BEES: Blockchain-Enabled Edge Intelligence Sharing Using Federated Learning

Hardhik Mohanty, Riya Tapwal, and Sudip Misra, *Senior Member, IEEE*

***Abstract*—Intelligent tasks involving Machine Learning / Deep Learning (ML/DL) model training, require massive data distributed across several edge nodes. This distributed data is privacy-protected and heterogeneous. Also, edge nodes are resource constrained in terms of computation capability, battery, and bandwidth. These factors make ML/DL model training across edge nodes challenging. We propose a federated learning and blockchain-enabled edge intelligence-sharing framework for distributed model training to address these issues. Each resource constrained edge node performs local computation with its local dataset or offloads the data to the nearest edge server to train the ML/DL model for a few epochs. Next, the selected main edge server encapsulates the updated weights to the blockchain. Further, the smart contract utilizes the updated weights collected from the selected edge nodes for the global aggregation process. Blockchain ensures security against cyber-attacks, while federated learning guarantees data privacy and faster convergence to the optimal solution. The efficacy of the proposed framework is demonstrated through extensive simulations using the MNIST handwritten dataset and considering the edge computing framework. The accuracy graphs indicate that the proposed model obtains a higher accuracy of 89% than the baseline model accuracy of 72% at the end of 50 communication rounds. Further, the accuracy of the proposed model drops only by 2%, when the number of local devices decreases to half. This result indicates robustness against user equipment dynamics. Additionally, the proposed model shows improved end-to-end latency and energy efficiency performance.**

***Index Terms*—Blockchain, Federated Learning, Edge Intelligence, Data Offloading, Intelligence Sharing, Internet of Things.**

I. Introduction

The last decade has witnessed rapid growth in Internet of Things (IoT) sensors, wearable devices, mobile phones, and edge servers [1]. The Machine Learning/Deep Learning (ML/DL) models learn useful features from the large data by utilizing the local storage and computing facilities of the edge devices. ML/DL models are utilized to predict future events, intelligent decision-making, and solve classification problems. Traditional methods proposed a centralized ML/DL model training approach where all the data present in different client devices was sent to a central cloud. However, such methods possess the risk of privacy leakage, copyright threats, and high latency. Consequently, edge computing technology [2] has been introduced to bring computational facilities closer to the data source. The integration of artificial intelligence (AI) into edge technologies [3] to enable efficient, effective, and privacy-aware wireless communication has led to the emergence of edge intelligence.

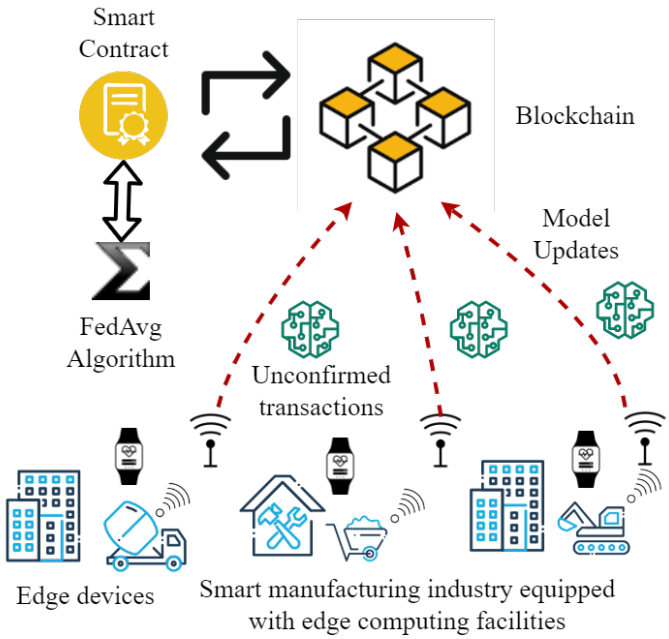

Figure 1: System architecture of the proposed blockchain based edge intelligence sharing framework.

Using edge intelligence in real-world scenarios faces the following challenges: 1. Edge nodes are resource constrained [4] in terms of bandwidth, energy, memory, and computational power, making decentralized training of ML/DL models difficult, 2. Due to its diverse geographical locations, the local data present in edge nodes are non-independent and non-identically distributed (non-IID) [5], and 3. The local training process of edge nodes can be under cyber-security attacks that can manipulate it to produce malicious model parameters [6] degrading the overall testing performance. Federated Learning (FL) has emerged as an essential and prevalent research topic among ML practitioners for processing non-IID data. One of the pioneer works in FL by McMahan *et al.* [7] proposed the `FedAvg` algorithm. The FL framework preserves the clients' data privacy by sharing only the local model updates with the centralized server instead of the raw data. However, one of the challenges faced by FL is the presence of malicious clients (edge nodes), which can manipulate the local training process and render the global aggregation procedure ineffective. Another evolving technology known as blockchain establish

H. Mohanty is with the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, India. Email: `hardhikiitkgp2018@iitkgp.ac.in`

R.Tapwal and S.Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. Email: `tapwalriya@kgpian.iitkgp.ac.in`, `sudipm@iitkgp.ac.in`

trust among the heterogeneous and untrustworthy edge nodes by decentralizing the framework. This framework enables data sharing, replication, and synchronization efficiently via a consensus mechanism. Qiu et al. [8] proposed the Proof of Learning (PoL) consensus mechanism to minimize the energy wastage in calculating the unique hash function.

