



TBBS-TO: Trustable Blockchain and Bandwidth Sensible-based Task Offloading and Resource Allocation in Cloud-IoT Network

K. Saravanan^{1,*}, R. Santhosh¹

¹Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India

Emails: santhoshrd@gmail.com; saraengg@gmail.com

Abstract

The ability to facilitate high-performance task offloading while maintaining participant confidence is crucial, but not essential, to Cloud-Edge-Network (CEN) computing due to the geographic distribution and operation by various parties. Additionally, conflicts of interest may arise among the highly dynamic and diverse CEN members who provide resources. This study proposes a collaborative task offloading framework for CEN computing, called Trustable Block Chain and Bandwidth Sensible-based Task Offloading (TBBS-TO) and resource allocation empowered CEN. The E-PEFT consensus algorithm for block chain in task offloading optimizes resource allocation and task execution by dynamically adjusting consensus parameters based on environmental factors and performance feedback. Moreover, in our work for alleviating heterogeneous issues IoT users are mobility aware clustering is performed using Bi-directional Clustering Algorithm based on Local Density (BCALoD). In this work, block chain is essential to BC-CED's core functions, such as task delegation, resource utilization brokerage, and bandwidth sensible resource allocation. By modifying the block chain consensus procedure, TBBS-TO distinguish itself from other solutions by enabling participants to reach a consensus on task offloading. To achieve this, we formulate the offloading problem by considering both network performance and the computational capabilities of potential nodes. Using Multi-agent Double Deep Q-Network (MA-DDQN) based technique, TBBS-TO allow participants to compete for the right to produce a block by evaluating offloading policies and selecting the most effective one for the next period. Additionally, dynamically bandwidth sensible resource allocation is performed by considering significant parameters. Comprehensive testing on a commercial block chain platform has shown that TBBS-TO outperforms existing solutions in task offloading and blockchain maintenance.

Keywords: Task Offloading; Trustable Blockchain; Multi-agent Double Deep Q-Network (MA-DDQN); Bi-directional Clustering; Resource Management; Resource Allocation

1. Introduction

The rapid proliferation of Internet of Things (IoT) devices has revolutionized numerous sectors, from smart homes and cities to industrial automation and healthcare. These devices generate massive amounts of data that need to be processed, analyzed, and acted upon in real-time or near-real-time. However, the computational and storage capacities of individual IoT devices are often limited, necessitating the use of more powerful infrastructure for task processing. This brings us to the concept of task offloading and resource allocation in a Cloud-Edge-IoT environment, which plays a crucial role in enhancing the efficiency and effectiveness of IoT networks. Task offloading involves transferring computational tasks from resource-constrained IoT devices to more capable edge servers or cloud data centers. This approach leverages the strengths of various computational resources distributed across the cloud and edge layers to meet the demands of IoT applications. Resource allocation, on the other hand, is the process of efficiently distributing these computational tasks among the available resources to optimize performance metrics such as latency, energy consumption, and throughput. An effective mechanism for task offloading and resource allocation is vital for several reasons. First, it ensures that computational tasks are processed promptly, reducing latency and improving the responsiveness of IoT applications.

Second, it optimizes the use of available resources, preventing bottlenecks and ensuring that no single node is overwhelmed with tasks. Third, it can significantly reduce energy consumption by allocating tasks to nodes that can execute them most efficiently. Finally, it enhances the overall performance and reliability of the IoT network by ensuring that resources are used in the most effective manner possible. While task offloading and resource allocation offer numerous benefits, they also introduce significant security concerns in IoT networks. These concerns arise due to the inherent characteristics of IoT environments, including the heterogeneity of devices, their large-scale deployment, and the often-sensitive nature of the data they handle. Ensuring data integrity and confidentiality is paramount as sensitive data transmitted across networks during task offloading must be protected from unauthorized access and data breaches. Additionally, robust authentication and authorization mechanisms are crucial to establish trust between IoT devices and edge/cloud servers, preventing unauthorized access and manipulation. Privacy concerns also play a significant role as many IoT applications handle personal data, necessitating stringent measures to protect user data during offloading and processing. IoT networks are also susceptible to attacks. Effective detection and mitigation mechanisms are needed to safeguard against such attacks. Furthermore, offloaded tasks can become vectors for malware and intrusion, compromising the security of the entire network. Regular updates, patches, and intrusion detection systems are necessary to protect against these threats.

Deep learning has shown promise in optimizing task offloading in IoT networks due to its ability to model complex patterns and make intelligent decisions. However, there are several limitations to this approach. Learning model can be impractical for real-time IoT applications. Scalability issues also arise as the number of IoT devices and tasks increases, leading to more complex deep learning models that are challenging to manage. Data dependency is another limitation, as deep learning models require large amounts of data for training, which might not always be available in IoT environments. Additionally, the time taken to make decisions using deep learning models can introduce latency, which is critical for time-sensitive IoT applications. Continuous updates and retraining of models are required to adapt to changing conditions in the network, which can be resource-intensive. To address the aforementioned challenges and limitations, we propose. This innovative framework leverages block chain technology and advanced consensus algorithms to enhance the security, efficiency, and reliability of task offloading and resource allocation in IoT networks.

By utilizing a modified block chain consensus procedure, TBBS-TO enables secure and trustable task offloading. The E-PEFT consensus algorithm optimizes resource allocation and task execution by dynamically adjusting consensus parameters based on environmental factors and performance feedback, ensuring secure and efficient operations. TBBS-TO formulate the offloading problem by considering both network performance and the computational capabilities of potential nodes, ensuring that tasks are allocated to nodes that can execute them most efficiently, reducing latency and optimizing resource utilization. Using Multi-agent Double Deep Q-Network (MA-DDQN) techniques, TBBS-TO allow participants to compete for the right to produce a block by evaluating offloading policies and selecting the most effective one for the next period. This dynamic and competitive process ensures continuous optimization of the offloading process, adapting to changing network conditions and task requirements. Additionally, TBBS-TO incorporate a dynamically bandwidth-sensitive resource allocation mechanism that considers significant parameters to allocate resources efficiently, optimizing the use of available bandwidth and improving overall performance. To address the heterogeneity of IoT users and their mobility, TBBS-TO uses the Bi-directional Clustering Algorithm based on Local Density (BCALoD), enhancing the overall efficiency and effectiveness of resource allocation and task offloading by considering the mobility patterns of IoT devices. Extensive testing on a commercial block chain platform has demonstrated that TBBS-TO outperform existing solutions in both task offloading and block chain maintenance, enhancing the efficiency, effectiveness, and security of task offloading in Cloud-Edge-IoT networks. The TBBS-TO framework represents a significant advancement in the field of Cloud-Edge-IoT computing, offering a robust and efficient solution for task offloading and resource allocation. By leveraging block chain technology, advanced consensus algorithms, and deep learning techniques, TBBS-TO addresses the key challenges and limitations of existing solutions, providing enhanced security, efficiency, and adaptability. This innovative framework holds great promise for the future of IoT networks, paving the way for further research and development in this rapidly evolving field.

A. Research Contribution

Here are the proposed work novelties provided:

- **E-PEFT Consensus Algorithm for Blockchain in Task Offloading:** The study introduces the E-PEFT consensus algorithm, which optimizes resource allocation and task execution by dynamically adjusting consensus parameters based on environmental factors and performance feedback.
- **Mobility-Aware Clustering Using BCALoD:** To address heterogeneous issues among IoT users, the study implements a Bi-directional Clustering Algorithm based on Local Density (BCALoD), which performs mobility-aware clustering, enhancing the efficiency and effectiveness of task offloading.

- Block chain Integration for Core Functions in BC-CED: The work highlights the essential role of block chain in BC-CED's core functions, such as task delegation, resource utilization brokerage, and bandwidth-sensitive resource allocation, achieved by modifying the block chain consensus procedure.
- Multi-Agent Double Deep Q-Network (MA-DDQN) Based Offloading: The study employs a MA-DDQN-based technique that allows participants to compete for the right to produce a block by evaluating and selecting the most effective offloading policies, alongside dynamically adjusting bandwidth-sensitive resource allocation based on significant parameters.

B. Paper Organization

The subsequent sections of this essay are organized as follows: Section II reviews previous research on secure task offloading and resource allocation, discussing various methods and algorithms used in the past. It highlights the strengths and weaknesses of these approaches to provide a clear picture of the current state of the field. Section III describes the overview of the proposed work. Section IV outlines the problem statement for task offloading in secure manner. Section V here, the proposed TBBS-TO model is detailed. It introduces a new approach for performing effective and secure task offloading, explaining the framework and innovative aspects. This section includes diagrams, pseudocode, and detailed algorithms to make the methodology clear. Section VI signifies the proposed trustable block chain in TBBS-TO framework. Section VII explains how the simulations were set up for the experimental analysis, detailing the environment, parameters, performance analysis and procedures used. Section V concludes the proposed TBBS-TO.

2. Related works

In this section, we reviewed state-of-art in enhancing of cloud security which is further divided into three sub-sections which are detailed as follows:

A. Security Task Offloading in Cloud-IoT

Aazam et. al [21] presented an innovative approach to task offloading in the Cloud of Things (CoT) environment, integrating cloud, fog, and IoT technologies. The proposed method addresses sustainability by optimizing resource utilization and reducing energy consumption. The authors provide a comprehensive analysis of the offloading strategy, demonstrating its effectiveness through extensive simulations. The paper is well-structured, offering clear explanations and robust experimental results. However, real-world implementation details and potential challenges in diverse environments are areas that could be explored further. Vishwakarma et. al [22] focused on enhancing cloud data storage for healthcare applications through improved resource scheduling and robust security measures. The paper excels in highlighting the critical need for secure and efficient data management in healthcare, providing a detailed architecture and scheduling algorithm. The integration of security features is well-executed, ensuring data integrity and confidentiality. The study is thorough, with strong theoretical foundations and practical implications. Nevertheless, more empirical data from real healthcare systems would strengthen the validation of the proposed model. Cao et. al [23] introduced a delay-aware and energy-efficient scheduling algorithm for IoT tasks in cloud-fog networks, leveraging double block chain technology. The approach is novel, addressing both performance and security concerns effectively. The dual blockchain setup enhances transparency and security, while the scheduling algorithm optimizes task execution time and energy consumption. The authors provide solid theoretical analysis and simulation results. However, the complexity and scalability of the double block chain implementation in large-scale IoT networks may require further investigation.

