (FN):** Legs correctly categorized as negative but identified as positive.
- **True Negative (TN):** Legs correctly categorized as negative.

Accuracy: A computation metric reflecting the system's error, calculated as the difference between potential and actual outcomes. Low accuracy arises when the machine consistently evaluates input variables with the same procedure, yielding consistent but incorrect results. The ratio of correct outcomes to the total is known as accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision:

It is a measure of random error in algebraic terms.

$$Precision = \frac{TP}{TP+FP}$$

Precision is a statistical metric used to gauge the accuracy of a classification or prediction model, particularly in binary classification scenarios. It measures the proportion of correctly predicted positive cases (true positives, TP) relative to all cases that were predicted as positive, including both true positives and false positives.

10.1Performance Comparison:

The performance of the proposed methodology is assessed on the metrics as discussed above ad its performance is compared with Collaborative Task Offloading and Resource Allocation Algorithm (CTORAA), Dynamic Energy-Efficient Load Balancing (DEELB), supervised learning-based computational offloading (DSLO) , and Dynamic Task Offloading in Mobile Edge (DTOME). The analysis of the Mean Delivery Time (MDT) of the proposed method in comparison with other methods as mentioned below is shown in Figure 6

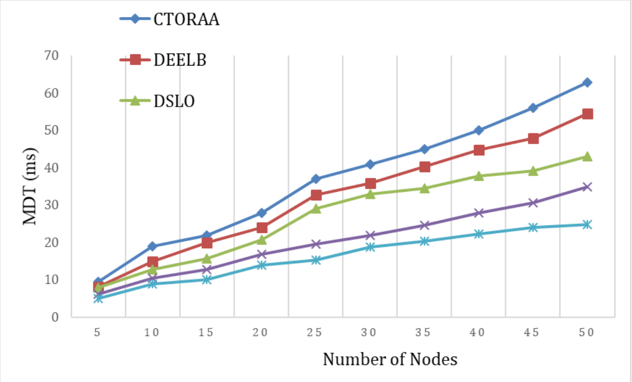


Fig 6. Comparative analysis of Mean Delivery Time

11. Existing System Performance

The comparative analysis of the proposed E-MADQ methodology highlights its superior Mean Delivery Time (MDT) performance across various nodes compared to alternative techniques. Specifically, the E-MADQ method demonstrates notable efficiency in delivering messages promptly, as evidenced by its consistently lower MDT across different node configurations. Particularly impressive is the average MDT of 16.33ms achieved by the proposed methodology, significantly outperforming competing techniques. Furthermore, as the number of nodes increases, although there is a slight decrease in delivery time, this reduction is significantly less pronounced compared to other methods. This suggests that the scalability of the E-MADQ approach is well-maintained, with delivery times remaining consistently low even under increased node loads. Such efficiency in message delivery is crucial for real-time systems where timely communication is imperative.

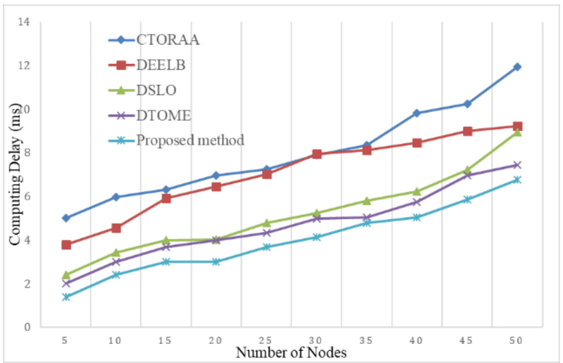
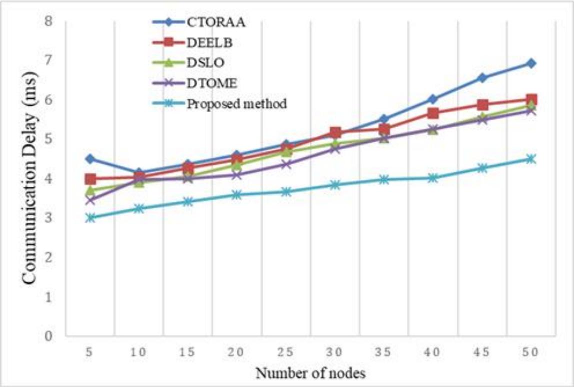


