Mesopotamian journal of Cybersecurity
Vol.5, No.3, **pp**. 1122–1140
DOI: https://doi.org/10.58496/MJCS/2025/060; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Blockchain-Integrated Edge-Cloud-Enabled Healthcare Data Analytics Based on Distributed Federated Learning and Deep Neural Networks

Mazin Abed Mohammed[1,2,3,*], Mohd Khanapi Abd Ghani[2,*], Israa Badr Al-Mashhadani[4], Sajida Memon[5], Abdullah Lakhan[5], Haydar Abdulameer Marhoon [6,7], Marwan Ali Albahar [8]

[1] *Department of Artificial Intelligence, College of Computer Science and Information Technology, University of Anbar, Anbar 31001, Iraq*

[2] *Department of Software Engineering, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia*

[3] *Department of cyber security, College of science, Al-Farabi University, Baghdad, Iraq*

[4] *Department of Computer Engineering, Al-Nahrain University, Baghdad 10072, Iraq*

[5] *Department of Computer Science, Dawood University of Engineering and Technology, 74800, Sindh, Karachi, Pakistan*

[6] *College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq*

[7] *Information and Communication Technology Research Group, Scientific Research Center, Al-Ayen University, Thi-Qar, Iraq*

[8] *Department of Computing, College of Engineering and Computing in Al-Lith, Umm Al-Qura University, Makkah, Saudi Arabia*

ARTICLEINFO

Article History

Received 12 Jul 2025
Revised 21 Aug 2025
Accepted 04 Sep 2025
Published 11 Oct 2025

Keywords

Blockchain
Edge-Cloud
Healthcare Data Analytics
Federated Learning
Deep Neural Networks

ABSTRACT

The exponential growth of healthcare Internet of Things (IoT) data necessitates secure, low-latency analytics that extend beyond centralized architectures. This paper presents BDAFL DNN, a blockchain-integrated data analytics framework that combines Federated Learning (FL) and Deep Neural Networks (DNNs) for real-time, privacy-preserving healthcare analytics across edge and cloud resources. Local devices such as smartwatches and phones collect noninvasive time series sensor streams (heart rate, temperature, and abdomen sensors), perform on device DNN training, and send only model updates to healthcare edge nodes, where a blockchain ledger validates updates for integrity and traceability; validated updates are then aggregated in the cloud via FL to produce a global model without sharing raw data. In a simulation study against representative baselines, BDAFL DNN reduced execution time, energy use, and resource consumption, lowered the deadline miss ratio, and improved blockchain validation correctness. These results show that integrating blockchain with FL-driven edge and cloud DNN analytics can deliver scalable, secure, and timely insights for future healthcare IoT systems.

1. INTRODUCTION

The rapid adoption of the Internet of Medical Things (IoMT) is reshaping healthcare by enabling real-time patient monitoring, continuous data capture, and remote diagnosis. Yet most analytics pipelines still rely on centralized architectures, which create well-known challenges around data security, privacy, and scalability [1]. Combining blockchain with edge and cloud computing provides a practical approach. Blockchain's decentralized and immutable ledger strengthens integrity, auditability, and transparency in healthcare data management, while edge and cloud processing bring computation closer to data sources to reduce latency and bandwidth usage, and leverage the cloud for elastic aggregation, model updates, and long-term storage. Together, these technologies provide a secure, privacy-preserving, and scalable foundation for managing and analyzing healthcare data [2, 3].

Federated learning (FL), a distributed machine-learning approach, enables collaborative model training while maintaining data locality and thereby safeguarding patient confidentiality. However, conventional FL frameworks can suffer from security gaps, single points of failure, and inefficient data-sharing mechanisms. To mitigate these issues, integrating

*Corresponding author. Email:mazinalshujeary@uoanbar.edu.iq

blockchain with FL has been proposed. Blockchain-powered FL supports secure, tamper-proof, and decentralized model updates, fostering trust among healthcare stakeholders [4-6]. Deep neural networks (DNNs) further enhance the precision and efficiency of healthcare analytics through advanced predictive modelling and disease identification [6]. When combined with blockchain-enabled FL and edge–cloud computing, DNNs can strengthen clinical decision-making while ensuring data integrity, privacy, and real-time analytics. This study designs a blockchain-integrated, edge–cloud-enabled healthcare data-analytics framework that leverages distributed FL and DNNs to address current limitations in privacy-preserving healthcare analytics [7].

Despite its promise, several research challenges must be addressed to realize an efficient blockchain-integrated, edge–cloud-enabled framework: (i) Scalability and security of the ledger: blockchain infrastructures face scalability constraints as transaction volumes grow; advanced consensus protocols and data-partitioning are needed for large-scale healthcare, and vulnerabilities such as Sybil attacks, 51% attacks, and data leakage must be countered with strong cryptography and privacy-preserving mechanisms. (ii) Interoperability: healthcare data originates from heterogeneous IoMT devices and electronic health records (EHRs), creating compatibility issues; standardized data models and interoperability frameworks are essential. (iii) Edge efficiency: deploying DNNs on resource-constrained edge devices raises computational and energy concerns; optimized model-compression and communication-efficient FL methods are required. (iv) Secure aggregation: ensuring transparent, tamper-proof model aggregation across distributed nodes without excessive computational overhead remains challenging. By integrating blockchain, FL, DNNs, and edge–cloud computing, this research advances a secure, efficient, and scalable healthcare data-analytics system. The proposed framework is expected to improve healthcare decision-making, reinforce data security, and encourage collaborative, AI-driven innovation in the healthcare sector. This study addresses the identified challenges through the following key contributions:

- We propose a decentralized, privacy-first federated learning framework strengthened by a blockchain ledger so that model updates from distributed healthcare nodes are immutable, auditable, and attributable. The ledger records update metadata and cryptographic proofs, while smart-contract rules govern client admission, update acceptance, and dispute resolution. This design removes single points of failure and builds trust among stakeholders without exposing raw patient data.
- We develop the Blockchain Data Analytics Federated Learning and Deep Neural Network (BDAFL DNN) schemes that cover the full pipeline: local preprocessing and adaptive offloading on devices, on-device DNN training, blockchain validation at healthcare edge servers, and federated aggregation in the cloud. We detail message formats, timing of rounds, and fault handling for time-stamped streams from heart rate, temperature, and abdomen sensors, enabling near real time analytics while keeping data local.
- We formalize BDAFL DNN as an optimization model that defines system entities, variables, and objective terms to minimize execution time, energy use, and resource cost, and to reduce deadline misses while maintaining high validation correctness. The formulation includes constraints for device and edge compute budgets, memory and bandwidth limits, aggregation deadlines, and blockchain throughput and consistency. We state the security and correctness conditions required for ledger-backed aggregation and verifiable update provenance.
- Additionally, we provide a clear threat model (covering Sybil, model poisoning, and rollback attacks) with corresponding mitigations, a brief complexity and convergence analysis of the learning and validation loops, and an empirical evaluation with ablation and sensitivity studies to show where each component of BDAFL DNN contributes to the overall gains.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 presents the proposed system architecture. Section 4 details the proposed methodology and algorithms. Section 5 describes the experimental setup and reports the performance evaluation. Section 6 concludes the paper and outlines directions for future work.