Vijayasekaran et. al [24] presented an advanced resource scheduling framework for edge computing, integrating cloud and IoT using clustering and deep learning techniques. The proposed method improves resource allocation efficiency and system performance. The use of deep learning for predictive analytics and clustering for task grouping is well-articulated and validated through simulations. The paper is comprehensive, with clear methodologies and significant results. Future work could explore real-time implementation and the impact of varying network conditions on the proposed framework. Zhang et. al [25] proposed a trust-based framework for secure multi-cloud collaboration in cloud-fog-assisted IoT environments. The framework enhances trust management and security in multi-cloud interactions, addressing the challenges of data integrity and confidentiality. The paper is detailed, providing a clear architecture and trust evaluation mechanism. The authors support their claims with theoretical analysis and simulation results. The study is insightful, but the practical applicability in diverse IoT ecosystems and the overhead of maintaining trust scores could be further explored. Raheema et. al [26] introduced a grouping technique for secure data transfer in cloud-assisted IoT applications. The proposed method focuses on enhancing data security and transfer efficiency through intelligent grouping of IoT devices. The approach is innovative, with a strong emphasis on security protocols and data protection. The authors provide a thorough analysis and simulation results demonstrating the technique's effectiveness. However, the scalability of the grouping method and its performance in high-density IoT environments are areas that could benefit from additional research [27]. The approach leverages block chain technology to enhance security and trust in data transactions. The paper is well-structured, offering a detailed explanation of the authentication process and its advantages. The use of block chain for decentralized

authentication is well-justified, with comprehensive theoretical analysis and simulation results. The research is compelling, but practical deployment challenges and the impact of blockchain overhead on system performance warrant further exploration.

B. Secure Resource Allocation in Cloud-IoT

The authors [28] address the significant challenge of data imbalance and its impact on resource utilization and performance. The model's effectiveness is validated through extensive simulations, demonstrating improved data distribution and resource consumption rates. The study is well-structured, with clear explanations and comprehensive results. However, practical implementation in real-world IoT scenarios and scalability across different cloud providers could be further explored. Chen et. al [29] presented a dynamic task allocation and service migration framework for edge-cloud IoT systems, leveraging deep reinforcement learning (DRL). The paper effectively addresses the complexities of task scheduling and service migration in a hybrid edge-cloud environment. The use of DRL enhances adaptability and decision-making efficiency, as demonstrated by the robust simulation results. The study is well-executed, with detailed methodologies and significant contributions to the field. Future work could focus on real-time implementation challenges and the impact of varying network conditions on DRL performance. Zolfaghari et. al [30] delved into the dichotomy of cloud and IoT, specifically examining cloud-assisted IoT systems from a security perspective. The authors provide a comprehensive overview of the security challenges and potential threats inherent in cloud-IoT integration. The study offers insightful analysis and proposes effective security measures to mitigate identified risks. The paper is well-researched, with clear articulation of issues and proposed solutions. However, empirical validation of the proposed security measures and their practical applicability in diverse IoT environments could enhance the study. Vishwakarma et. al [31] focused on enhancing cloud data storage for healthcare applications through improved resource scheduling and robust security measures. The paper excels in highlighting the critical need for secure and efficient data management in healthcare, providing a detailed architecture and scheduling algorithm. The integration of security features is well-executed, ensuring data integrity and confidentiality. The study is thorough, with strong theoretical foundations and practical implications. Nevertheless, more empirical data from real healthcare systems would strengthen the validation of the proposed model.

The authors [32] effectively address the complexities of resource management in CPES, proposing a model that enhances efficiency and performance. The study provides a thorough analysis of the proposed framework, supported by simulation results that demonstrate significant improvements. The research is well-detailed and impactful, though real-world validation and scalability across different energy systems could be further investigated [33]. By employing a decision tree-based multi-objective automated approach, the authors enhance the adaptability and efficiency of resource allocation processes. The proposed model shows significant potential in handling complex resource allocation scenarios, offering a scalable solution for futuristic cloud computing environments. The research is well-founded, with strong theoretical underpinnings and practical implications. However, the complexity of the decision tree algorithms may require careful consideration to ensure ease of implementation and maintenance. Tay et. al [34] explored various resource allocation algorithms in the context of edge, fog, and cloud computing environments. The authors provide a detailed comparative analysis of different algorithms, highlighting their strengths and weaknesses in various scenarios. The study offers valuable insights into the optimization of resource allocation across the computing continuum. The paper is well-structured, with thorough evaluations and clear presentations. However, real-world case studies and empirical data would further strengthen the findings and applicability of the reviewed algorithms.

C. Clustering and Secure Task Offloading in Cloud-IoT

Chowdhary et. al [35] presented a task clustering-based Quality of Service (QoS) aware scheduling algorithm tailored for task execution in a cloud-IoT model, specifically targeting education services. The authors effectively address the unique demands of educational applications, such as high reliability and low latency. The proposed algorithm optimizes resource allocation and enhances QoS by clustering similar tasks, thereby improving efficiency and performance. The study is well-structured with clear methodologies and significant simulation results. However, real-world implementation and the adaptability of the algorithm in different educational environments could be explored further. Bu et. al [36] provided a comprehensive review of task scheduling in the Internet of Things (IoT), outlining the key challenges, current solutions, and future trends. The authors cover a wide range of issues, including resource constraints, dynamic environments, and security concerns. The review is thorough, offering detailed discussions on various scheduling algorithms and techniques. The paper is well-organized and informative, making it a valuable resource for researchers and practitioners. Future work could delve deeper into emerging technologies and their potential impact on IoT task scheduling. Fazel et. al [37] explored the potential of mist computing in IoT networks, focusing on the use of clustering techniques to enhance performance and efficiency. Mist computing, positioned between edge and fog computing, offers unique advantages in terms of latency and resource management. The paper presents innovative clustering methods that optimize task execution and resource allocation in mist computing environments. The study is insightful, with clear explanations and robust simulation results. However, practical deployment challenges and the integration of mist computing with existing infrastructures warrant further investigation.

Su et. al [38] introduced a prediction-based resource deployment and task scheduling framework for edge-cloud collaborative computing environments. The authors leverage predictive analytics to anticipate resource demands and optimize task scheduling, enhancing overall system performance and efficiency. The proposed framework is validated through extensive simulations, demonstrating significant improvements in resource utilization and task execution times. The study is well-executed, with detailed methodologies and comprehensive results. Future research could focus on real-time implementation and the impact of unpredictable workloads on the prediction accuracy. Mahapatra et. al [39] presented an energy-aware task offloading and load balancing framework designed for latency-sensitive IoT applications within the fog-cloud continuum. The authors address the critical issue of energy consumption while ensuring low-latency task execution. The proposed framework intelligently offloads tasks and balances the load between fog and cloud resources, optimizing both energy efficiency and performance. The study is thorough, with clear methodologies and significant simulation results. However, real-world validation and scalability across diverse IoT environments are areas that could be further explored. Yadav et. al [40] introduced a clustering-based energy-efficient task offloading approach for sustainable fog computing. The authors propose innovative clustering techniques that enhance energy efficiency by grouping similar tasks and optimizing their offloading to fog nodes. The study effectively addresses the dual goals of sustainability and performance, providing a detailed analysis and robust simulation results. The paper is well-structured and insightful, offering valuable contributions to the field. Future work could investigate the real-world applicability of the proposed techniques and their performance under varying network conditions. These reviews provide a concise evaluation of each paper, highlighting their strengths and suggesting areas for future research.

3. Methodology

This division represents the framework of TBBS-TO, a trustable block chain and bandwidth sensible-based task offloading and resource allocation. The notifications utilized in the problem formulation are listed in Table II. The proposed structure of the TBBS-TO be illustrated in Figure 1, consisting primarily of three layers. Network applications operating on top of the TBBS-TO fall under the applications layer. These applications, functioning as TBBS-TO clients, submit resource requests and assign tasks to the TBBS-TO computation units. The block chain layer is crucial for the operation of TBBS-TO, impacting decisions on resource management, incentives, and job offloading. This layer also implements the enhanced consensus mechanism, proof-of-optimum, which optimizes the TBBS-TO work offloading schemes to allow block chain to compete for the right to produce output blocks. In addition to leveraging block chain technology, TBBS-TO aims to provide high-performance, decentralized, and reliable task offloading. The resource layer monitors the computation units that supply resources to support TBBS-TO task offloading. It is assumed

Subscriber: Network applications or clients indicate their resource requirements and quality of service expectations, delegating their tasks to the TBBS-TO.

Transaction: Task brokerage is initiated when a customer submits a resource request in TBBS-TO. This transaction includes the resource and task QoS requirements. Details of the computing units assigned to this task are entered into the transaction once the decision to offload the task is made. The payment for completing the task offloading is recorded in the transaction after the computing unit processes the task and provides the results to the client.

Blockchain Worker: Blockchain employees process these transactions. Similar to current blockchain operations, workers compete to complete a predetermined task to produce the block.

Optimizer: Using a reinforcement learning (RL) model, workers chosen from the block chain members create the task offloading policy. Optimizers compare the outcomes and select the optimizer with the best performance approximation as the leader for the next time slot. The block reward is given to the leader who outputs the block.

Computation Unit: Organizations that provide resources to handle the tasks delegated to them.

Brokerage: This organization facilitates communication between subscribers and the block chain and computation units (CUs). The TBBS-TO brokerage acts as a channel for sending tasks and CU data to the block chain. Its primary responsibilities include resource management, work offloading, and blockchain-based incentive deployment. The brokerage can either be a trust component provided by the CEN operator or be distributed operated by multiple entities, such as block chain workers.

Reward: Subscribers must pay for the resource consumption cost associated with task offloading in TBBS-TO. The collected money is distributed to the organizations supporting TBBS-TO, specifically to the block chain workers and computation units. Block chain workers receive compensation for deciding which jobs to offload and for tracking transactions, while computation units receive compensation for allocating resources to handle the offloaded tasks.

3.1. Problem formulation

We tackle the task offloading issues in TBBS-TO in this division. We summarized the system approach to commence that comprises the communication and computation models. Then, we surrounded the task offloading issues as a partially observable Markov decision process (POMDP) and observed the possible solution.