Fig7. Comparative analysis of Computational Delay

11.1 Proposed system Performance:



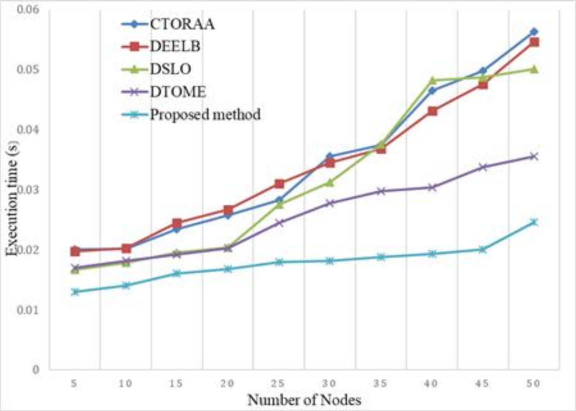


Fig 8. Communication Delay and execution time

12. Conclusion

The primary focus of this research endeavor revolves around the development and implementation of an energy-conscious offloading scheduling model tailored specifically for distributed edge-IoT environments, with a particular emphasis on time-critical applications such as medical devices. Termed as the Energy-aware Multi-Agent Deep Q-learning (E-MADQ) method, the proposed approach unfolds in two distinct phases, each contributing to its overall efficacy and functionality. Initially, the methodology undertakes the pivotal task of task classification originating from diverse IoT devices. This classification is facilitated by employing an Ensemble Empirical Mode Decomposition (EEMD) technique, further refined through the application of XGBoost, a machine learning algorithm known for its prowess in classification tasks. The aim here is to effectively segregate tasks based on their inherent characteristics and requirements, laying the groundwork for subsequent decision-making processes.

The second phase of the E-MADQ methodology involves the actual offloading decision-making process. This critical aspect leverages the power of Reinforcement Learning (RL), a paradigm well-suited for dynamic decision-making in uncertain environments. By integrating RL techniques within the offloading framework, the methodology endeavors to optimize offloading decisions based on the prevailing energy levels of the respective IoT devices. This adaptive approach ensures that offloading decisions are not only timely but also considerate of the energy constraints imposed by the devices, thereby mitigating potential resource wastage and optimizing overall system efficiency..

To delve deeper into the task classification phase, the methodology adopts a multifaceted approach. Signal processing techniques are employed to extract Intrinsic Mode Functions (IMFs) from the input signals originating from IoT devices. These IMFs, subjected to varying noise levels, undergo a process of averaging to derive meaningful features that facilitate task classification. This meticulous preprocessing step is instrumental in ensuring the robustness and accuracy of the subsequent classification process, laying the foundation for effective decision-making in the offloading phase. Accessibility is a pivotal focus, catering to farmers and agricultural practitioners with varying technical backgrounds. The system's real-time processing capability and automated tool for early detection address the urgent need for timely interventions, minimizing crop losses, and improving overall agricultural productivity.

Looking ahead, the proposed methodology presents promising avenues for future extensions and refinements. One potential area of exploration involves the implementation of separate queues for classified tasks, thereby enhancing task management and allocation efficiency. Additionally, incorporating additional feature extraction techniques and smoothening algorithms could further augment the accuracy and reliability of the task classification process, thereby enhancing the overall robustness of the offloading framework. Through continuous iteration and refinement, the proposed methodology stands poised to catalyze advancements in energy-aware offloading scheduling for distributed edge-IoT environments, with far-reaching implications for diverse application domains, including healthcare, manufacturing, and smart infrastructure.