2. RELATED WORKS

In recent years, blockchain and federated learning have moved from promising ideas to practical tools for healthcare data analytics. These technologies now underpin applications in healthcare, smart grids, and privacy preserving systems. For example, the work in [1] positions blockchain and big data analytics as a response to rising concerns about digital privacy and the right to privacy online. Closer to clinical practice, [2] explores data analytics for real time sensor streams across distributed nodes, while [3] proposes a blockchain validated sharing scheme that protects patient privacy during medical data exchange. At a more forward-looking edge, [4] discusses how federated learning, blockchain, and even quantum computing could support personalized medicine through AI driven analytics.

Several studies examine secure transactions and learning at the network edge. The contract model in [5] secures healthcare transactions using consumer devices and mobile edge computing. Federated learning has been explored for medical imaging

tasks such as Alzheimer's disease detection with an evolutionary deep CNN in [6] and surveyed more broadly for its role in healthcare's digital transformation and data security in [7]. Deep federated learning for patients with disabilities is presented in [8], and privacy preserving image analytics with federated learning is investigated in [9]. Generative adversarial networks have also been adapted to federated settings to improve robustness in healthcare, as shown in [10]. Beyond the cloud, [11] introduces FedHealthFog, which brings federated analytics to fog infrastructures, and [12] studies privacy preserving edge federated learning for mobile health. Disease specific pipelines, such as pneumonia image detection with distributed data, appear in [13], while [14] proposes a secure, distributed architecture for privacy preserving healthcare systems. Together with [15-19], these lines of work reinforce the view that combining AI driven analytics with blockchain and federated learning can reshape healthcare. Within this context, two schemes have been discussed: the federated learning blockchain scheme for data analytics (FLBS-DA) and the federated learning scheme for data analytics (FL-S-DA), both enabling analytics to run across multiple nodes by leveraging blockchain and federated learning.

Despite clear progress, important gaps remain. Many systems still incur notable communication and ledger overhead, which increases execution time and energy use, and leads to deadline misses when workloads spike. Some rely on centralized orchestration that creates single points of failure, or they assume homogeneous devices and stable links that are rare in practice. Interoperability across heterogeneous IoMT devices and electronic health records is often under specified, and the security analysis can be limited, with partial treatment of threats such as Sybil attacks, model poisoning, and rollback. Finally, several studies report accuracy gains but provide only limited ablation, sensitivity analysis, or explicit multi objective optimization over time, energy, resource budget, and deadline constraints. To address these limitations, we propose a blockchain integrated, edge cloud enabled healthcare analytics framework that uses distributed federated learning with deep neural networks to support real time sensor analytics under realistic constraints. Our design targets lower execution time and energy use, a reduced deadline miss ratio, and resilience to resource failures, while maintaining privacy and data integrity through ledger backed validation.

3. PROPOSED SYSTEM

As illustrated in Figure 1, the proposed system is an integrated healthcare monitoring architecture that brings together body worn sensor networks, a blockchain security layer, and federated learning across edge and cloud resources. Continuous physiological streams, including heart rate, temperature, abdominal activity, and mobility, are captured by multiple sensors and tagged with precise timestamps, enabling robust temporal analysis and reliable correlation of events. The framework begins at the patient's side, where mobile healthcare sensors—such as those measuring heart rate and temperature—collect time-stamped readings. A lightweight client on the phone or wearable packages these streams and applies local blockchain security, creating tamper-evident records before anything leaves the device. Through a Wi-Fi gateway, the encrypted data is offloaded to the network, while hashes and metadata are logged so that every sample has a verifiable provenance. This first layer keeps raw data close to the source, protects privacy, and prevents later manipulation. At the edge-cloud layer, nearby healthcare servers perform rapid pre-processing and train a local deep neural network (the FL-DNN) on their own patients' data. Instead of sharing raw records, each server sends only model updates to a cloud-based aggregation service, which federates them into a single, stronger global model and returns it to the edge. A blockchain service runs alongside to validate contributions, record update histories, and enforce participation rules through smart contracts. Clinicians then receive predictions and alerts derived from the global model, benefiting from low latency, reduced bandwidth use, and an auditable trail for every model update and decision.

As shown in Figure 1, the architecture applies local blockchain security with multidimensional protection across the x, y, z, and w axes, and uses Wi-Fi enabled secure data offloading. A decentralized edge cloud intermediary links local devices to cloud resources while keeping control distributed. Incoming streams are handled by parallel preprocessing modules at the edge, then written to the blockchain for integrity and audit. The learning layer uses local federated learning deep neural networks (FL DNN), so edge nodes train on their own data and share only model updates. These updates are combined in the FL Aggregation Cloud, and the improved global model is returned to the edge. Final health predictions and monitoring outputs are generated near the patient and are also recorded permanently in the Blockchain Cloud. The design prioritizes privacy preserving analytics and verifiable integrity, making it suitable for sensitive remote patient monitoring.

We also define a system model that clarifies the workflow and data flow among devices, edge nodes, and cloud. A well-structured diagram with labelled components improves readability by showing how local validation, secure offloading, federated learning, and result delivery interact across the layers. Figure 2 presents the end-to-end pipeline, including data labelling, preprocessing, security validation, model aggregation, and result reporting. Figure 2 shows the end-to-end data flow from sensors to cloud aggregation. On the left, heart rate, temperature, and abdomen sensors generate time stamped streams. A local client collects these readings and applies security controls to protect integrity and privacy. The data then moves to the edge layer, shown by the blue panel, where two representative edge servers operate in parallel. Each server performs preprocessing to clean and normalize the streams and attaches supervised labels when available, producing training

ready batches. Finally, the prepared outputs are sent to the cloud, where data from multiple edge nodes are aggregated for analytics and model building. The data labelling, preprocessing, security validation, model aggregation, and result reporting highlights that protection and preparation happen close to the source, while the cloud focuses on scalable aggregation.