(i) Communication Model

Before the communication approach is suggested, we create the subsequent conjecture to TBBS- TO such as (i) a subscriber utilize the wireless networks and the backbone to contact the computational resource of cloud. (ii) Clients utilize the base stations as a relay to contact the subtract resources from the mobile devices inside the same cell. (iii) Edge computing servers straightly connect the base station and the resources are only applicable through the wireless link. (iv) $N = \{1, 2, \dots, N\}$ Signifies the slotted illustration of the system time. The mechanism will alter among the slots and resides constant within every one. CEN topology able to form as an undirected graph (N, L) , here N, L signifies the set of nodes and links. Assume that N_S, N_C, N_E, N_M and N_R signifies the subscriber, clouds, edge servers, MDs and routers. Assume that $\mathbb{X}(a, b)\theta\varphi$ signifies the network link among node $(a, b)\theta N$. We shorten $\mathbb{X}(a, b)$ to \mathbb{X} to alleviate the report. To $\mathbb{X}\theta\varphi$, $m_{\mathbb{X}}(Z)$ signifies accessible bandwidth at time slot Z . $m_{\mathbb{X}}(Z)$ Signifies a casual variable of Z contained by the interval $[0, m_{\mathbb{X}}^{max}]$, here $m_{\mathbb{X}}^{max}$ signifies the capacity of Z .

Assume that \mathcal{Q}_e and $\mathcal{R}_e(Z)$ signifies the quantity of the raw information which is necessary to be distributed while task offloading e contained by Z . We consequently have $\mathcal{Q}_e = \sum_{Z=Z_0}^{Z'} \mathcal{R}_e(Z) \nabla Z$, here Z_0, Z' signifies the commence and absolute time slot of the data transmission. ∇Z is the slot time length Z . And assume that $\mathcal{R}|_e$ signifies the average transmission interruption, then we have $\mathcal{Q}_e = (Z' - Z_0) \nabla Z \mathcal{R}|_e$. Therefore, the transmission interruption $(Z' - Z_0) \nabla Z$ equals to the $\mathcal{Q}_e / \mathcal{R}|_e$. Then, a higher average transmission rate diminishes the downloading time of raw information. The entire data over \mathbb{X} that should be inside bandwidth constriction throughout Z .

$$\sum_{e \in \mathbb{G}_{\mathbb{X}}(Z)} \mathcal{R}_e(Z) < m_{\mathbb{X}}(Z)$$

Here $\mathbb{G}_{\mathbb{X}}(Z)$ signifies the collection of tasks by employing link \mathbb{X} . The mentioned restrictions will be commonly explained by a virtual queue approach with length $m_{\mathbb{X}}(Z)$ as indicated in Fig. This approach point out the accumulation data expecting to be transmitted at Z . The appearance rate of the queue is the available bandwidth $m_{\mathbb{X}}(Z)$. The vibrant of $m_{\mathbb{X}}(Z)$ is

$$m_{\mathbb{X}}(Z + 1) = m_{\mathbb{X}}(Z) + \sum_{e \in \mathbb{G}_{\mathbb{X}}(Z)} \mathcal{R}_e(Z) - m_{\mathbb{X}}(Z)$$

To stay away from the blocking at Z , following discrimination should hold;

$$\frac{1}{N} \sum_{Z=1}^N \sum_{e \in \mathbb{G}_{\mathbb{X}}(Z)} \mathcal{R}_e(Z) < \frac{1}{N} \sum_{Z=1}^N m_{\mathbb{X}}(Z)$$

Here $m_{\mathbb{X}}(Z)$ is constant.

(ii) Computational Model

Remember that $\mathcal{X}_w, w = i, j, k$ are the CUs in TBBS- TO, their obtainable computation resources signified by a time dependent variable $\mathbb{D}_p(Z)$. We calculate the computation ability of cloud and edge computing servers is steady therefore these profitable platforms are committed to task offloading. Here $\mathbb{D}_p(Z), \mathbb{P}\theta\mathcal{X}_w, w = i, j$ is a stable value. The resources offered will be a time difference for MDs so they will only supply their redundant resources that are self-motivated. Therefore, $\mathbb{D}_p(Z), \mathbb{P}\theta\mathcal{X}_y$ is as common variable of Z . Here we have

$$\mathbb{D}_p(Z) \gg \mathbb{D}_s(Z) \gg \mathbb{D}_0(Z) \gg \mathbb{P}\theta\mathcal{X}_i, \mathbb{S}\theta\mathcal{X}_j, \mathbb{O}\theta\mathcal{X}_y$$

This difference continues due to the cloud computing is related on the most effective data centers, that have a large amount of resources in relation to edge and device. Edge computing servers can nevertheless handle offloading tasks better than MDs even with their restricted resources. To every task e produced at time Z , we describe a dual tuple $\{T_e(Z), \mathcal{Q}_e\}$. Here $T_e(Z)$ signifies the computational resources involved by processing e and we remember \mathcal{Q}_e is the size of raw information which is necessary to be delivered the CUs. Supposing computation unit commences to accomplish e while established all the raw data of e , the task processing holdup of e assigned to \mathbb{P} that will be abstract of rescue latency and implementation time. We signifies the accumulation of tasks coming up to be processed as \mathbb{P} as $\mathbb{L}_p(Z)$. The line up input is the total computation resource demand of tasks arrived during Z and the opposite of line up rate equals $\mathbb{D}_p(Z)$. Therefore, the self-motivated of $\mathbb{L}_p(Z)$ portrayed as;

$$\mathbb{L}_{\mathbb{P}}(Z + 1) = \mathbb{L}_{\mathbb{P}}(Z) + \sum_{e \in \mathbb{B}_{\mathbb{P}}(Z)} T_e(Z) - \mathbb{D}_{\mathbb{P}}(Z)$$

Here $\mathbb{B}_{\mathbb{P}}(Z)$ signifies the collection of tasks allocated to \mathbb{P} . We will be evaluating the total processing time of a task by

$$\mathbb{W}_{e,\mathbb{P}}(Z) = \frac{(\mathbb{L}_{\mathbb{P}}(Z)) + T_e(Z)}{\mathbb{D}_{\mathbb{P}}(Z)}$$

(iii) Problem Formulation

The primary goal of task offloading is to reduce the total execution time of tasks and minimize resource utilization overhead. The execution delay for each job e is defined as the sum of its processing and transmission times.

$$\mathbb{E}_e(Z) = \mathbb{W}_{e,\mathbb{P}}(Z) + \frac{\mathbb{Q}_e}{\mathfrak{x}_e}$$

In CEN, Computing Units (CUs) are often managed by third parties and are compensated for their services. The rate for utilizing unit resources for each computing unit \mathbb{P} is denoted as $\mathfrak{R}_{\mathbb{P}}$, and the revenue for processing job e at unit \mathbb{P} can be determined as $\mathfrak{R}_{\mathbb{P}}T_e(Z)$. By integrating execution latency and resource usage, we can define the expense function of project execution as follows:

$$\varphi_e \mathbb{E}_e(Z) + \mathbb{Q}_e \mathfrak{R}_{\mathbb{P}} T_e(Z)$$

where the weights are denoted by $\varphi_e \in [0,1]$ and $\mathbb{Q}_e \in [0,1]$. Assuming the first term execution delay has a dimension of seconds, we can eliminate this dimension by setting the dimension of φ_e to 1/seconds. Thus, we phrase the problem of task offloading as follows:

$$\min \sum_{Z=1}^T \sum_{e \in \mathcal{G}(Z)} (\varphi_e \mathbb{E}_e(Z) + \mathbb{Q}_e \mathfrak{R}_{\mathbb{P}} T_e(Z))$$

$$\text{C1: } \mathcal{X}_{\ell}(Z) \text{ is stable, } \forall \ell \in \omega$$

Where $\mathcal{G}(Z)$ represents the set of tasks created at time (Z) , and the goal of (7) is to minimize the cost function for all jobs generated during the system's operation. The constraint C1 specifies that network congestion should be avoided when selecting tasks to offload. Additionally, an online solution to this problem can be achieved by solving the unconstrained optimization problem at each time (Z) .

$$\min \mathcal{X} \sum_{e \in \mathcal{G}(Z)} (\varphi_e \mathbb{E}_e(Z) + \mathbb{Q}_e \mathfrak{R}_{\mathbb{P}} T_e(Z)) + \sum_{\ell \in \omega} \mathcal{X}_{\ell}(Z) \left(\sum_{e \in \mathbb{B}_{\ell}(Z)} \mathfrak{x}_e(Z) - i_{\ell}(Z) \right)$$

Here \mathcal{X} represents the penalty to the first term and is a nonnegative parameter. This issue is complicated by several factors: First, the second term's pair involves the judgments $\mathfrak{x}_e(Z)$ of computation units (CUs) that share the same connection. Under such circumstances, optimizing $\mathfrak{x}_e(Z)$ to minimize the total delay is challenging because it requires considering decisions made by other CUs within the same connection, in addition to e 's cost function. Second, obtaining a complete perspective of the system dynamics is typically difficult due to the intricate network state. For example, multiple network operators may run the BC-CED underlying network. The connection conditions required in (9) cannot be directly observed by the BC-CED participants. Specifically, the coordinator finds it challenging to determine $\mathcal{X}_{\ell}(Z)$ for each connection ℓ . Instead, we are limited to estimating the end-to-end delay $D_e(Z)$ for any given subscriber and CU pair,

$$D_e(Z) \approx \sum_{e \in L_e(Z)} \frac{\mathcal{X}_{\ell}(Z)}{i_{\ell}(Z)}$$

Here the set of links used by e is denoted by $L_e(Z)$

3.2. TBBS-TO FRAMEWORK

In this study, we primarily focused on secure offloading by adapting blockchain and multi-agent RL techniques. Figure 1 represents the overall architecture of proposed TBBS-TO framework.