The present study intricately interweaves blockchain and FL into edge intelligence to simultaneously address both the security and privacy concerns arising from cyber-attacks. This is achieved through a decentralized approach and keeping the data restricted to the source device. Other existing approaches such as B-EI [8] and IFLBC [9] which do not integrate blockchain and FL for edge intelligence sharing, face the disadvantage of single-point server failure due to cyber-attacks and inferior model performance due to non-IID data present among the edge nodes. The proposed model also learns the optimal ML/DL model weights from heterogeneous data which are non-IID and dispersed across resource constrained edge devices. The resource constrained edge nodes decide to offload their data to their nearest edge servers or utilize their local computational resources based on the residual battery level. Then the model training is performed by either the local device itself or the nearest edge server for only a few epochs. Further, the updated weight parameters and the accuracy obtained on the test dataset are uploaded as a transaction to the block by the edge server. The proposed work adapts the Proof of Learning (PoL) consensus to select the main edge server, responsible for verifying the transactions and calling the smart contract to execute the `FedAvg` algorithm. A smart contract is an executable code that resides within the blockchain. Once the `FedAvg` algorithm computes the updated model weights, all the edge nodes can download it from the blockchain for the next round of local training with initial weights the same as the updated weights. The edge nodes are modeled based on their energy and computational resources.

Example Scenario: We demonstrate the proposed edge intelligence framework with the help of an example. One of the major application domains of edge intelligence is smart manufacturing industry. As shown in Fig. 1, each smart manufacturing industry has a small base station (SBS) associated with it. The SBS provides edge computing facilities for the local training process. The sensory devices that record medical data are resource constrained in battery, storage, and computational facilities. Consequently, these end devices either perform local computation or offload the data to the corresponding edge server, whichever leads to lower latency. The industrial data is sensitive, and privacy protected. Consequently, we utilize federated learning to confine the data to the smart manufacturing industry and only share the model updates. Blockchain technology shares intelligence among the different smart manufacturing industry through a decentralized mechanism. The blockchain component of the proposed model can prevent the degradation of machines' health prediction model accuracy even in the presence of malicious edge nodes compromised by cyber-attacks. Further, the federated averaging algorithm is executed by invoking a smart contract. This algorithm efficiently learns the optimal model weights from different smart manufacturing industries' non-IID data due to their varying geographical location.

A. Motivation

Edge Intelligence has opened up numerous opportunities for developing path-breaking applications. The real-world scenario of Industrial IoT (IIoT) can be modeled as an Edge Intelligence framework. IIoT networks are highly complex [10] due to the vast range of interconnected devices, sensors, and other industrial machines generating a massive amount of data. The IIoT framework [11] has strict latency, computational power, caching storage, and energy capacity demands. However, this framework consists of resource constrained devices that affect intelligent decision-making. Moreover, it is prone to cyber-security threats such as model poisoning attack [12] which send malicious ML/DL updates to severely affect the training process of global model. This work proposes integrating blockchain and FL into edge computing to resolve these critical issues. The blockchain utilizes asymmetric cryptographic keys to secure the transactions made by the edge nodes. Moreover, selecting the main node at each round for verifying transactions prevents potential cyber-attacks. FL ensures data privacy and faster convergence to the optimal solution. When combined with the advantages of edge computing, these two frameworks can support delay-critical industrial applications, maximize resource utilization, and guarantee secure intelligence sharing among edge nodes.

B. Contribution

The following are the primary *contributions* of this research work:

1) We propose an inter-weaved blockchain and federated learning architecture into the edge computing framework to learn the optimal ML/DL model parameters from data distributed among resource-constrained IoT nodes.
2) We safeguard the edge computing framework against cyber-attacks such as model poisoning attacks and simultaneously keeping the data private to the local end device.
3) We formulate the energy and computational model for each end device. This information is utilized for intelligent decision-making on task offloading by end devices.
4) We built a decentralised mechanism for sharing edge intelligence using blockchain technology. The local model updates are considered as transactions to the blockchain that are further verified by the main node.

As highlighted above, the significant advantage of the proposed model compared to the existing approaches lies in sharing of edge intelligence among resource-constraint end-users following a secure and privacy-aware technique. It may be further noted that, in this paper, we are focusing on using a combination of FL and blockchain into an edge computing framework to learn the best set of parameters effectively from data spread across resource-constrained edge devices. We plan to test the performance of the proposed framework in our extended work.

The rest of this paper is organized as follows. Section II reviews the present research in the subject of blockchain and federated learning applications to edge computing frameworks and points out the flaws in existing approaches. Section III gives a brief background behind the proposed framework and describes the system model. Section IV gives the implementation details of the simulated experiments, discusses the results, and demonstrates the efficacy of the proposed method over the baseline. Furthermore, Section V concludes up the paper and discusses future scope of the work.

II. Related Work

This section summarizes the existing literature on edge intelligence and discusses the application of blockchain and federated learning in this research domain. Blockchain provides a secure data sharing framework, while simultaneously ensuring data integrity, quality, and reliability. On the other hand, FL enhances data privacy by only exchanging local model updates rather than raw data.