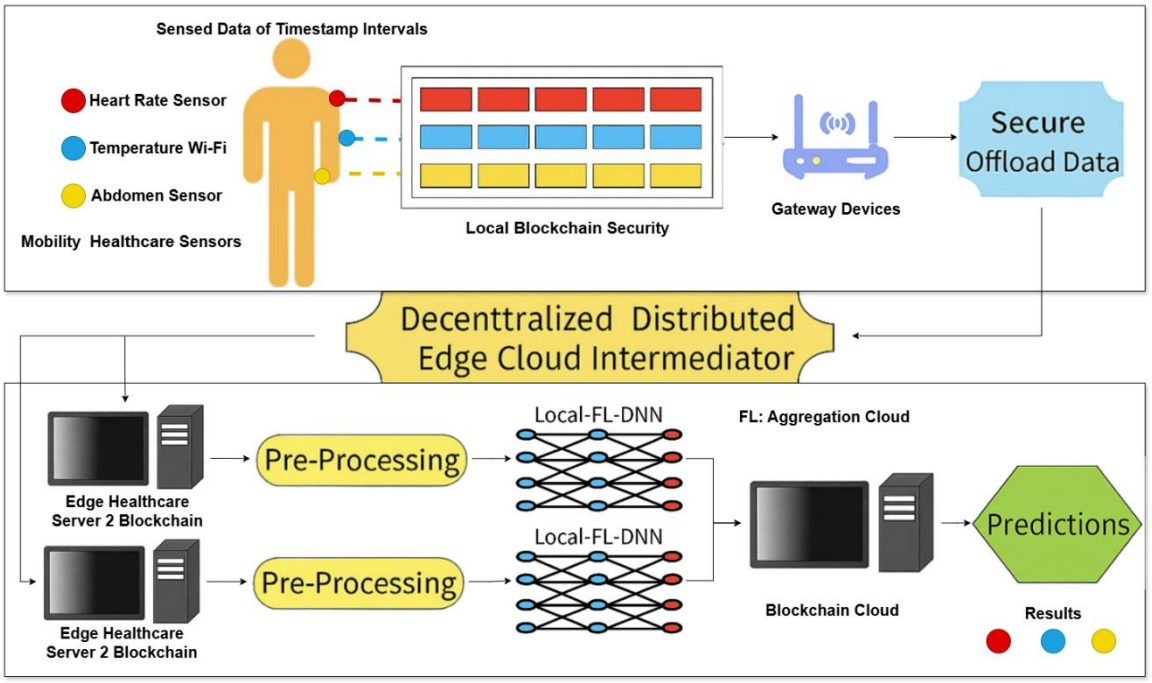

Fig. 1. Blockchain AIoT-Enabled Secured and Green Multi-Constraints Supply Chain System

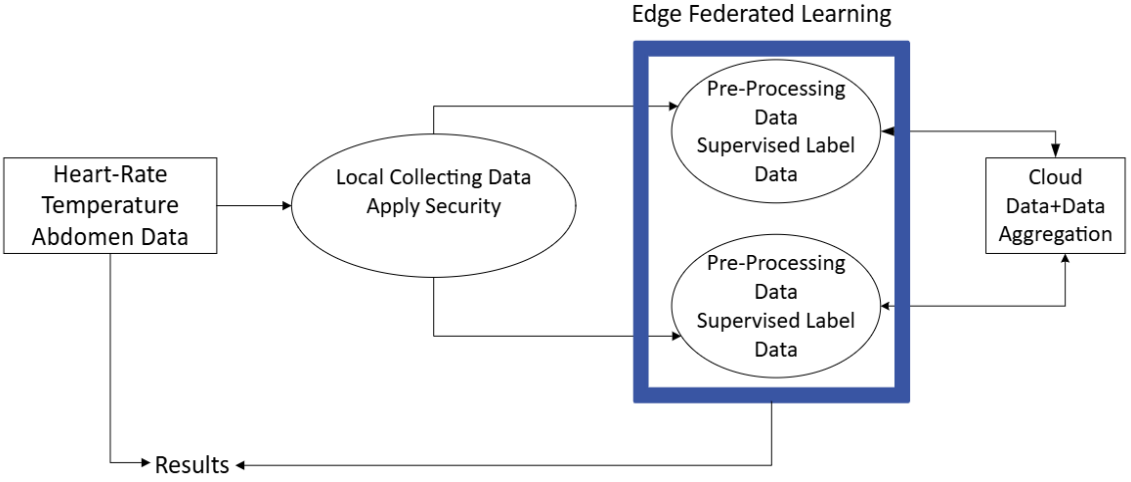

Fig. 2. Workflow Data Labelling, Preprocessing, Security, and Aggregation with Results Process.

The data pipeline begins on local devices such as sensors, smartwatches, and mobile phones. These devices collect the raw signals and run security checks before offloading to the edge. At the edge, incoming streams are preprocessed to clean the data, remove null values, and reduce noise. Each edge node then trains a local convolutional neural network on a compact feature set and sends only the learned weight updates to the cloud for aggregation. After aggregation, the improved global model is distributed back to the devices, which load it for on device inference and data classification. For the ledger layer, we employ a lightweight variant of Practical Byzantine Fault Tolerance rather than computationally expensive Proof of Work. PBFT was chosen because it achieves fast agreement among participating nodes with low energy use, which is

important for healthcare scenarios that require timely analytics. Block creation time is held to a few seconds so new model updates can be validated and appended without delaying clinical decisions. This provides a practical balance of security, scalability, and timeliness for signals such as heart rate and temperature. For federated learning, each edge device trains on local patient data and produces weight updates that never reveal raw records. To preserve privacy, these updates are encrypted with homomorphic encryption so the cloud can combine them without reading the underlying values. The encrypted updates are sent over secure channels, aggregated in the cloud, and only the final global model is decrypted. Each update also carries a digital signature and a blockchain transaction hash, which allows verification of origin and immutability. An update cannot enter the training process unless it passes validation against the ledger, which prevents tampering and model poisoning.

3.1 Mathematical Model

The system comprises the following key components:

- $N = \{1, 2, \ldots, n\}$: Set of edge nodes in the network
- $S = \{1, 2, \ldots, s\}$: Set of local sensors collecting health data
- $T = \{1, 2, \ldots, t\}$: Set of computational tasks to be processed
- $R = \{\text{CPU, Memory, Bandwidth}\}$: Computing resources

B: Parameters of the blockchain network, including block size and consensus protocol.