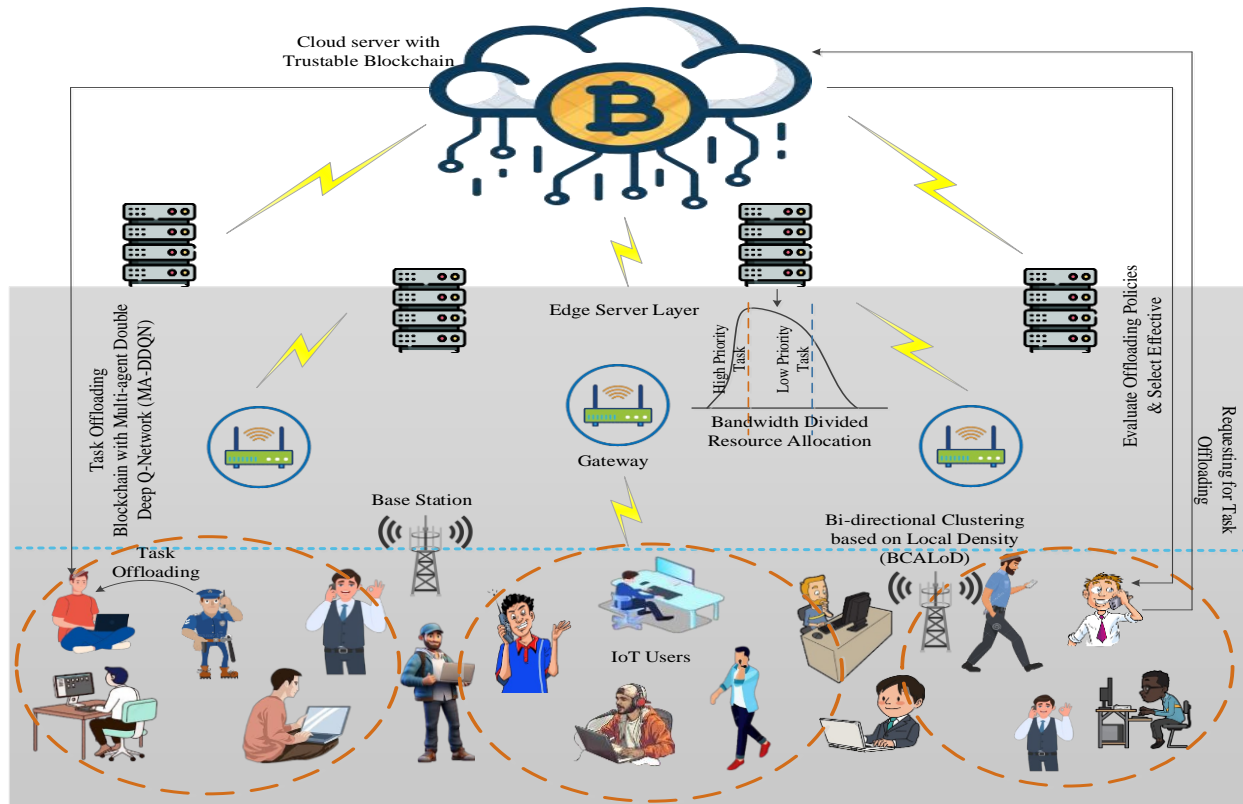


Figure.1 Overall Architecture of Proposed TBBS-TO Framework

A. Bi-directional Clustering

After the network construction, users are clustered to ensure power management and energy efficiency of the IoT users. The clustering is performed using a Bi-directional Clustering Algorithm based on Local Density (BCALoD) as shown in figure 2, which clusters in both upward and downward directions. This algorithm considers metrics such as geographical location, distance, local density, link stability, energy, and mobility. By employing this clustering method, network lifetime, throughput, and energy efficiency are improved. To further enhance energy efficiency, the cluster head is selected based on metrics including distance, head count, coverage, throughput, feedback, and packet delivery ratio. Nodes in each cluster are ranked according to these metrics, and then a super cluster head (super CH) and a sub cluster head (sub CH) are selected. If the energy of the super cluster head is depleted, the sub cluster head takes over as the cluster head. An invigilator agent monitors the clustering process, and if both the super cluster head and sub cluster head are depleted of energy or if node disconnectivity reaches a certain limit, re-clustering is initiated to maintain clustering performance. Assume that the data points with the highest local density are the clustering centres. Next, calculate the local density ($\mathcal{Q}i$) for each point in the dataset.

$$\mathcal{Q}i = \sum_{j \neq i} e^{-\left(\frac{d_{ij}}{d_c}\right)^2}$$

The cutoff distance d_c is chosen to be between 1% and 2% of the total distance between all data points, with d_{ij} representing the Euclidean distance between points i and j . Given that the local density $\mathcal{Q}^{8,14}$ is represented by a Gaussian kernel, it is unlikely that the local density is uniform throughout the region. Data points are ranked according to their local density \mathcal{Q} , under the assumption that each point has a unique local density. The BCALoD schematic design is shown in Figure 1. In the upward process, no point within the cutoff distance has a higher local density than point l . This criterion is used to identify point l as a cluster centre. The selection procedure for the up process starts at the location where the local density, \mathcal{Q} is the lowest. The local density of the cluster centres remains constant, with the addressing sequence only influencing the path from bottom to top. Small clusters are merged into a single category if no point within the cutoff distance has a higher local density. Essentially, the up process can be considered the inverse operation of the density peak clustering algorithm, which clusters data points. Algorithms 1 and 2 present the pseudo-code for the up process and down process, respectively.

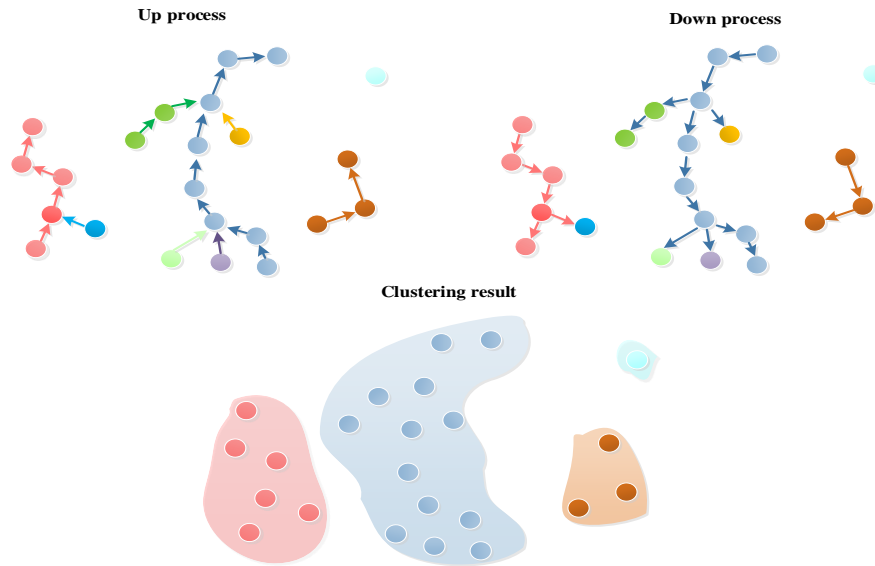


Figure 2. BCALoD based Clustering

When large clusters have significantly more data than small clusters, the quick search and find of density peaks can lead to subjective determination of the number of clusters, as small clusters' information is easily overshadowed. BCALoD effectively clusters small clusters. The mean shift algorithm's need to incorporate random data points and continually cycle the sliding window increases computational costs and complicates the analysis of high-dimensional clusters. The proposed method of constructing data clusters through data chains ensures that the selection of clustering centres is independent of the initial selection, while also reducing operational costs.

B. Trustable Blockchain based Task Offloading

Effective resource management and task offloading are crucial for both performance and energy efficiency in an IoT cloud-edge network. To achieve this, we employ a reliable blockchain combined with a Multi-Agent Double Deep Q-Network (MAD-DQN) to ensure secure and efficient task offloading. Consider a cloud-edge network comprising M edge servers, N IoT devices, and a centralized cloud server. Each IoT device generates tasks that need to be offloaded to either the cloud or an edge server. Utilizing blockchain technology, we ensure trust and security while minimizing the overall cost, which includes execution time and energy consumption. The state, action and reward are represented as follows,

State: The state s_t at time t for every IoT devices includes the current computational load L_t , available bandwidth B_t , energy level E_t , task arrival rate λ_t , and trust level T_t derived from trustable blockchain. Thus, $s_t = (L_t, B_t, E_t, \lambda_t, T_t)$.

Action: The action a_t for every IoT device involves choosing an offloading destination; local processing, offloading to an edge server $e \in \{1, 2, \dots, M\}$, or offloading to the cloud. Thus, $a_t \in \{1, 2, \dots, M + 1\}$.

Reward: Reward r_t is applied to reflect the performance of the task offloading decision, considering the execution time, energy consumption and trust level. Then, the reward function is described as,

$$r_t = (\alpha \cdot T_t, \beta \cdot E_t, \gamma \cdot \text{Execution time}_t)$$

Where α, β , and γ are weighting factors.

Each IoT device utilizes a D-DQN to evaluate the action-value function $Q(s_t, a_t)$. D-DQN consists of two neural network: the primary network $Q_\theta(s, a)$ and the target network $Q_{\theta'}(s, a)$. The update rule for the primary network is,

$$Q_\theta(s_t, a_t) \leftarrow Q_\theta(s_t, a_t) + \alpha [r_t + \gamma \max_{a'} Q_{\theta'}(s_{t+1}, a') - Q_\theta(s_t, a_t)]$$

Target network parameters θ' are periodically updated to match the primary network parameters θ . Every IoT device functions as an individual agent in a multi-agent environment. A blockchain ensures security and trust by facilitating coordination among these agents. The blockchain maintains a distributed ledger accessible to all agents, recording trust levels and offloading activities. The integrity and reliability of offloading choices are ensured by the blockchain. Each block contains the decision to offload, the degree of trust, and a digital signature with a timestamp. The trust level T_t of each IoT device is updated according to the blockchain's consensus mechanism, ensuring safe and dependable task offloading. Set random weights for the target network, $Q_{\theta'}$, and the

primary network, Q_θ . Set up the blockchain's ledger initially. Set the state initialization for every IoT device s_θ . Observe each IoT device's state s_t for each time step t . Then, using an epsilon-greedy policy based on $Q_\theta(s_t, a_t)$, choose an action a_t . Perform the chosen action a_t , calculate the reward r_t according to the energy usage, trust level, and execution time, and log the action and trust level in the blockchain.

C. Resource Management

The computation resources of users in TBBS-TO can be geo-distributed and provided randomly. Therefore, managing the resources to update the status of users promptly is crucial for the performance of task offloading. Furthermore, the dynamic bandwidth allocation uses the Improved Fuzzy Bankruptcy Game Theory (IFBGT) with MTD mechanism approach, where user act as players proposing strategies based on various metrics, ensuring faultless, flexible resource allocation that prioritizes high-priority packets first in secure manner. When users j sends a join request to the brokerage during time t , the users registration contract is invoked to generate an ID and record resource information such as CPU frequency and memory size. The blockchain maintains this information and records it in the blocks generated within t to ensure trustworthy resource provision. Once the users join the TBBS-TO, it updates the available resources at each t to ensure that the task offloading scheme can always make decisions based on the latest CEN resource status. Thus, any optimizer in TBBS-TO can access the resource state by accessing the user's information in the block. When a user's needs to leave the CED system, a quit notification is sent to the TBBS-TO brokerage. User's registration contract will add new information to the blocks that set the available resources of the users to zero.