References

[1] Song, Z., Liu, Y. and Sun, X., 2020. Joint task offloading and resource allocation for NOMA-enabled multi-access mobile edge computing. IEEE Transactions on Communications, 69(3), pp.1548-1564.

[2] Kumaran, K., Sasikala, E. (2023). An efficient task offloading and resource allocation using dynamic arithmetic optimized double deep Q-network in cloud edge platform. Peer-to-Peer Networking and Applications, 16(2), 958-979.

[3] Lin, Z., Lin, M., De Cola, T., Wang, J.B., Zhu, W.P. and Cheng, J., 2021. Supporting IoT with rate-splitting multiple access in satellite and aerial-integrated networks. IEEE Internet of Things Journal, 8(14), pp.11123-11134.

[4] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X. and Zhao, L., 2020. Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. IEEE Transactions on Industrial Informatics, 17(7), pp.4925-4934.

[5] Huang, J., Tong, Z. and Feng, Z., 2022. Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization. International Journal of Communication Systems, p.e5161.

[6] Lin, H., Zeadally, S., Chen, Z., Labiod, H. and Wang, L., 2020. A survey on computation offloading modeling for edge computing. Journal of Network and Computer Applications, 169, p.102781.

[7] Saranya, G., Sasikala, E. (2023). An efficient computational offloading framework using HAA optimization-based deep reinforcement learning in edge-based cloud computing architecture. Knowledge and Information Systems, 65(1), 409-433..

[8] Li, M., Cheng, N., Gao, J., Wang, Y., Zhao, L. and Shen, X., 2020. Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization. IEEE Transactions on Vehicular Technology, 69(3), pp.3424-3438.

[9] Qi, L., Hu, C., Zhang, X., Khosravi, M.R., Sharma, S., Pang, S. and Wang, T., 2020. Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. IEEE Transactions on Industrial Informatics, 17(6), pp.4159-4167.

[10] Asim, M., Abd El-Latif, A.A., ELAffendi, M. and Mashwani, W.K., 2022. Energy consumption and sustainable services in intelligent reflecting surface and unmanned aerial vehicles-assisted MEC system for large-scale Internet of Things devices. IEEE Transactions on Green Communications and Networking, 6(3), pp.1396-1407.

[11] Wang, J., Pan, J., Esposito, F., Calyam, P., Yang, Z. and Mohapatra, P., 2019. Edge cloud offloading algorithms: Issues, methods, and perspectives. ACM Computing Surveys (CSUR), 52(1), pp.1-23.

[12] Sharanya, S., Venkataraman, R. and Murali, G., 2022. Predicting remaining useful life of turbofan engines using degradation signal based echo state network. International Journal of Turbo & Jet-Engines.

[13] Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y. and Guo, L., 2022. Computation offloading in mobile edge computing networks: A survey. Journal of Network and Computer Applications, p.103366.

[14] Mao, Y., Zhang, J. and Letaief, K.B., 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), pp.3590-3605.

[15] Dinh, T.Q., La, Q.D., Quek, T.Q. and Shin, H., 2018. Learning for computation offloading in mobile edge computing. IEEE Transactions on Communications, 66(12), pp.6353-6367.

[16] Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S. and Zhang, Y., 2016. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE access, 4, pp.5896-5907.

[17] Kumaran, K., Sasikala, E. (2023, March). Deep Reinforcement Learning algorithms for Low Latency Edge Computing Systems. In 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP) (pp. 1-5). IEEE.

[18] Maray, M. and Shuja, J., 2022. Computation offloading in mobile cloud computing and mobile edge computing: survey, taxonomy, and open issues. Mobile Information Systems, 2022.

[19] Alam, M.G.R., Hassan, M.M., Uddin, M.Z., Almogren, A. and Fortino, G., 2019. Autonomic computation offloading in mobile edge for IoT applications. Future Generation Computer Systems, 90, pp.149-157.

[20] Chen, M., Wang, T., Zhang, S. and Liu, A., 2021. Deep reinforcement learning for computation offloading in mobile edge computing environment. Computer Communications, 175, pp.1-12.