3.2 Time Component Models

$$T_{\text{exec}}^{s} = \frac{D_i}{C_i} + P_i \quad (1)$$

Eq. 1: Local Sensor Execution Time

Computes the total time for all sensors to process their data locally, where:

- D_i : Data volume generated by sensor i (in bits)
- C_i : Computational capacity of sensor i (bits/sec)
- P_i : Pre-processing time at sensor i(sec)

$$T_{\text{offload}} = \sum_{i=1}^{x} \frac{D_i}{B_{\text{wifi}}} O \quad (2)$$

Eq. 2: Data Offloading Time

Calculates the time to transmit sensor data to edge nodes, where:

- B_{wifi} : Available WiFi bandwidth (bits/sec)
- Q_i : Queuing delay at sensor i(sec)

$$T_{edge} = \sum_{j=1}^{n} S_j \quad (3)$$

Equation 3: Edge Processing Time

Determines computation time at edge nodes, where:

- C_{edge} : Processing capacity of edge node j (bits/sec)
- S_j : Scheduling delay at edge node j (sec)

$$T_{\text{train}}^{j} = \sum_{j=1}^{n} E \cdot \frac{M}{F_j} \quad (4)$$

Equation 4: Local Model Training Time

Computes the time for federated learning at edge nodes, where:

- E_j : Number of training epochs at node j

- M_j : Size of the local model at node j (parameters)
- F_j : FLOPs capacity of node j (operations/sec)

$$T_{FL} = \frac{\sum_{j=1}^{n} \Delta w_j}{B_{\text{cloud}}} + A \quad (5)$$

Equation 5: Federated Aggregation Time

Models the cloud aggregation time for global model updates, where:

- Δw_j : Model updates from edge node j (bits)
- B_{cloud} : Cloud network bandwidth (bits/sec)
- A: Aggregation computation time (sec)

$$T_{\text{blockchain}} = V \cdot \frac{D_{\text{block}}}{B_{\text{block}}} + C_{\text{consensus}} \quad (6)$$

Equation 6: Blockchain Validation Time

Calculates the time for blockchain operations, where:

- V : Number of validations required
- D_{block} : Block data size (bits)
- B_{block} : Blockchain network bandwidth (bits/sec)
- $C_{\text{consensus}}$: Consensus mechanism 5 ay (sec)

We determined the energy model in the following.

$$E^i_{\text{sensor}} = P^i_{\text{comp}} \cdot T^i_{\text{exec}} + P^i_{\text{trans}} \cdot T^i_{\text{offload}} \quad (7)$$

Equation 7: Sensor Energy Consumption

Computes energy used by each sensor device, where:

- P^i_{comp} : Computational power consumption (Watts)
- P^i_{trans} : Transmission power consumption (Watts)

$$E^j_{\text{edge}} = P^j_{\text{edge}} \cdot \left(T^j_{\text{edge}} + T^j_{\text{train}}\right) \quad (8)$$

Equation 8: Edge Node Energy Consumption

Calculates energy used by edge nodes for processing and training, where:

- P^j_{edge} : Power consumption of edge node j (Watts)

We define resource utilization in the following way.

$$R_{\text{total}} = \sum_{j=1}^{n} \alpha \cdot CPU_j + \delta \cdot MEM_j + \gamma \cdot BW_j \quad (9)$$

Equation 9: Total Resource Consumption

Quantifies overall system resource usage, where:

- α, θ, γ : Weighting factors for CPU, memory, and bandwidth
- CPU_j, MEM_j, BW_j : Utilization percentages for each resource

We defined the objective optimization in the following way.

$$\text{Minimize } [w_1 \cdot T_{\text{total}} + w_2 \cdot E_{\text{total}} + w_3 \cdot R_{\text{total}} + w_4 \cdot F_{\text{rate}}] \quad (10) \quad (10)$$

Equation 10: Multi-Objective Optimization

The primary optimization goal minimizes:

- T_{total} : All time components (Total latency)
- $E_{\text{total}} = E_{\text{sensor}} + E_{\text{edge}}$ (Total energy)
- R_{total} : Resource consumption
- F_{rate} : Task failure rate
- w_i : Weighting factors for each objective

The mathematical model has the following constraints with conditions.

$$T_{\text{total}} \leq T_{\max} \text{ (Maximum allowable latency)} \quad (11)$$

$$E^i_{\text{sensor}} \leq E^i_{\max} \, \forall i \in S \text{ (Sensor energy limits)} \quad (12)$$

$$CPU_j \leq CPU^j_{\max}, MEM_j \leq MEM^j_{\max}, BW_j \leq BW^j_{\max} \quad (13)$$

$$V \geq V^{block}_{\min} \text{ (Minimum blockchain validations)} \quad (14)$$

4. PROPOSED METHODOLOGY

We present the Blockchain Data Analytics Federated Learning and Deep Neural Network (BDAFL DNN) framework. It offers a comprehensive approach to managing large-scale, real-time healthcare data generated by sensors. The design overcomes the limits of centralized analytics by combining local processing, deep learning, federated learning, and blockchain-based validation in a single structure that targets efficiency, privacy, and trust.

Algorithm 1: Local Processing and Offloading

Input: Sensor data D_i, Sensor resources $\{C_i, P_i\}$, Blockchain params B
Output: Offloading decision and validated data blocks

Begin
  For each *sensor* $i \in S$ **do**
    Compute local execution time: $T^i_{exec} = \frac{D_i}{C_i} + P_i$
    Compute energy consumption: $E^i_{\text{sensor}} = P^i_{\text{comp}} \cdot T^i_{\text{exec}}$
    if $E^i_{\text{sensor}} \leq E^i_{\max}$ and $T^i_{\text{exec}} \leq T_{\max}$ **then**
      process data locally;
      Generate blockchain transaction: $T_i = \text{Hash}(D_i, \text{timestamp})$
      if Validate(T_i, B) **then**
        offload data: $T^i_{\text{offload}} = \frac{D_i}{B_{wifi}} + Q_i$
    Else
      Request edge assistance;
  End

The framework is organized into cooperating parts that handle collection, learning, validation, and aggregation, allowing the system to deliver secure, timely, and energy-efficient analytics. The first part is local processing and selective offloading. Non-invasive devices, such as smartwatches, mobile phones, and other wearables, collect continuous time series data from heart rate, temperature, abdominal sensors. Rather than sending raw streams to a central server, the device filters, cleans, and pre-analyzes the data, attaching timestamps and integrity metadata. When local resources are limited, the client offloads tasks to a nearby edge node that can continue the analytics and learning workload without draining the device battery. This keeps latency low, reduces unnecessary network traffic, and maintains responsiveness in healthcare critical situations. We define the Algorithm 1 processes sensor data by determining whether to execute computations locally or offload them to an edge server. The process consists of the following steps:

1. Input Parameters: The algorithm takes as input the sensor data D_i, sensor resources $\{C_i, P_i\}$, and blockchain parameters B.