3.3. Trustable framework

In this section, we describe the details of TBBS-TO. Smart contract is employed upon trustable block chain for executing secure task offloading and resource management. A trustable blockchain in task offloading ensures secure and transparent execution of computational tasks by leveraging decentralized and immutable ledgers. This technology mitigates the risk of fraud and unauthorized alterations, providing a reliable environment for verifying task completion and data integrity. Additionally, the use of smart contracts automates and enforces agreements, enhancing trust and efficiency in the task offloading process.

Smart Contracts in TBBS-TO: In TBBS-TO, blockchain play a vital role in CEN task offloading. Specifically, the blockchain of TBBS-TO can implement the following functions: (i) Maintain the details regarding resource providers in TBBS-TO; (ii) Validate and record the details of task offloading; and (iii) Accomplish the resource allocation in TBBS-TO.

- 1) CU registration agreement: this contract generates the CU's worldwide unique identity, or "public key," which may be utilized to encrypt messages delivered to the relevant MD for safe communication. After CU triggers this smart agreement, it also updates the blockchain with CU's resource status on a regular basis.
- 2) Job offloading contract: This contract will be triggered to construct a transaction that records the job details, including resource and QoS needs, etc., when the client makes their task offloading request to the brokerage of BC-CED. The task information is specifically specified by a 6-tuple $S = \{\mathfrak{X}ID, re_{req}, Dur, reward(\mathfrak{X}ID), Quality(\mathfrak{X}ID)\}$, where $\mathfrak{X}ID$ is the job's unique ID. $Curr(\mathfrak{X}ID)$ is the incentive for completing the tasks, $Sequality(\mathfrak{X}ID)$ describes the QoS need, such as time constraint of task processing, and re_{req} specifies the detail of resource demand, such as CPU frequency and storage capacity. To signal if the job has been completed, use $Conf(\mathfrak{X}ID)$.
- 3) Task offloading agreement: Every t , the CUs get tasks from the BC-CED brokerage by the task offloading plan produced by the consensus process. Each job's assigned CUs will be detailed in the transaction for that task.
- 4) Rewarding contract: the rewarding contract will determine whether to deliver the reward to the CU in accordance with the BC-CED incentive mechanism when the CU completes the job or meets the resource provision deadline.

4. Numerical results and discussion

We placed our TBBS-TO prototype into follow. Initially, we employed our block chain framework by utilizing the Hangzhou Blockchain Technology Research Institute's BRPOP stage. A block chain related stage known as BRPOP assists efficient and genuine relationship among many stakeholders. Moreover, BRPOP helps assorted consensus approaches and creating the perfect material for appraising POO's efficiency. 4 workers create our blockchain prototype and they are employed on a server called Intel i7-11700k, 8 cores, 1.7GHz, 32GB. We added the block ID and the time trample which was produced, the number of transaction and the transaction information are explored in section IV in every block. Every block surrounds 10,000 transactions unless otherwise specified. The consensus protocol can be utilized to concur on a block and include to the block chain in every 3 seconds. A CNN technique is implemented as the model in the RL approach. The original convolutional layer is pursued by a max pooling layer. Convolutional layers contain out channels which are 25,481, 64, 256 and 16 in order. There are nine, two, seven, three, four kernel

size strata. The most recent outputs prospect distributions for the action and fully connected. The hyper-parameter that contains the weight parameters for the efficacy function and learning rate is changed to imitate the difference among the workers. We also carried out into 5 distinct RL approaches such as Deep Q network (DQN), Actor-Critic Network (AC), Asynchronous Advantage Actor-Critic (A3C), Multi-agent DQN (MADQN) with proposed Multi-agent DDQN (MADDQN).

Deep Q network (DQN): Every client has dual networks installed. Target_{net} Remains the pathway of a copy of Eval_{net}, that is employed to evaluate the state action assessment. A collection of memory barriers provisions the final 200 behaviors. There is a batch size of 32. There are 5 iterations dividing Target_{net} and Eval_{net}.

Actor-Critic Network (AC): An AC network is employed. The detractor outputs the current state value function by employing the TD approach. By implementing the detractor output as a source, reduces the benefits of the actions to offer the action allocation.

Asynchronous Advantage Actor-Critic (A3C): It enlarged the AC functionality by including the parallel processing. The learning agent appeals to four threads that work collectively to modernize a sharded approach which combines the training data.

Multi-agent DQN (MADQN): Every four learning agents in this approach separately produce the policy.

Then, our formerly created mutual video Trans coding technology can utilize the offloading approach produced from the blockchain to constrain the Trans coding work allowance to additional test offloading concert. On this proposal, each connected device will create resources for video Trans coding. Mobile devices conclude that CU to the offload and create suggestions for video Trans coding. Then, video information will be passed to the intended CU. MD will accept the arrival pleasant leading end of the Trans coding method. This proposal includes 4 laptops such as Intel i7-7700k, 4 cores 4.25GHz/16GB, AMD Ryzen 5 Quad Core 3.2GHz/16GB, Intel i7-10750H, 6 cores 2.6GHz/16GB, Intel i7-10510U, 4 cores 1.8 Ghz/16GB performing as the MD and subscribers, and one server is performing as the cloud as Intel Xeon Platinum 8163, 8 cores 2.5Ghz/64GB.

Trans coding material is FFmpeg when the operating system is CentOS 7. Each node updates a resource monitor to identify changes in resources. The wired link has a 100Mbps bandwidth. The Trans coded video substance extent is signified by the random variable $U[2, 10]$ s. The real video Trans coded bitrate set is 1080p (60fps), 720p (60fps), 1080p (30fps), 720p (30fps), and 480p (30fps). At every time slot, the subscriber offers a common amount of behavior with unreliable computational resource necessities and video sizes. The amount of tasks produced a Poisson distribution with $\lambda 1$ values. An identical distribution with $U[5, 20]$ is experimental for the amount of time slots which divided to dual consecutive tasks that produce while $U[a, b]$ signifies an identical distribution across the range $[a, b]$.

A. Performance Analysis of RL Models

To validate the overview of our introduced task offloading mechanism, we initially determine the performance of five RL models including as DQN, AC, A3C, and MADQN, while solving this issue. The evaluated metrics involve loss variance via training and delay of unit processing during testing.

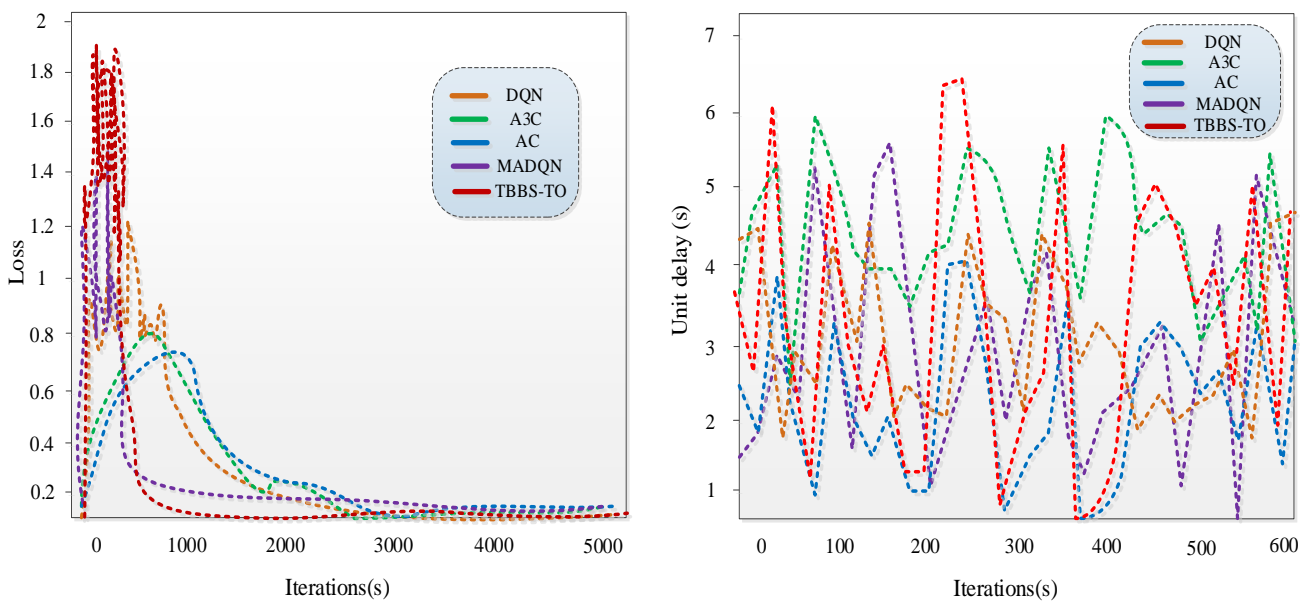


Figure 3. Performance of RL Algorithm (a) Loss convergence while training and (b) Reward variation while testing

Loss Variance: Figure 3 (a) illustrates the value of loss function while training. As per the figures, entire existing models experienced slow trend minimizing whereas rapid minimizing trend and then arrived the constant stage, depicting that the algorithm is congregated. This effect arises from the learning agent's initial lack of knowledge about the system dynamics, prompting it to estimate the values of actions in various states to improve policies. The curves stabilize once the agent's policy converges after learning the value function, or Q-value. The convergence of all curves around the 2500s indicates that their performance in work offloading issues is comparable. Notably, the AC-based approach initially shows an increasing trend before declining around the late 900s, while DQN and MADQN continue to increase as training progresses. Therefore, DQN and MADQN are far better than other models for job offloading when training time is limited. However, due to insufficient parameter tuning effectiveness of task offloading is minimized. In contrast, our proposed TBBS-TO adapt MA-DDQN with optimization algorithm which aids for enhancing task offloading performance.

Unit Delay: The latency of processing a task's unit of raw data is referred to as the UD. We use datasets of 600s to assess the UD of various solutions, and the results are displayed in Fig. 3(b). As shown in the image, A3C outperforms the other options in most cases due to its minimal delay. A3C's advantage lies in the AC model's ability to consistently seek the activity more effectively than the average. Additionally, A3C explores the environment using four distinct methods simultaneously, achieving greater performance. The performance of the other three solutions is comparable. The DQN curves start higher than the AC and MADQN curves but converge closely after 300s. Anyhow, the AC network may exhibit higher unit delay metrics in task offloading due to the complexity of simultaneously learning both policy and value functions.