A. ML/DL applications to edge computing and caching

One of the essential research works has been to utilize deep learning and game theory-based techniques for intelligent edge caching and computing. For instance, Zhang *et al.* [13] proposed a novel LSTM-based caching replacement framework with strong representational ability. The popularity of the content was predicted using a LSTM-based deep learning model. Other basic modules further utilized this information to calculate caching priority for intelligent cache replacement. Moreover, the model's low training and prediction time implied that it could be deployed for real-time operations. Likewise, Gu *et al.* [14] formulated cooperation between small base stations for edge caching as a game model. The edge nodes acted as the players, and the strategy involved decision-making on cache storage and replacement. Next, the optimal strategy was identified, which maximized the global utility of the edge network by finding the Nash equilibrium. Finally, the equilibrium point was achieved using a distributed algorithm to reduce data transportation latency and backhaul traffic.

B. Blockchain for edge intelligence

Other research methodologies utilized the blockchain framework to enhance the security of edge data transfer and storage. For instance, Qui *et al.* [8] proposed a blockchain-based edge intelligence sharing mechanism among heterogeneous and dishonest nodes. Proof of Learning (PoL) consensus protocol shared the learning intelligence among the other edge nodes. This novel consensus protocol used the computational power of the resource-constrained edge nodes for training the deep learning model instead of wasting it for finding a hash with a unique pattern. The blockchain-assisted edge intelligence mechanism was tested on a joint resource assignment problem to demonstrate its time effectiveness and better resource utility. To ensure secure transmission and enable physical layer security, Sun *et al.* [21] proposed a blockchain-based edge caching framework. A probabilistic caching strategy

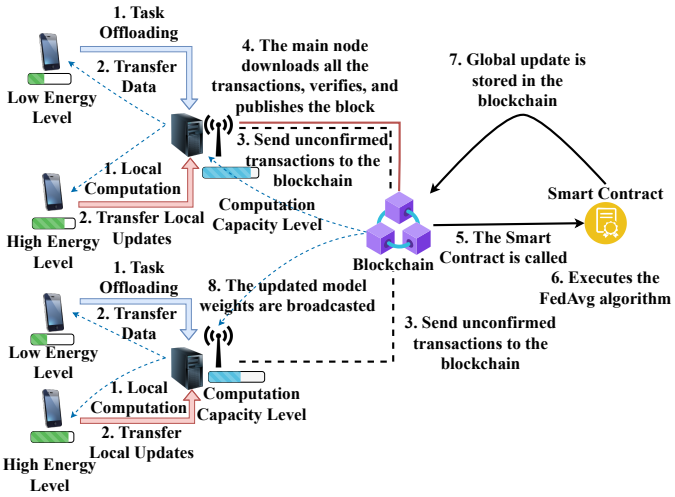

Figure 2: Network architecture of blockchain-enabled edge intelligence sharing using federated learning.

was adopted, which maximized the utilization of available cache resources. Furthermore, simultaneous optimization of cache hit and redundancy rate was performed to enable secure transmission. Liu *et al.* [22] proposed a blockchain-based decentralized edge caching infrastructure to improve content providers' quality of service (QoS). A future market-based cache resource trading system was designed between the content provider and edge device to enhance the utilization of caching resources. Furthermore, a trade contract management system with data integrity verification was developed for safe data exchange between edge nodes.

Other research works studied the application of blockchain to create incentives among edge nodes for providing computational and storage facilities. For instance, Xu *et al.* [15] proposed a blockchain-based decentralized crowd intelligence framework to alleviate network congestion in edge computing platform. They devised a reward-penalty framework to match the platform's, publisher's, and workers' incentives. It also included a hybrid human-machine platform to address the workforce scarcity issue. Finally, a strong Nash Equilibrium point identified the edge server's optimal strategy, maximizing its interest in serving as the blockchain main node. Li *et al.* [16] presented a novel user-centric blockchain (UCB) based architecture to facilitate secure knowledge sharing among intelligent edge nodes. Moreover, an energy-efficient consensus mechanism called Proof of Popularity (PoP) was formulated. This consensus mechanism considered the knowledge contribution of each edge node to create incentives among themselves and to safeguard against security threats. Xu *et al.* [23] proposed Edgence, which added a new feature of master node technology into blockchain to extend its applicability to IoT-based decentralized applications (dApps). A three-tier validation approach was developed to assist numerous IoT-based dApps in order to improve efficiency. The selection of the validator node was implemented in an energy-efficient manner to validate and generate a new block. The proposed framework demonstrated effectiveness against Sybil attack by

Table I: Comparison of existing works with BEES

Paper	Blockchain	Federated Learning	Edge Computing	Intelligence Sharing	Description
Qiu *et al.* [8]	✓	✗	✓	✓	Blockchain-based edge intelligence sharing mechanism among heterogeneous and dishonest nodes.
Xu *et al.* [15]	✓	✗	✓	✗	Blockchain-based decentralized crowd intelligence framework to alleviate network congestion in edge computing platform.
Li *et al.* [16]	✓	✗	✓	✓	UCB based architecture to facilitate secure knowledge sharing among intelligent edge nodes.
Zhang *et al.* [17]	✓	✗	✓	✗	Blockchain empowered and edge intelligent framework for IIoT to ensure secure edge service management.
Nawaz *et al.* [18]	✓	✗	✓	✗	Decentralized ledger architecture using blockchain for for applications that require high levels of privacy and data security.
Doku *et al.* [9]	✓	✓	✓	✗	EI framework based on blockchain and FL to make AI services omnipresent and accessible by all end devices.
Fan *et al.* [19]	✓	✓	✓	✗	Hybrid blockchain-based resource exchange technique between end nodes and service requesters to implement FL.
Lu *et al.* [20]	✓	✗	✓	✗	Blockchain-based FL architecture to provide users with data security and privacy guarantees.
BEES (Proposed Work)	✓	✓	✓	✓	Blockchain-enabled intelligence sharing among resource constrained edge nodes using federated learning.

using the collateral of master nodes to increase the entry threshold for malicious edge nodes.