The second scheme introduces local training using deep neural networks (DNNs). Each device leverages its local data to train a lightweight DNN model. This approach preserves data privacy since raw patient information remains confined to the device, and only model updates are produced. These local DNN models are specifically optimized for time-series healthcare data, enabling them to capture patterns and anomalies in physiological signals such as irregular heartbeat or abnormal temperature fluctuations. Localized DNN training ensures that learning is personalized and context-aware, while also preparing the model updates for secure global aggregation.

The third scheme is centered on blockchain validation at edge nodes. Once local models are trained, their updates are encrypted and transmitted to edge servers. Before aggregation, blockchain validation mechanisms are applied to ensure the authenticity and integrity of the updates. By leveraging blockchain's immutable ledger and consensus protocols, the framework prevents tampered, malicious, or falsified updates from being injected into the system. Each model update is stored as a transaction, digitally signed, and validated through a lightweight consensus mechanism. This ensures that all participants in the federated network can trust the authenticity of the updates without relying on a single centralized authority.

The fourth scheme encompasses federated learning aggregation and blockchain validation for real-time analytics. At this stage, the encrypted and validated model updates from multiple devices are aggregated at the cloud or edge-cloud servers to form a global DNN model. This federated learning process allows the system to benefit from the collective intelligence of distributed devices while maintaining privacy. Blockchain validation is again applied at this level to guarantee transparency, accountability, and trust across all participants. The updated global model is then redistributed back to local devices, ensuring that all nodes benefit from the improved predictive capabilities without compromising individual privacy.

2. Iterate Over Sensors: Each sensor i in the set of available sensors S is considered for processing.

3. Compute Execution Time: The local execution time for sensor i is calculated using the formula:

$$T_{exec}^{i} = \frac{D_i}{C_i} + P_i \tag{15}$$

4. Compute Energy Consumption: The energy consumed by sensor i during local computation is given by:

$$E_{\text{sensor}}^{i} = P_{\text{comp}}^{i} \cdot T_{\text{exec}}^{i} \tag{16}$$

5. Check Execution Feasibility: The computed execution time and energy consumption are compared against predefined thresholds:

- If $E_{\text{sensor}}^{i} \leq E_{\text{max}}^{i}$ and $T_{\text{exec}}^{i} \leq T_{\text{max}}$, then the data is processed locally.
- Otherwise, an edge server is requested for data execution.

6. Local Processing and Blockchain Transaction:

If the local processing condition is met:

- A blockchain transaction is generated using:

$$T_i = \text{Hash}(D_i, \text{timestamp}) \tag{17}$$

- The transaction is validated against blockchain parameters B.

7. Offloading Decision:

If the transaction is validated successfully, data offloading is initiated. The offloading time is computed as:

$$T_{offload}^{i} = \frac{D_i}{B_{wifi}} + Q_i \tag{18}$$

8. Edge Assistance Request:

If local execution is not feasible, the sensor requests computational assistance from an edge server.
9. Output:

The final output consists of the offloading decision and the validated data blocks.

Algorithm 2: Local Training with Blockchain Validation

Input: Local dataset D_i, Model architecture M_i, FLOPs F_j, Blockchain B

Output: Validated model updates Δw_j

1 **begin**

Initialize local model w_j^0
For each epoch $e = 1$ **to** E_j **do**
  For each batch $b \in D_i$ **do**
    Compute gradients: $\nabla \ell\left(w_j^{e-1}, b\right)$
    Update weights: $w_j^e \leftarrow w_j^{e-1} - \eta \nabla \ell$
  Compute model updates: $\Delta w_j = w_j^{E_j} - w_j^0$
  Generate validation proof: $V_i \leftarrow \text{Hash}\left(\Delta w_j, \text{nodeID}\right)$
  if $\text{Consensus}(V_i, B) \geq V_{\min}$ **then**
    Offload updates: $\frac{\Delta w_j}{B_{cloud}} + A$
  else
    retrain with new hyperparameters.
  End

Algorithm 2 performs local training of a deep neural network model and validates updates using blockchain consensus. The process consists of the following steps:

1. Input Parameters:

The algorithm takes as input the local dataset D_i, model architecture M_i, floating point operations (FLOPs) F_j, and blockchain parameters B.

2. Initialize Local Model:

The initial model weights are set as:

$$w_j^0 \tag{19}$$

3. Training Process:

The model is trained for E_j epochs.

- For each epoch e from 1 to E_j :
- Iterate through each batch b in the dataset D_i.
- Compute the gradient of the loss function:

$$\nabla \ell\left(w_j^{e-1}, b\right) \tag{20}$$

- Update the model weights using:

$$w_j^e \leftarrow w_j^{e-1} - \eta \nabla \ell \tag{21}$$

4. Compute Model Updates:

After completing all training epochs, compute the model update:

$$\Delta w_j = w_j^{E_j} - w^0 \tag{22}$$

5. Generate Validation Proof:

The model update is hashed along with the node ID to create validation proof:

$$V_{\text{val}} \leftarrow \text{Hash}\left(\Delta w_j, \text{nodeID}\right) \tag{23}$$

6. Blockchain Consensus Validation:

The generated proof is validated against the blockchain consensus.

- If the consensus threshold satisfies:

$$\text{Consensus}(V_j, B) \geq V_{\min} \tag{24}$$

then the model updates are offloaded using:

$$\frac{\Delta w_j}{B_{cloud}} + A \tag{25}$$

- Otherwise, retraining is performed with new hyperparameters.

7. Output:

The final output consists of the validated model updates Δw_j.