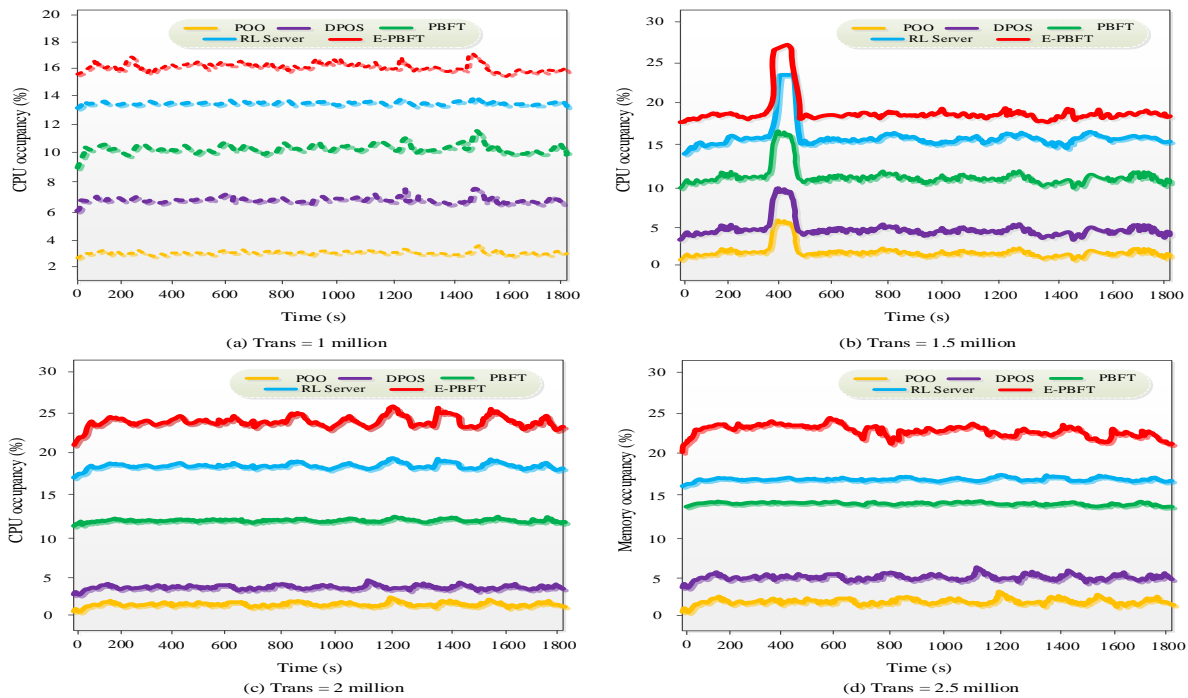


Figure 4. Occupancy of CPU while testing

B. Blockchain Result Analysis

Using the consensus method to process the blocks may cause the system to lag and consume more resources. Optimizing resource consumption and speeding up block processing can enhance the overall performance of the system. We tested our proposed E-PEFT in terms of CPU and memory occupancy, bandwidth usage, and block building time against two popular consensus mechanisms: DPoS and PBFT.

(i) CPU Usage

CPU Usage: Figure 4 displays the CPU occupancy for each variation in transaction volume during the course of the test. Our E-PEFT has the highest CPU occupancy across four distinct scenarios; when the transaction volume reaches 2.5 million, E-PEFT is roughly 500% and 200% higher than DPoS and PBFT, respectively. This high CPU consumption is due to each blockchain worker using an RL model to generate the job offloading policy, which consumes a significant portion of computational resources. DPoS maintains relatively low and steady CPU usage even with fluctuations in transaction traffic because blocks are agreed upon and

output by a fixed committee of workers. PBFT performs similarly, as each node casts a vote for the block's validation. Despite their lower CPU consumption, PBFT and DPoS cannot handle task offloading decisions independently; instead, an external optimizer must manage task offloading. The purple curves in the figures indicate the CPU usage of an RL-based task offloading optimizer, whose performance closely matches that of the E-PEFT, especially under high transaction volumes. Applying E-PEFT to the blockchain-based system will therefore not lead to an overall rise in CPU usage.

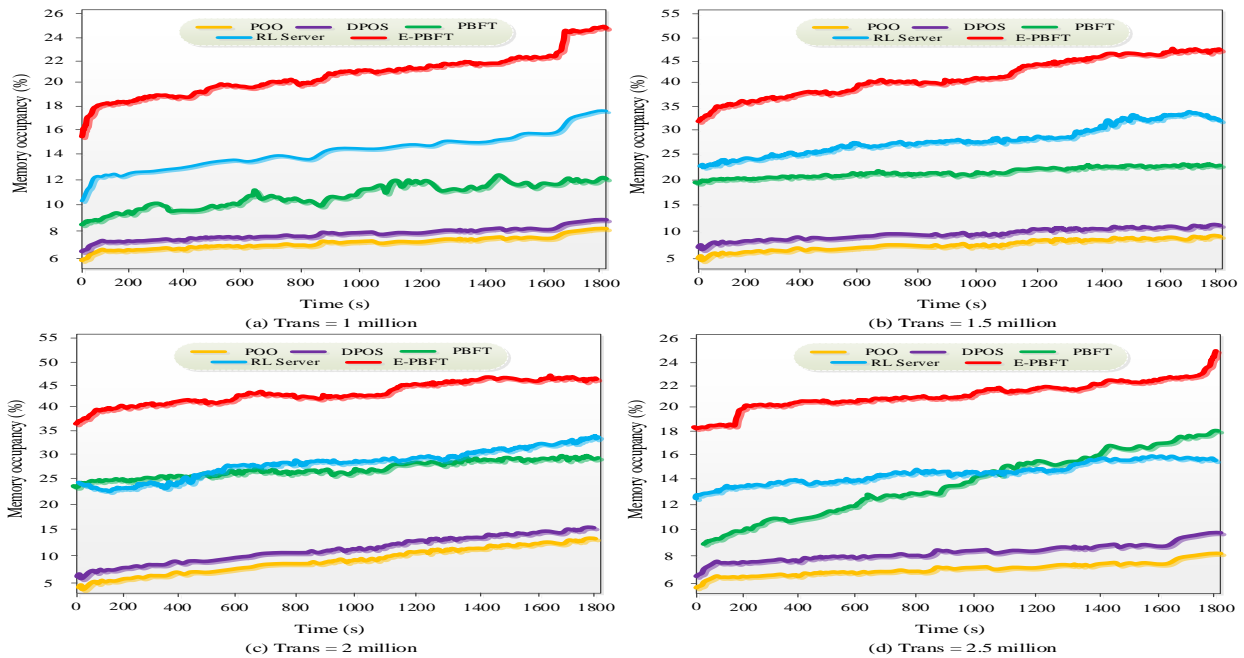


Figure 5. Occupancy of Memory while testing

(ii) Memory Occupancy

Figure 5 displays the memory occupancy during the test alongside the variation in transaction volumes. Across all transaction volume settings, the E-PEFT exhibits the highest memory occupancy, consistent with the CPU occupancy statistics. However, the autonomous optimizer also experiences elevated memory consumption, especially under high transaction volumes. For instance, at 1000s, the E-PEFT uses 24%, 39%, 42%, and 48% of memory, while the independent optimizer uses 16%, 27.2%, 37%, and 40%. In contrast, PBFT and DPoS maintain memory occupancies around 10% for most of the test. These results indicate that using the E-PEFT, rather than a separate consensus mechanism and independent task offloading optimization, can reduce overall memory utilization.

(iii) Bandwidth Consumption

Figure 6 shows the average bandwidth usage during the test at various transaction volumes. The chart indicates that when the transaction volume is one million, POO has greater bandwidth occupancy compared to the other two consensus mechanisms. However, as the number of transactions increases, POO's bandwidth usage becomes the lowest. This is because POO employees need to gather environmental data, which initially increases bandwidth usage. As the transaction volume rises, communication between employees consumes a larger portion of the bandwidth, eventually becoming the primary user of communication resources. When the transaction volume is between 1.5 and 2.5 million, DPoS consumes the most bandwidth due to the frequent elections of witnesses and witness leaders. Additionally, to enhance the performance of the TBBS-TO model, bandwidth allocation is dynamically optimized to ensure efficient resource allocation without delays. This dynamic allocation strategy aims to minimize energy consumption by precisely distributing bandwidth according to real-time demands, thereby avoiding unnecessary use of network resources. By effectively managing bandwidth allocation in this manner, the TBBS-TO model optimizes energy efficiency, ensuring that resources are utilized in a sustainable and economical manner for task offloading operations.