C. Blockchain for IIoT

Recent research utilizes a blockchain-enabled intelligent edge framework to solve complex Industrial Internet of Things (IIoT) network scenarios. For instance, Zhang *et al.* [17] proposed a blockchain-enabled edge intelligent framework for IIoT to ensure secure edge service management. An edge resource scheduling scheme was formulated for a cross-domain sharing strategy and Deep Reinforcement Learning (DRL) model. Furthermore, a novel credit-differentiated transaction approval mechanism was devised for reaching edge resource transaction consensus. Kumar *et al.* [24] proposed a blockchain-enabled edge computing framework named BlockEdge for IIoT applications. This framework was developed to fulfill the IIoT service requirements: low latency, data integrity, decentralized trust, and security management. Nawaz *et al.* [18] integrated a decentralized ledger architecture with edge intelligence using Ethereum blockchain for applications that require high levels of privacy and data security. Additionally, the smart contract was built to allow edge devices to immediately analyze, distribute, and integrate data. Furthermore, the risk of privacy leakage was reduced by saving only the processed information instead of raw data in the blockchain.

D. Integration of Blockchain-FL for edge intelligence

Similar to our work, Doku *et al.* [9] proposed an edge intelligence framework based on blockchain and federated learning named iFLBC to make artificial intelligence (AI) services omnipresent and accessible by all end devices. The global model formed by aggregating the local edge models were securely stored in the blockchain. Moreover, a consensus protocol called proof of common interest (PoCI) was devised to filter only the relevant data from the massive amount of irrelevant data for solving the issue of helpful data scarcity. Fan *et al.* [19] proposed a hybrid blockchain-based resource exchange technique between end nodes and service requesters to implement FL. for implementing federated learning. Data quality-driven reverse auction (DQDRA) mechanism was built to facilitate edge node auctions that are self-contained and auditable. Moreover, the public channel was integrated into the blockchain to enable credible, fast payment transactions between the participating edge nodes and requesters. Cui *et al.* [25] proposed a novel algorithm named CREAT to cache popular files intelligently using blockchain and remote cloud technology. Furthermore, this research developed a compression technique to improve FL's communication efficiency by lowering the bandwidth needed for uploading local models. Furthermore, blockchain was integrated into the FL framework to ensure trustful verification of locally trained models and enhance data security. Lu *et al.* [20] proposed digital twin edge networks (DITENs) to simulate the operating states of IoT device interactions. DITENs are created by combining digital twins with edge networks. It offered a blockchain-based FL architecture to provide users with data security and privacy guarantees. Moreover, an improved gradient descent scheme was utilized for the model aggregation process, and a deep reinforcement algorithm was developed for optimal user scheduling and allocation of spectrum resources.

Synthesis: We observe from Table I that previous work utilized blockchain to build incentives among edge nodes to participate in the computational resources sharing process. However, the resource-constrained aspect of IoT nodes is generally ignored by authors. Moreover, the Proof of Work (PoW) consensus protocol used in the blockchain framework leads to energy wastage in calculating the unique hash function. Other works

integrated federated learning and edge computing to keep the data private to the end device. However, these methods faced the risk of cyber-attacks due to a single central server where the local model updates are processed. Additionally, some authors integrated blockchain and edge computing to secure the framework against cyber-attacks. However, such methods faced the issue of training with non-IID data distribution at the different edge nodes. The proposed work integrates FL and blockchain to the edge computing framework for secure and faster edge intelligence sharing among resource constrained IoT devices to resolve these issues. Additionally, the proposed work adapts the Proof of Learning (PoL) consensus mechanism to utilize the available battery power for local model training, further encapsulated as a transaction to the blockchain.

III. System Model

This study considers a Blockchain-based FL framework for intelligence sharing among edge devices as shown in Fig. 2.

A. Preliminaries

1) Picocell: In our proposed framework, the Picocell is considered the small base station, which provides computing facilities and serves as an edge server. Picocell is a small cellular base station that generally covers a limited area such as a house, mall, or office. The transmit power between a Picocell and UE is in the order of 250 milliwatts, and it can simultaneously support the communication of 30-60 users.
2) Blockchain: Our proposed model utilizes blockchain technology for securely sharing the model parameters among the edge nodes. It is a type of immutable database to record transactions in chronological order. A consensus mechanism is used for sharing, replicating, and synchronizing across distributed devices. Moreover, it also safeguards the framework from cyber-attacks and malicious edge nodes.
3) Smart Contract: After the main node verifies all the unconfirmed transactions in our proposed framework, the smart contract is invoked to execute the FedAvg algorithm. It is an executable piece of code that resides within the blockchain.
4) Federated Learning: In contrast to the centralized technique of routing all data to a single server, federated learning provides an efficient framework for preserving data privacy across clients by bringing the ML/DL model to the data. FL is suitable for systems with many clients and data that is imbalanced, non-independent, and non-IID. FL can also tolerate participant client drop-outs due to unpredictability in the system, such as battery fatigue or a poor network.