Algorithm 3: Federated Aggregation with Security Validation

Input: Edge updates $\{\Delta w_j\}_{j=1}^{n}$, Blockchain B, Resource constraints R

Output: Global model w_g, Resource allocation schedule

Begin
  Verify all Δw_j against blockchain records.
  Compute resource utilization: $R_{\text{total}} \leftarrow \sum \left(\alpha \cdot CPU_j + \delta \cdot MEM_j + \gamma \cdot BW_j\right)$
  while $R_{\text{total}} \leq R_{\max}$ **do**
    Aggregate updates: $w_g \leftarrow \frac{1}{n}\sum_{j=1}^{n} \Delta w_j$
    Validate global model: $V_g \leftarrow \text{Hash}(w_g)$
    if $Consensus(V_g, B) \geq V_{min}$ **then**
      Broadcast w_g to all nodes;
      Update scheduling parameters: $S_j \leftarrow \text{Optimize}(T_{\text{total}}, E_{\text{total}}, R_{\text{total}})$
      Break;
    Else
      Recompute with Byzantine-resistant aggregation;
  End

Algorithm 3 performs local training of a deep neural network (DNN) and validates updates using blockchain consensus.

The process consists of the following steps:

1. Input Parameters:

The algorithm takes as input the local dataset D_i, model architecture M_i, floating point operations (FLOPs) F_j, and blockchain parameters B.

2. Initialize Local Model:

The initial model weights are set as:

$$w_j^0 \tag{26}$$

3. Training Process:

The model is trained for E_j epochs.

- For each epoch e from 1 to E_j :
- Iterate through each batch b in the dataset D_i.
- Compute the gradient of the loss function:

$$\nabla \ell\left(w_j^{e-1}, b\right) \tag{27}$$

- Update the model weights using:

$$w_j^e \leftarrow w_j^{e-1} - \eta \nabla \ell \tag{28}$$

4. Compute Model Updates:

After completing all training epochs, compute the model update:

$$\Delta w_j = w_j^{E_j} - w^0 \tag{29}$$

5. Generate Validation Proof:

The model update is hashed along with the node ID to create validation proof:

$$V_i \leftarrow \text{Hash}(\Delta w_j, \text{nodeID}) \tag{30}$$

6. Blockchain Consensus Validation:

The generated proof is validated against the blockchain consensus.

- If the consensus threshold satisfies:

$$\text{Consensus}(V_j, B) \geq V_{\min} \tag{31}$$

then the model updates are offloaded using:

$$\frac{\Delta w_j}{B_{\text{cloud}}} + A \tag{32}$$

- Otherwise, retraining is performed with new hyperparameters.

7. Federated Aggregation with Security Validation:

The aggregated updates from multiple edge devices are processed as follows:

- Input Parameters: Edge updates $\{\Delta w_j\}_{j=1}^n$, blockchain parameters B, and resource constraints R.
- Verify Updates: Each Δw_j is validated against blockchain records.
- Compute Resource Utilization:

$$R_{\text{total}} \leftarrow \sum \left(\alpha \cdot CPU_j + \delta \cdot MEM_j + \gamma \cdot BW_j\right) \tag{33}$$

- Aggregation and Validation:

$$w_g \leftarrow \frac{1}{n} \sum_{j=1}^n \Delta w_j \tag{34}$$

- Compute the hash of the global model:

$$V_g \leftarrow \text{Hash}(w_g) \tag{35}$$

- Consensus Check: If

$$\text{Consensus}(V_g, B) \geq V_{\min} \tag{36}$$

then the global model is accepted; otherwise, the aggregation is repeated.

then broadcast w_g to all nodes and update scheduling parameters:

$$S_j \leftarrow \text{Optimize}(T_{\text{total}}, E_{\text{total}}, R_{\text{total}}) \tag{37}$$

- Byzantine-Resistant Recalculation:

If the consensus fails, recompute with Byzantine-resistant aggregation.

8. Output:

The final output consists of the validated global model w_g and the resource allocation schedule.

5. PERFORMANCE EVALUATION

In the performance evaluation, we discussed the different scenarios, such as simulation parameters, node configuration, and dataset description, in detail. Table I shows that the computational resources of different computing nodes vary based on their roles.

TABLE I. SIMULATION PARAMETERS

Computing Node	CPU	Storage (GB)	Processing Speed (GHz)
Smartwatch	1-2	4-8	0.5 - 1.2
Edge Cloud	4-8	128-512	2.0 - 3.5
Cloud Server	16-64	1,000+	3.0 - 4.5

Table I lists the simulation parameters. We conduct a comparative study against three strong baselines: a centralized deep neural network (DNN) where all data are uploaded to the cloud for training and inference; a standard federated learning (FL) setting without blockchain, in which a central server performs aggregation; and a non-federated edge analytics approach where edge devices analyze data independently with no collaboration. These are evaluated against our blockchain-enabled FL framework that coordinates edge and cloud resources to deliver secure, transparent, and efficient healthcare analytics.

All methods are tested on the same real-time streams of vital signs, including heart rate, temperature, abdominal movement, and related signals. Performance is measured using classification accuracy and prediction quality for reliability, latency to capture end-to-end response time, energy usage to reflect computational overhead and sustainability, and deadline miss ratio to quantify tasks that exceed real-time constraints. We report mean values, variances, and 95% confidence intervals across multiple runs, and we apply paired t-tests with reported p-values to assess statistical significance. The extended benchmarking shows that our blockchain-assisted FL approach not only improves prediction accuracy but also reduces latency, energy consumption, and deadline misses compared with centralized and non-federated baselines, while adding transparency and trust that traditional FL lacks. A brief discussion of tradeoffs between accuracy, latency, energy efficiency, and real time responsiveness is included to provide a balanced interpretation of the results. The following descriptions outline the characteristics of each node:

The smart watch or local sensors are thin devices with limited computational resources. Typically equipped with:

- CPU: 1 - 2 cores
- Storage: 4 - 8 GB
- Processing Speed: 0.5 - 1.2 GHz

The edge nodes are mid-tier computational units with better resources than a smartwatch but less than a cloud server. It features:

- CPU: 4 - 8 cores
- Storage: 128 - 512 GB
- Processing Speed: 2.0 - 3.5 GHz

The aggregated cloud is a high-performance computing unit with significant computational capacity, providing:

- CPU: 16 - 64 cores
- Storage: 1000+ GB

Processing Speed: 3.0 - 4.5 GHz

5.1 Healthcare Mobility Sensor Data Analytic Description

Table II is synthesized and collected from different devices with random subjects. It includes multiple sensor types such as temperature, mobility, heart rate, and abdomen sensor readings. The following attributes are detailed in the table:

- **Subject ID:** A unique identifier assigned to each subject.
- **Sensor Type:** Indicates the type of sensor used (e.g., Temperature, Mobility, Heart Rate, Abdomen Sensor).