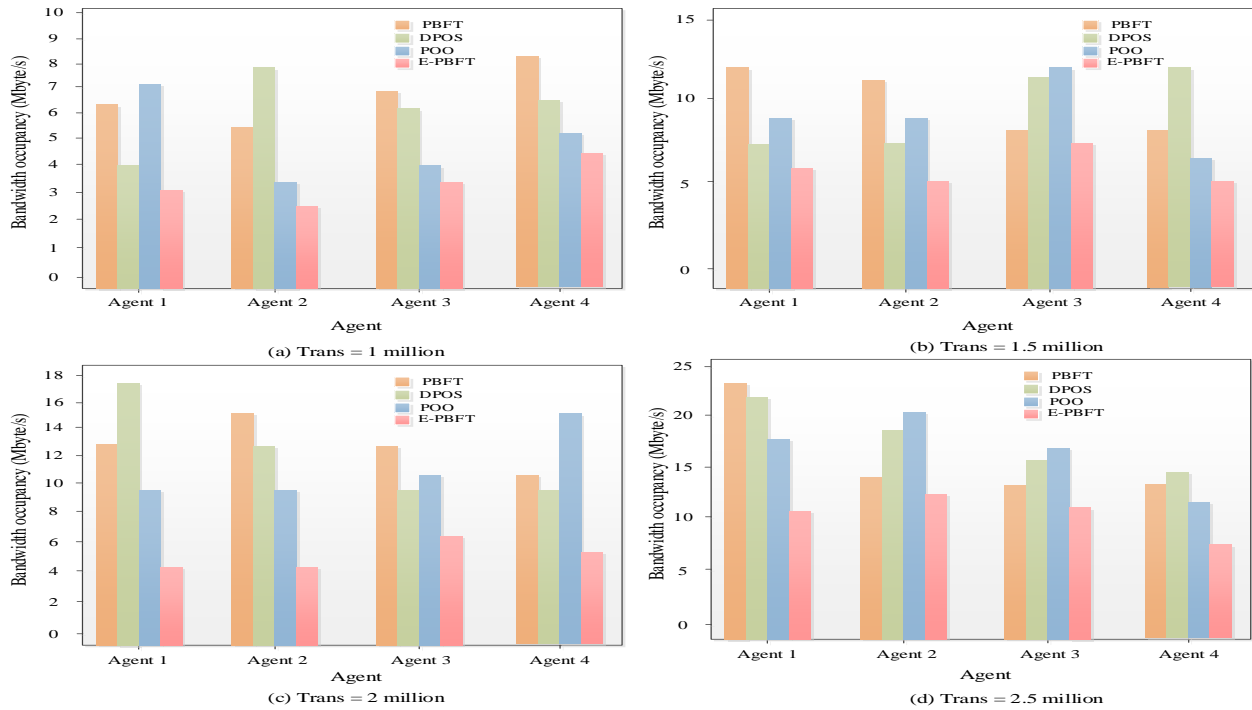


Figure 6. Consumption of Average Bandwidth

(iv) Transaction Validating Delay

Consequently, there are more pending transactions, which increases the Transaction Validating Delay (TVD). When the transaction arrival rate is less than the processing rate, the backlog of pending transactions decreases. Every curve in the test for the stabilized pattern oscillates around 2 seconds. This variation occurs because each block is added to the blockchain when the total number of transactions within it reaches 10,000 or the 3-second production deadline. As a result, transactions added to a block that is nearly full or nearing the building deadline may experience less delay. Additionally, we note that the three consensus techniques perform similarly in TVD because the workers' TBBS-TO network is trained and the optimization process progresses quickly compared to the block processing latency.

Therefore, when transaction volumes increase, more transactions remain pending, raising the TVD. Conversely, when the transaction arrival rate is lower than the processing rate, the backlog of pending transactions diminishes. Each test curve for the stabilized pattern oscillates around 2 seconds. This is because blocks are added to the blockchain when they contain 10,000 transactions or meet the 3-second production deadline. Hence, transactions added to nearly full blocks or those close to the building deadline may experience reduced delay. Furthermore, the three consensus techniques show similar TVD performance due to the fast training and optimization process of the workers' RL network relative to block processing latency.

C. Offloading Performance Evaluation

Using our designed collaborative transcoding platform as a basis, we compare the TBBS-TO with our previously proposed task offloading algorithm AGO policy based on the stochastic network optimization framework, while the random policy chooses the node for offloading transcoding requests using a uniform distribution. Figures 8(a) and 8(b) show the average CPU and memory usage of various types of CUs. Our problem formulation considers both the task's processing delay and transmission capability. Heavily relying on the edge servers can congest the links between subscribers and edge servers, increasing overall latency. This conclusion is supported by Figure 8(c), which shows that TBBS-TO has a lower latency on the edge side compared to AGO. Furthermore, as illustrated in Figure 8(d), TBBS-TO also preserves edge bandwidth pressure. Time-sensitive compute tasks do not favor the random strategy since it uniformly distributes the tasks among two to three types of CUs, resulting in the largest processing delay and highest throughput on the cloud side.

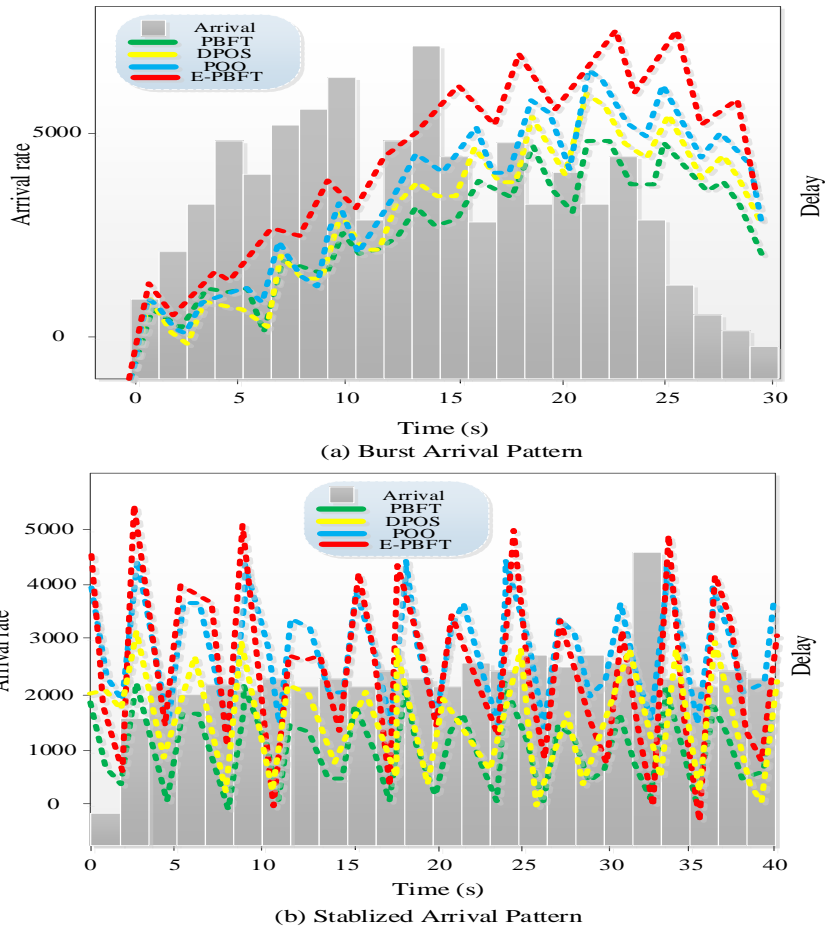


Figure 7. Validating Delay of Transaction

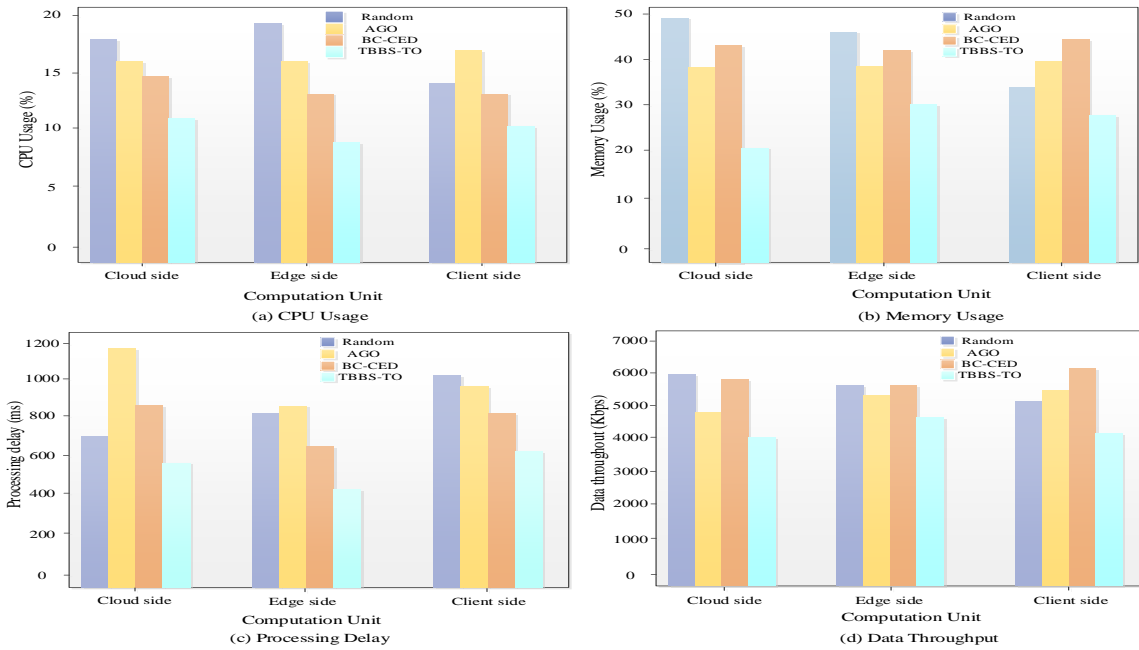


Figure 8. Performance Analysis of Three Solutions

5. Conclusion

The ability to facilitate high-performance task offloading while maintaining participant confidence is crucial, though not essential, to Cloud-Edge-Network (CEN) computing due to the geographic distribution and operation by various parties. Furthermore, conflicts of interest may arise among the highly dynamic and diverse CEN members who provide resources. This study introduces a novel collaborative task offloading framework for CEN computing, termed TBBS-TO and resource allocation empowered CEN. This innovative framework incorporates the E-PEFT consensus algorithm, which optimizes resource allocation and task execution by dynamically adjusting consensus parameters based on environmental factors and performance feedback. One of the core challenges addressed in this work is the heterogeneity of IoT users and their mobility. To alleviate these issues, mobility-aware clustering is performed using the BCALoD. This approach enhances the overall efficiency and effectiveness of resource allocation and task offloading in the CEN environment. Blockchain technology plays a pivotal role in BC-CED's core functions, such as task delegation, resource utilization brokerage, and bandwidth-sensitive resource allocation. By modifying the blockchain consensus procedure, TBBS-TO distinguishes itself from other solutions by enabling participants to reach a consensus on task offloading. To achieve this, the offloading problem is formulated by considering both network performance and the computational capabilities of potential nodes. This formulation is crucial in ensuring that the task offloading is both efficient and effective, leveraging the strengths of the network while minimizing its weaknesses. The use of MA-DDQN techniques allows TBBS-TO to dynamically adapt and optimize the task offloading process. This method enables participants to compete for the right to produce a block by evaluating offloading policies and selecting the most effective one for the next period. This competitive process ensures that the best possible offloading policy is implemented, leading to improved performance and efficiency.

In addition to task offloading, TBBS-TO also incorporates dynamically bandwidth-sensitive resource allocation. This process considers significant parameters to ensure that resources are allocated in the most efficient manner possible, considering the current state of the network and the specific needs of the tasks being offloaded. This dynamic approach to resource allocation ensures that the system can adapt to changing conditions and continue to operate effectively even in the face of varying network conditions and task requirements. Comprehensive testing of TBBS-TO was conducted on a commercial blockchain platform, demonstrating its superiority over existing solutions in both task offloading and blockchain maintenance. The results of these tests show that TBBS-TO not only outperforms other solutions in terms of efficiency and effectiveness but also provides a more reliable and trustworthy framework for CEN computing. This study highlights the potential of blockchain technology in enhancing the performance and reliability of task offloading in Cloud-Edge-Network environments, offering a promising solution to the challenges faced in this field. In conclusion, TBBS-TO represents a significant advancement in the field of CEN computing, providing a robust and efficient framework for task offloading and resource allocation. By leveraging blockchain technology and advanced algorithms such as E-PEFT and MA-DDQN, this framework addresses the core challenges of geographic distribution, participant diversity, and resource heterogeneity. The comprehensive testing and positive results demonstrate the effectiveness of TBBS-TO, offering a promising solution for future applications in Cloud-Edge-Network computing. This study paves the way for further research and development in this area, highlighting the potential of blockchain-enabled task offloading frameworks in enhancing the performance and reliability of distributed computing environments.