B. System Architecture

Assume that the wireless network consists of N small base stations (SBS) with computational capabilities which provide edge computing facilities. Let us denote the edge servers by $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$. Moreover, we consider K user equipment (UE) denoted by $\mathcal{U} = \{u_1, u_2, \cdots, u_K\}$, where K > N. Initially, each UE performs local training of its ML/DL model using its local dataset $\mathcal{D}_k$. The loss function at each client $u_k \in \mathcal{U}$ for each data sample $(x_j, y_j) \in \mathcal{D}_k$ is formulated as,

$$\mathcal{L}_k(w_k(t)) = \frac{1}{n_k} \sum_{j \in \mathcal{D}_k} l(x_j, y_j; w_k(t)) \tag{1}$$

where, $n_k = |D_k|$ denotes the data size of u_k. Next, the UE makes a decision $d_k \in \{0, 1\}$. Where, $d_k = 0$ indicates that the UE, u_k performs local computation. In contrast, $d_k = 1$ implies that the UE, u_k offloads its data to the corresponding ES which performs model training to calculate the updated weights. After receiving the updated weights from all the UE within its communication range, the ES makes an unconfirmed transaction to the blockchain corresponding to each individual UE. Next, the ES which will serve as the main blockchain node is selected to verify the unconfirmed transactions. The importance of selecting an ES (I_k) is based on Proof of Learning (PoL) protocol and is given by the following expression.

$$I_k = \frac{n_k \cdot c_k}{B_k} \tag{2}$$

where, n_k denotes the total data instances of all UEs associated with the ES, c_k denotes the computational capacity, and B_k indicates the leftover battery power corresponding to the ES s_k. Once the main node publishes the block to the blockchain, the smart contract is called to execute the FedAvg [7] algorithm. The main node selects a subset of the UE based on their accuracy score to perform the aggregation process. Let us denote the subset of UE by $\mathcal{M} = \{u_1, u_2, \cdots, u_M\}$, then the global aggregated weight w_g for block index t is given as follows:

$$w_g(t) = \frac{1}{n} \sum_{u_m \in \mathcal{M}} n_m \cdot w_m(t) \tag{3}$$

where, n_m denotes the number of data instances corresponding to UE (u_m) and $n = \sum_{u_m \in \mathcal{M}} n_m$.

C. Computation and Communication Model

Computational Model: For UE ($u_k \in \mathcal{U}$), the time required (t_k) for completion of its local model training is given as follows:

$$t_k^{\mathrm{C}} = \frac{L_k C_k n_k}{f_k} \tag{4}$$

where, C_k, L_k, n_k, f_k denotes the CPU cycles/Data samples, local iterations, number of data instances, and CPU cycles/second, respectively.

Communication Model: According to Shannon-Hartley theorem, the data transfer rate (r_k) that is achievable between UE (u_k) and ES (s_n) is given as follows:

$$r_k = b_k \log_2 \left(1 + \frac{p_k h_k}{N_0}\right) \tag{5}$$

where, b_k denotes the bandwidth assigned to UE (u_k), p_k denotes the average transmit power, h_k denotes the channel gain between UE and SBS, which is assumed to be equal to $r^{-\gamma}$ with γ signifying the path loss exponent and r denotes the distance between UE and ES. N_0 indicates the power spectral

Algorithm 1: Blockchain-enabled FedAvg algorithm.

Input: Heterogeneous data sensed by IoT sensors
Output: Updated global model w_g
Procedure:
while *sensor* $s_i \in S$ *senses data* **do**
  for *each participant* $u_i \in \mathcal{U}$ **do**
    $w_l^i \leftarrow$ **LocalTraining**(i, w_g)
  end
  Select the main node via PoL consensus mechanism
  Main node verifies all unconfirmed transactions
  $w_g \leftarrow$ **SmartContract**($w_l, \mathcal{N}$)
end
// Local model training
LocalTraining(i, w)**:**
Prepare mini-batch set $\mathcal{B}_i$ from local dataset $\mathcal{L}_i$
Fix learning rate α and epoch set $\mathcal{E}_i$
for *epoch* $e \in \mathcal{E}$ **do**
  for *mini-batch* $b \in \mathcal{B}_i$ **do**
    $w \leftarrow w - \alpha \nabla \mathcal{L}(w; b)$
  end
end
// Smart contract execution
SmartContract($w_l, \mathcal{N}$)**:**
Intialize $w_g \leftarrow 0$
for *local data points* $n_i \in \mathcal{N}$ **do**
  $w_g \leftarrow w_g + \frac{n_i}{\sum_{n_i \in \mathcal{N}} n_i} w_l^i$
end

density corresponding to Gaussian noise. The time delay (t_k^T) due to transmission of data between UE and ES is given by,

$$t_k^{\mathrm{T}} = \frac{L_k}{r_k} \tag{6}$$

where, L_k is the data size offloaded by UE (u_k).

D. Energy Model

The energy required by UE (u_k) for the computation of updated local model weights is given by the following formula [26]:

$$E_k^{\mathrm{C}} = \kappa L_k C_k n_k f_k^2 \tag{7}$$

On the other hand, the energy consumed in transmitting data between UE and SBS via a wireless channel is given as follows:

$$E_k^{\mathrm{T}} = p_k t_k^{\mathrm{T}} \tag{8}$$

The total energy consumed at UE (u_k) at each communication round/issue of a block is given by:

$$E_k^{total} = E_k^{\mathrm{T}} + E_k^{\mathrm{C}} \tag{9}$$

IV. Performance Evaluation

This section presents our experimental setup and observations while employing the proposed scheme.