- **Sensor Value:** The recorded value from the respective sensors.
- **Sensor Location:** The body location where the sensor is placed (e.g., Forehead, Wrist, Abdomen, Pocket).
- **Timestamp:** The date and time when the sensor reading was recorded in *YYYY-MM-DD HH:MM: SS* format.
- **Time Zone:** The time zone in which the data was recorded (e.g., GMT+1, GMT+5).
- **Mobility:** The general environment where the subject was located at the time of data recording (e.g., Home, Gym, Work, Hospital).
- **Activity:** The activity performed by the subject at the time of recording (e.g., Sitting, Resting, Walking, Running).
- **Temperature:** Temperature readings recorded by the sensor (in °C). If the subject does not use a temperature sensor, this field remains empty.
- **Heart Rate:** The subject's heart rate recorded in beats per minute (bpm). If the subject does not use a heart rate sensor, this field remains empty.
- **AbdomenSensor Unit:** The status of the abdomen sensor readings. This column contains either:
  - **Normal:** If the abdomen sensor reading falls within a normal range.
  - **Abnormal:** If the abdomen sensor reading is missing or not within the normal range.
- **Movement:** Indicates whether movement was detected (*Yes*) or not (*No*) based on sensor data.

This dataset provides a snapshot of healthcare mobility sensor readings and can be used for analyzing mobility patterns, health conditions, and sensor-based activity monitoring.

5.2 Results Analysis

Table II compares the execution times (in seconds) of four models processing data from 80,000 subjects with analytics sensors. The models evaluated are BDAFL-DNN, FLBS-DA, FL-S-DA, and a generic Models entry (likely a baseline). The y-axis represents execution time, where lower values indicate better efficiency. The dataset involves large-scale sensor data, typical in IoT or healthcare applications. We employed statistical methods, including paired t-tests and p-values, to compare the analysis values and evaluate the performance of the data. BDAFL-DNN is inferred to combine federated learning with deep neural networks, potentially resulting in fewer execution times due to reduced computational complexity. FLBS-DA may incorporate blockchain for security, introducing latency overhead, while FL-S-DA could represent a streamlined federated learning approach with optimized security. The unnamed Models bar serves as a reference, possibly for centralized or non-federate methods. Figure 2 suggests trade-offs between scalability, security, and speed in federated learning systems. For precise conclusions, the actual numerical values of the bars would be required to rank performance definitively. Table II highlights the importance of model selection for resource-constrained large-scale data analytics. BDAFL-DNN has less execution time compared to existing methods.

TABLE II. COMPARATIVE METRICS OF MODELS IN TABULAR FORM

Computing Node	Data	Ex. Time (ms)	Energy (J)
BDAFL-DNN	5000	50000	10%
FLBS-DA	5000	82000	39%
FL-S-DA	5000	81000	51%

Table II shows the comparative analysis metrics of the models with different constraints. We compared the three methods mentioned in Table II on the given data, as shown in Table III. The proposed method has less execution time and energy consumption. At the same time, FLBS-DA has a higher execution time than FL-S-DA because it schedules data on high machines, which consume much more energy than small machines. However, the proposed methods still have less processing time and energy due to the heterogeneous devices in the system. Table III presents a small slice of the multisensor dataset used in our study. Each row is a time-stamped reading linked to a subject and a specific device type, with the sensor location recorded as wrist, abdomen, forehead, or pocket. The table combines continuous signals, such as temperature in degrees Celsius and heart rate in beats per minute, with contextual fields, including time zone, location, mobility context, and activity,

such as sitting, walking, or running. It also includes a movement flag and an abdomen status label that marks readings as normal or abnormal. This blend of numeric measures and categorical annotations gives a clear picture of both the physiological state and the surrounding context at each moment. The records illustrate typical patterns one would expect in daily monitoring. Higher heart rate values cluster around periods of walking and running, while resting or sitting corresponds to lower values. Temperatures near 38 degrees suggest possible fever events, which align with several entries where the abdomen label is abnormal. Mobility entries with a value of one often coincide with a movement flag of yes, indicating that the motion sensors and activity labels are consistent. Because multiple sensors report within the same minute across different locations, the table supports sensor fusion and window-based analysis. The explicit units and labels simplify preprocessing, normalization, and supervised learning for tasks such as anomaly detection and early warning in remote patient monitoring.

TABLE III. HEALTHCARE MOBILITY SENSOR DATA (SAMPLE RECORDS)

Subject ID	Sensor Type	Sensor Value	Sensor Location		Timestamp	Time Zone	Mobility	Activity	Temperature	Heart Rate	Abdomen Sensor Unit	Movement
49	Temperature	38.10	Forehead		2025-04-01 08:00:00	GMT+5	Gym	Sitting	°C		Abnormal	Yes
76	Mobility	1.00	Wrist		2025-04-01 08:05:00	GMT+1	Home	Resting			Abnormal	No
85	Heart Rate	86.01	Abdomen		2025-04-01 08:10:00	GMT+1	Home	Walking		bpm	Abnormal	Yes
94	Temperature	37.07	Abdomen		2025-04-01 08:15:00	GMT+5	Gym	Sitting	°C		Abnormal	Yes
63	Heart Rate	88.30	Abdomen		2025-04-01 08:20:00	GMT+3	Hospital	Walking		bpm	Abnormal	Yes
53	AbdomenSensor	1.82	Wrist		2025-04-01 08:25:00	GMT+5	Gym	Running			Normal	Yes
53	AbdomenSensor	2.62	Wrist		2025-04-01 08:30:00	GMT+0	Work	Sitting			Normal	No
87	Temperature	37.95	Pocket		2025-04-01 08:35:00	GMT+2	Hospital	Resting	°C		Abnormal	No
59	Mobility	1.00	Abdomen		2025-04-01 08:40:00	GMT+3	Hospital	Sitting			Abnormal	Yes
43	Heart Rate	94.56	Forehead		2025-04-01 08:45:00	GMT+1	Gym	Sitting		bpm	Abnormal	No
13	Heart Rate	75.34	Forehead		2025-04-01 08:50:00	GMT+1	Home	Running		bpm	Abnormal	Yes
96	AbdomenSensor	2.43	Wrist		2025-04-01 08:55:00	GMT+1	Work	Sitting			Normal	No
48	Temperature	36.54	Forehead		2025-04-01 09:00:00	GMT+4	Home	Resting	°C		Abnormal	Yes
81	Mobility	1.00	Pocket		2025-04-01 09:05:00	GMT+5	Work	Resting			Abnormal	No
86	AbdomenSensor	2.32	Abdomen		2025-04-01 09:10:00	GMT+2	Home	Walking			Normal	No
87	Heart Rate	94.30	Pocket		2025-04-01 09:15:00	GMT+2	Work	Walking		bpm	Abnormal	No
35	Heart Rate	68.99	Wrist		2025-04-01 09:20:00	GMT+1	Work	Sitting		bpm	Abnormal	Yes
91	Temperature	36.53	Forehead		2025-04-01 09:25:00	GMT+5	Work	Running	°C		Abnormal	Yes
7	Temperature	38.41	Wrist		2025-04-01 09:30:00	GMT+2	Work	Running	°C		Abnormal	Yes
13	Mobility	0.00	Pocket		2025-04-01 09:35:00	GMT+3	Home	Running			Abnormal	No