References

- [1] Ecer, F., & Pamucar, D. (2022). A novel LOPCOW-DOBI multi-criteria sustainability performance assessment methodology: An application in developing country banking sector. *Omega*, 112, 102690.
- [2] Ambeth Kumar, V.D. Vaishali, S. Shweta, B. (2015). Basic Study of the Human Foot. *Biomedical and Pharmacology*, 8(1), 435-444.
- [3] Zhang, Y., Li, B., Liu, B., Hu, Y., & Zheng, H. (2021). A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain. *IEEE Internet of Things Journal*, 8(18), 13958-13974.
- [4] Devi Murugavel, Kiruthiga, Parthasarathy Ramadass, Rakesh Kumar Mahendran, Arfat Ahmad Khan, Mohd Anul Haq, Sultan Alharby, and Ahmed Alhussen. 2022. "Maintaining Effective Node Chain Connectivity in the Network with Transmission Power of Self-Arranged AdHoc Routing in Cluster Scenario" *Electronics* 11, no. 15: 2455. <https://doi.org/10.3390/electronics11152455>.
- [5] Zhang, F., Han, G., Liu, L., Martínez-García, M., & Peng, Y. (2021). Joint optimization of cooperative edge caching and radio resource allocation in 5G-enabled massive IoT networks. *IEEE Internet of Things journal*, 8(18), 14156-14170.
- [6] Sherubha, "Graph Based Event Measurement for Analyzing Distributed Anomalies in Sensor Networks", *Sādhanā* (Springer), 45:212, <https://doi.org/10.1007/s12046-020-01451-w>
- [7] Piyush K. Pareek, Pixel Level Image Fusion in Moving objection Detection and Tracking with Machine Learning "Fusion: Practice and Applications, Volume 2, Issue 1, PP: 42-60, 2020
- [8] Shivam Grover, Kshitij Sidana, Vanita Jain, "Egocentric Performance Capture: A Review", *Fusion: Practice and Applications*, Volume 2, Issue 2, PP: 64-73, 2020.

- [9] Abdel Nasser H. Zaid, Mahmoud Ismail and Salwa El- Sayed, A Survey on Meta-heuristic Algorithms for Global Optimization Problems, Journal of Intelligent Systems and Internet of Things, Volume 1 , Issue 1, PP: 48-60, 2020
- [10] Mahmoud H. Alnamoly, Ahmed M. Alzohairy, Ibrahim M. El-Henawy, "A survey on gel images analysis software tools, Journal of Intelligent Systems and Internet of Things, Volume 1 , Issue 1, PP: 40-47, 2021.
- [11] Yu, S., & Park, Y. (2022). A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. *IEEE Internet of Things Journal*, 9(20), 20214-20228.
- [12] Jacob, I. J., & Darney, P. E. (2021). Design of deep learning algorithm for IoT application by image based recognition. *Journal of ISMAC*, 3(03), 276-290.
- [13] Kumar, V.D.A., Sharmila, S., Kumar, A. et al. (2023). A novel solution for finding postpartum haemorrhage using fuzzy neural techniques. *Neural Comput & Applic.* 35(33), 23683–23696
- [14] Panahi, U., & Bayılmış, C. (2023). Enabling secure data transmission for wireless sensor networks based IoT applications. *Ain Shams Engineering Journal*, 14(2), 101866.
- [15] Nguyen, X. H., Nguyen, X. D., Huynh, H. H., & Le, K. H. (2022). Realguard: A lightweight network intrusion detection system for IoT gateways. *Sensors*, 22(2), 432.
- [16] Sathya Preiya, V., and V. D. Ambeth Kumar. (2023). Deep Learning-Based Classification and Feature Extraction for Predicting Pathogenesis of Foot Ulcers in Patients with Diabetes. *Diagnostics* 13(12), 1983.
- [17] Indhumathi, M., Ambeth Kumar, V.D, " Future prediction of cardiovascular disease using deep learning technique", *Advances in Parallel Computing*, 2021, 38, pp. 219–223
- [18] Chen, L., Zhang, D. G., Zhang, J., Zhang, T., Wang, W. J., & Cao, Y. H. (2023). A novel offloading approach of IoT user perception task based on quantum behavior particle swarm optimization. *Future Generation Computer Systems*, 141, 577-594.
- [19] Abd Elaziz, M., Abualigah, L., & Attiya, I. (2021). Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments. *Future Generation Computer Systems*, 124, 142-154.
- [20] Hussain, F., Abbas, S. G., Shah, G. A., Pires, I. M., Fayyaz, U. U., Shahzad, F., ... & Zdravevski, E. (2021). A framework for malicious traffic detection in IoT healthcare environment. *Sensors*, 21(9), 3025.
- [21] Ruphitha, S.V., Ambeth Kumar, V.D., " Predictive analysis of postpartum haemorrhage using deep learning technique", *Advances in Parallel Computing*, 2021, 38, pp. 168–172.
- [22] Aazam, M., ul Islam, S., Lone, S. T., & Abbas, A. (2020). Cloud of things (CoT): Cloud-fog-IoT task offloading for sustainable Internet of Things. *IEEE Transactions on Sustainable Computing*, 7(1), 87-98.
- [23] Vishwakarma, S., Goswami, R. S., Dutta, S., Sakthivel, V., Prakash, P., Vijayakumar, P., & Thangavelu, L. (2023). Cloud data storage with improved resource scheduling in healthcare application based on security system. *Optik*, 272, 170225.
- [24] Cao, S., Zhan, Z., Dai, C., Chen, S., Zhang, W., & Han, Z. (2023). Delay-Aware and Energy-Efficient IoT Task Scheduling Algorithm with Double Blockchain Enabled in Cloud-Fog Collaborative Networks. *IEEE Internet of Things Journal*.
- [25] Vijayasekaran, G., & Duraipandian, M. (2022). An efficient clustering and deep learning based resource scheduling for edge computing to integrate cloud-IoT. *Wireless Personal Communications*, 124(3), 2029-2044.
- [26] Zhang, J., Li, T., Ying, Z., & Ma, J. (2022). Trust-based secure multi-cloud collaboration framework in cloud-fog-assisted IoT. *IEEE Transactions on Cloud Computing*, 11(2), 1546-1561.
- [27] Raheema, A. Q., & Tarish, H. A. (2023). Secure data transfer aware grouping technique for cloud-assisted Internet of things applications. *International Journal of Communication Systems*, 36(1), e4129.
- [28] Narayanan, U., Paul, V., & Joseph, S. (2022). Decentralized blockchain based authentication for secure data sharing in Cloud-IoT: DeBlock-Sec. *Journal of Ambient Intelligence and Humanized Computing*, 13(2), 769-787.
- [29] Raghavendar, K., Batra, I., & Malik, A. (2023). A robust resource allocation model for optimizing data skew and consumption rate in cloud-based IoT environments. *Decision Analytics Journal*, 7, 100200.
- [30] Hemamalini, Selvamani, and Visvam Devadoss Ambeth Kumar. (2022). Outlier Based Skimpy Regularization Fuzzy Clustering Algorithm for Diabetic Retinopathy Image Segmentation. *Symmetry*, 14(12), 2512.
- [31] Zolfaghari, B., Yazdinejad, A., Dehghantanha, A., Krzciok, J., & Bibak, K. (2022). The dichotomy of cloud and iot: Cloud-assisted iot from a security perspective. *arXiv preprint arXiv:2207.01590*.
- [32] Vishwakarma, S., Goswami, R. S., Dutta, S., Sakthivel, V., Prakash, P., Vijayakumar, P., & Thangavelu, L. (2023). Cloud data storage with improved resource scheduling in healthcare application based on security system. *Optik*, 272, 170225.
- [33] Bai, Z., Li, C., Pourzamani, J., Yang, X., & Li, D. (2024). Optimizing the resource allocation in cyber physical energy systems based on cloud storage and IoT infrastructure. *Journal of Cloud Computing*, 13(1), 59.
- [34] Godhrawala, H., & Sridaran, R. (2022, November). Improving architectural reusability for resource allocation framework in futuristic cloud computing using decision tree based multi-objective automated approach. In *International*

- conference on advancements in smart computing and information security (pp. 397-415). Cham: Springer Nature Switzerland.
- [35] Tay, M., & Senturk, A. (2023). A research on resource allocation algorithms in content of edge, fog and cloud. *Materials Today: Proceedings*, 81, 26-34.
- [36] Chowdhary, S. K., & Rao, A. L. N. (2023). A task clustering based QoS aware scheduling algorithm for task execution in cloud-Iot model for education services. *Multimedia Tools and Applications*, 82(29), 44783-44800.
- [37] Bu, T., Huang, Z., Zhang, K., Wang, Y., Song, H., Zhou, J., ... & Liu, S. (2024). Task scheduling in the internet of things: challenges, solutions, and future trends. *Cluster Computing*, 27(1), 1017-1046.
- [38] Fazel, E., Najafabadi, H. E., Rezaei, M., & Leung, H. (2023). Unlocking the power of mist computing through clustering techniques in IoT networks. *Internet of Things*, 22, 100710.
- [39] Su, M., Wang, G., & Choo, K. K. R. (2022). Prediction-Based Resource Deployment and Task Scheduling in Edge-Cloud Collaborative Computing. *Wireless Communications and Mobile Computing*, 2022(1), 2568503.
- [40] Mahapatra, A., Majhi, S. K., Mishra, K., Pradhan, R., Rao, D. C., & Panda, S. K. (2024). An energy-aware task offloading and load balancing for latency-sensitive IoT applications in the Fog-Cloud continuum. *IEEE Access*.
- [41] Yadav, A., Jana, P. K., Tiwari, S., & Gaur, A. (2022). Clustering-based energy efficient task offloading for sustainable fog computing. *IEEE Transactions on Sustainable Computing*, 8(1), 56-67.