A. Experiment Setup

For the simulation of the proposed framework, we consider 20 edge nodes equipped with computational facilities. There are a total number of 100 user equipment which are limited in terms of energy and processing capabilities. We consider the picocell as the edge server with a network coverage radius of 100-250 m. Additionally, it can support the communication of around 30 user equipment. The number of UE associated with each edge is uniformly distributed according to $\mathcal{U}(3, 7)$. Furthermore, the transmit power of the channel between UE and edge node is considered as $p = 10$ dB with noise power spectral density as $N_0 = -174$ dBm/Hz [27]. The channel bandwidth is considered as $B = 20$ MHz. We consider the variable C_k to be uniformly distributed according to $\mathcal{U}(1, 3) \times 10^4$ CPU cycles / Data Samples for the case of user equipment. While in the case of computation by edge server, the variable C_k is uniformly distributed according to $\mathcal{U}(1, 3) \times 10^2$ CPU cycles / Data Samples. When the task offloading variable d_k is equal to 1, we consider the offloaded data size L_k to be 80 MB. On the other hand, when d_k is equal to 0, we consider the offloaded data size L_k to be 80 KB as only the local updates are transmitted. The effective capacitance value κ required for the calculation of energy consumed in computation is considered as 10^{-28} [26]. In the setup, we consider the distance between UEs and SBS with computational facilities to be uniformly spaced according to the distribution, $\mathcal{U}(100, 250)$ m. The UE communicates only with its nearest SBS for the data offloading process in our proposed architecture.

B. Baseline

The Blockchain-assisted Edge Intelligence (B-EI) [8] model has been chosen as the baseline model to evaluate the proposed (BEES) model's performance. B-EI employs the Proof of Learning (PoL) consensus protocol to minimize the wastage of computational resources and improve the learning among edge nodes. Moreover, B-EI utilizes blockchain technology to secure the framework against malicious nodes. The simulation results also showed that B-EI outperformed the traditional deep reinforcement learning model.

C. Results

Several simulations were carried out to validate the efficacy of the proposed model utilizing the Python programming language. Furthermore, the superior performance of the proposed model over baseline algorithm Blockchain-assisted Edge Intelligence (B-EI) is demonstrated by comparing the performance metrics. We use the MNIST handwritten digit dataset [28] to test the performance of both models. There are 60,000 training samples and 10,000 testing samples in this dataset. The training samples are equally distributed among the user equipment for the local training process.

1) Accuracy: The metric for evaluating the performance of the proposed model is overall accuracy. As shown in Fig. 3, the global accuracy score improves as the number of blocks in the blockchain increases. We find that the B-EI model is marginally more accurate than the proposed model when

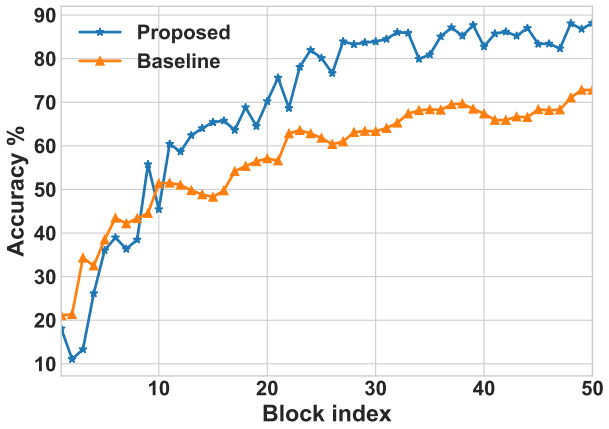

Figure 3: Improvement of accuracy with number of blocks.

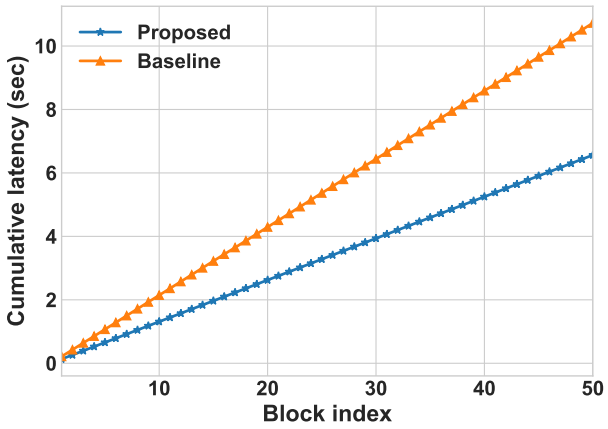

Figure 5: End-to-end latency with increasing number of blocks.

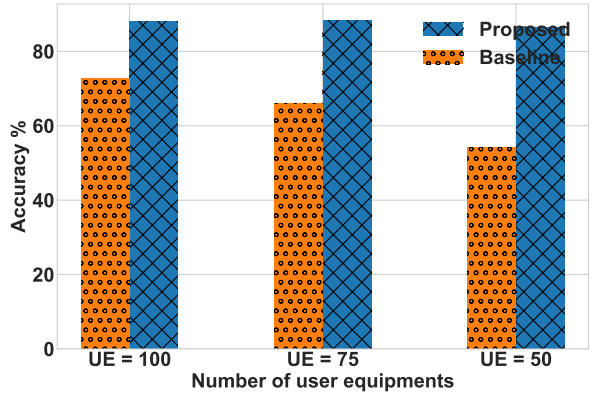

Figure 4: Change in accuracy with user equipment dynamics.

there are fewer than ten blocks in the blockchain. However, the proposed model outperforms the baseline by achieving an accuracy of 89% with an increased number of blocks. In contrast, the baseline achieves an accuracy of 72% after adding 50 blocks to the blockchain.

2) Robustness: Fig. 4 depicts the change in accuracy score with the reduction in the number of user equipment. It can be observed that the accuracy score of the baseline model at the end of 50 communication rounds falls drastically from 72% to 54% when the number of UE reduces from 100 to 50. On the other hand, the accuracy score obtained by the proposed model remains nearly constant at 88% even with the decrease in the number of user equipment. This observation indicates the robustness of the proposed model against the baseline. The proposed model executes `FedAvg` algorithm, which performs weighted averaging of the local weights of the top 10 selected UE based on their accuracy score on test data. Conversely, the baseline model selects only the UE, which obtains the best accuracy score on test data and shares it with the other edge nodes. With the reduction in the number of UEs, the number of training instances increases, due to which the local training process will be under-fitting as the number of epochs are less. We observe that selecting the local updates of only one UE for the global sharing process becomes inefficient. In contrast, the accuracy score of the proposed model does not get affected due to the implementation of `FedAvg` algorithm.

3) End-to-End Latency: Fig. 5 demonstrates the end-to-end delay in publishing the blocks to the blockchain. We observe that the end-to-end latency increases linearly with the increasing number of blocks in the blockchain. The proposed model performs better than the baseline by obtaining a lower end-to-end latency value of 5.71 seconds at the end of 50 blocks. Moreover, the performance gap increases with the increase in the block index. The baseline model performs the local computation inside the device and then offloads the updated weights to the SBS. However, the proposed model intelligently decides to either perform local computation or offload the entire data to the SBS to reduce the end-to-end delay.

4) Energy Consumption: Fig. 6-8 illustrate the total energy consumed with the increasing number of blocks in the blockchain. We observe that the proposed model consumes less energy than the baseline model, even with the increase in UE. In particular, for publishing 50 blocks to the blockchain, the proposed model consumes 6294 J, 7631 J, and 8895 J of energy corresponding to 100, 150, and 200 numbers of UE, respectively. As discussed earlier, the proposed model either performs local computation or offloads its data to the edge server, which reduces the end-to-end delay. On the other hand, the baseline model simultaneously performs local computation and then offloads its local updates to the edge server. As a result, network congestion increases, resulting in longer transmission time. Consequently, transmitting local model updates to the edge server will take more energy.

V. Conclusion

This paper addressed the problem of sharing local model updates among resource-constrained edge nodes through a secure and decentralized approach by integrating blockchain and federated learning into the edge computing framework. The blockchain framework secured the local intelligence sharing process against cyber-attacks. While federated learning was

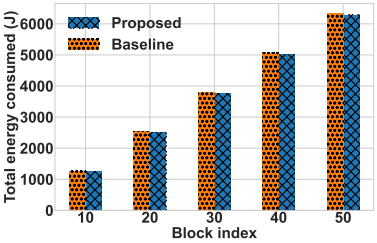

Figure 6: Energy consumed with increasing block index for 100 user equipment.

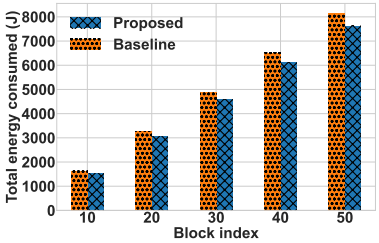

Figure 7: Energy consumed with increasing block index for 150 user equipment.

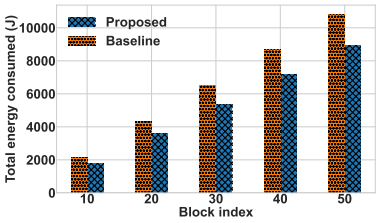

Figure 8: Energy consumed with increasing block index for 200 user equipment.

used to efficiently aggregate the local model updates acquired through training with non-IID data distribution, it was also used to safeguard the privacy of the data. We evaluated the performance of our proposed model against the baseline model of Blockchain-assisted Edge Intelligence (B-EI). According to the simulation results, the proposed model outperformed the baseline algorithm in terms of classification accuracy, robustness to UE dynamics, end-to-end latency, and energy efficiency.

In future, we plan to validate the security by testing the performance of BEES against various cyber-attacks and analyze the proposed framework from the perspective of network usage.

REFERENCES

[1] S. N. K. Marwat, Y. Mehmood, A. Khan, S. Ahmed, A. Hafeez, T. Kamal, and A. Khan, "Method For Handling Massive IoT Traffic In 5G Networks," *Sensors*, vol. 18, no. 11, p. 3966, 2018.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey On Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[3] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge Artificial Intelligence For 6G: Vision, Enabling Technologies, And Applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, 2021.

[4] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive Federated Learning In Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[5] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart Devices Are Different: Assessing And Mitigating Mobile Sensing Heterogeneities For Activity Recognition," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 127–140.

[6] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge Intelligence: Architectures, Challenges, And Applications," *arXiv preprint arXiv:2003.12172*, 2020.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning Of Deep Networks From Decentralized Data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[8] C. Qiu, X. Wang, H. Yao, Z. Xiong, F. R. Yu, and V. C. Leung, "Bring Intelligence Among Edges: A Blockchain-Assisted Edge Intelligence Approach," in *Proceedings of the GLOBECOM IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[9] R. Doku and D. B. Rawat, "IFLBC: On The Edge Intelligence Using Federated Learning Blockchain Network," in *Proceedings of the IEEE 6^{th} Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 2020, pp. 221–226.

[10] X. Zheng and Z. Cai, "Privacy-Preserved Data Sharing Towards Multiple Parties In Industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[11] L. Da Xu, W. He, and S. Li, "Internet Of Things In Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[12] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning Attack In Federated Learning Using Generative Adversarial Nets," in *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2019, pp. 374–380.

[13] C. Zhang, H. Pang, J. Liu, S. Tang, R. Zhang, D. Wang, and L. Sun, "Toward Edge-Assisted Video Content Intelligent Caching With Long Short-Term Memory Learning," *IEEE Access*, vol. 7, pp. 152 832–152 846, 2019.

[14] H. Gu and H. Wang, "A Distributed Caching Scheme Using Non-Cooperative Game For Mobile Edge Networks," *IEEE Access*, vol. 8, pp. 142 747–142 757, 2020.

[15] J. Xu, S. Wang, B. K. Bhargava, and F. Yang, "A Blockchain-Enabled Trustless Crowd-Intelligence Ecosystem On Mobile Edge Computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3538–3547, 2019.

[16] G. Li, M. Dong, L. T. Yang, K. Ota, J. Wu, and J. Li, "Preserving Edge Knowledge Sharing Among IoT Services: A Blockchain-Based Approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 653–665, 2020.

[17] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge Intelligence And Blockchain Empowered 5G Beyond For The Industrial Internet Of Things," *IEEE network*, vol. 33, no. 5, pp. 12–19, 2019.

[18] A. Nawaz, T. N. Gia, J. P. Queralta, and T. Westerlund, "Edge AI And Blockchain For Privacy-Critical And Data-Sensitive Applications," in *Proceedings of the 12^{th} International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 2019, pp. 1–2.

[19] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid Blockchain-Based Resource Trading System For Federated Learning In Edge Computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2252–2264, 2020.

[20] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-Efficient Federated Learning And Permissioned Blockchain For Digital Twin Edge Networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2276–2288, 2020.

[21] W. Sun, S. Li, and Y. Zhang, "Edge Caching In Blockchain Empowered 6G," *China Communications*, vol. 18, no. 1, pp. 1–17, 2021.

[22] J. Liu, S. Guo, Y. Shi, L. Feng, and C. Wang, "Decentralized Caching Framework Toward Edge Network Based On Blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9158–9174, 2020.

[23] J. Xu, S. Wang, A. Zhou, and F. Yang, "Edgence: A Blockchain-Enabled Edge-Computing Platform For Intelligent IoT-Based dApps," *China Communications*, vol. 17, no. 4, pp. 78–87, 2020.

[24] T. Kumar, E. Harjula, M. Ejaz, A. Manzoor, P. Porambage, I. Ahmad, M. Liyanage, A. Braeken, and M. Ylianttila, "BlockEdge: Blockchain-Edge Framework For Industrial IoT Networks," *IEEE Access*, vol. 8, pp. 154 166–154 185, 2020.

[25] L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, and W. Xiao, "Creat: Blockchain-Assisted Compression Algorithm Of Federated Learning For Content Caching In Edge Computing," *IEEE Internet of Things Journal*, 2020.

[26] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading For Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[27] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy Efficient Federated Learning Over Wireless Communication Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[28] L. Deng, "The Mnist Database Of Handwritten Digit Images For Machine Learning Research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

Hardhik Mohanty is currently enrolled in the Interdisciplinary Dual Degree Program (BTech. + MTech.) in Electrical Engineering / Artificial Intelligence, Machine Learning, and Applications at Indian Institute of Technology (IIT), Kharagpur, India. He is a 4th year undergraduate student from the department of Electrical Engineering. His current research interests include Deep Reinforcement Learning, Machine Learning, Edge Intelligence, Darknet Traffic Classification, Future Wireless Networks, and Stochastic Control.

Riya Tapwal is a Ph.D. Research Scholar in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. She has completed her M.Tech degree in Mobile Computing from the National Institute of Technology, Hamirpur, India, in 2020. Prior to that, she received the B.Tech degree in Computer Science and Engineering in 2018 from University Institute of Information Technology under-utilized Himachal Pradesh University, Shimla, India. The current research interests of Riya include Wireless Networks, Fog Computing, Industrial Internet of Things, and BC.

Sudip Misra (SM'11) is a Professor at IIT Kharagpur. He received his Ph.D. degree from Carleton University, Ottawa, Canada. Prof. Misra is the author of over 350 scholarly research papers. He has won several national and international awards, including the IEEE ComSoc Asia Pacific Young Researcher Award during IEEE GLOBECOM 2012, the INSA NASI Fellow Award, the Young Scientist Award (National Academy of Sciences, India), Young Systems Scientist Award (Systems Society of India), and Young Engineers Award (Institution of Engineers, India). He has also been serving as the Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, the IEEE SYSTEMS JOURNAL, and the INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS. Dr. Misra has 11 books published by Springer, Wiley, and World Scientific. For more details, please visit http://cse.iitkgp.ac.in/∼smisra.