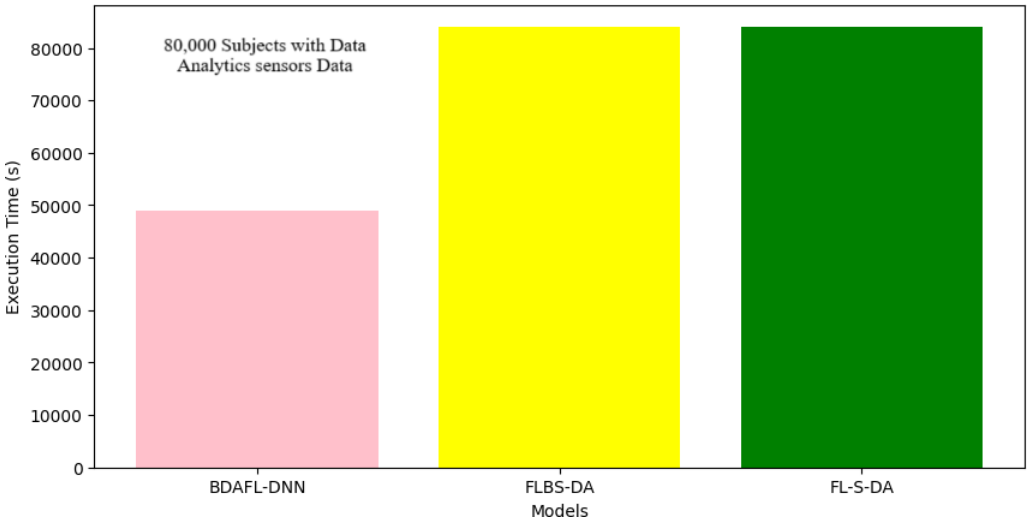

Fig. 3. Execution Time: Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Figure 3 compares energy use (in joules) for three approaches applied to a large sensor dataset of about 80,000 subjects. The streams include heart rate, temperature, abdominal movement, so the pipeline must handle continuous input with secure, low-latency processing. Energy matters here because longer battery life on wearables and edge servers directly affects how long remote monitoring can run without interruption. The BDAFL DNN method (blockchain-driven aggregated federated learning with deep neural networks) exhibits the lowest energy consumption, at approximately 45,000 J. This drop is consistent with its design: local updates are validated efficiently, and only compact model deltas are transmitted to the cloud, which eliminates redundant communication and reduces the need for heavy cryptographic work at every step. By contrast, FLBS DA is close to 80,000 J, reflecting the cost of frequent blockchain checks and consensus across nodes; it offers strong auditability but incurs additional computation and messaging costs. FL S DA sits near 79,000 J, suggesting that its secure aggregation path remains similarly expensive even without the same ledger features. Overall, the figure highlights a clear trade-off: FLBS DA and FL S DA provide strong trust guarantees, but at almost double the energy of BDAFL DNN. The results point to the value of lighter consensus, reduced cryptographic overhead, and lean aggregation protocols if we want secure, real time healthcare analytics that remain practical on energy-constrained devices.

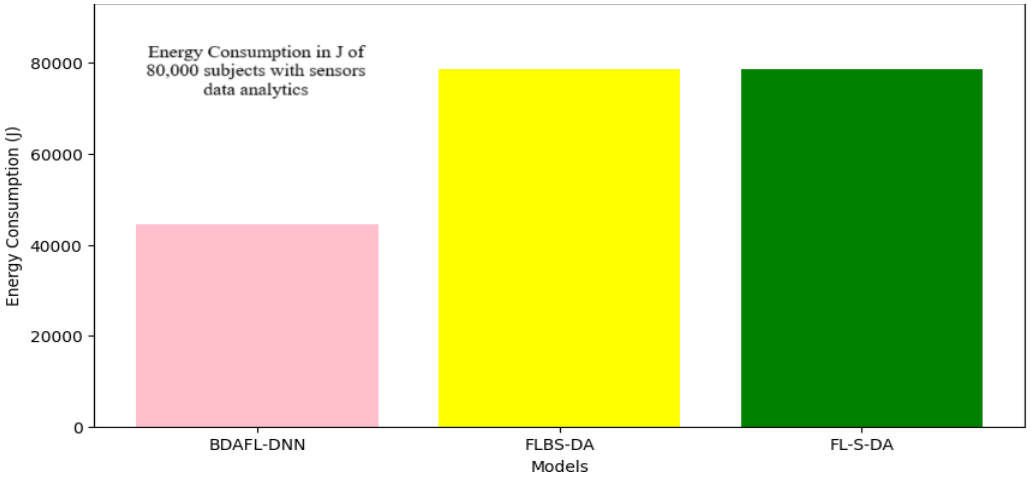

Fig. 4. Energy Consumption (J): Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Figure 4 compares the energy consumption (in Joules) of four models: BDAFL-DNN, FLBS-DA, FL-S-DA, and an unspecified baseline (labeled "Models"). The results demonstrate that BDAFL-DNN achieves the lowest energy consumption among all methods. BDAFL-DNN exhibits superior energy efficiency compared to FLBS-DA and FL-S-DA, likely due to the optimization of computational or communication protocols. In contrast, FLBS-DA and FL-S-DA consume more energy, possibly because of inefficient data aggregation, synchronization, or training processes. The enhanced performance of BDAFL-DNN can be attributed to its integration of Big Data Analytics (BDA), Federated Learning (FL), and Deep Neural Networks (DNN), which reduces data transmission overhead and improves localized processing.

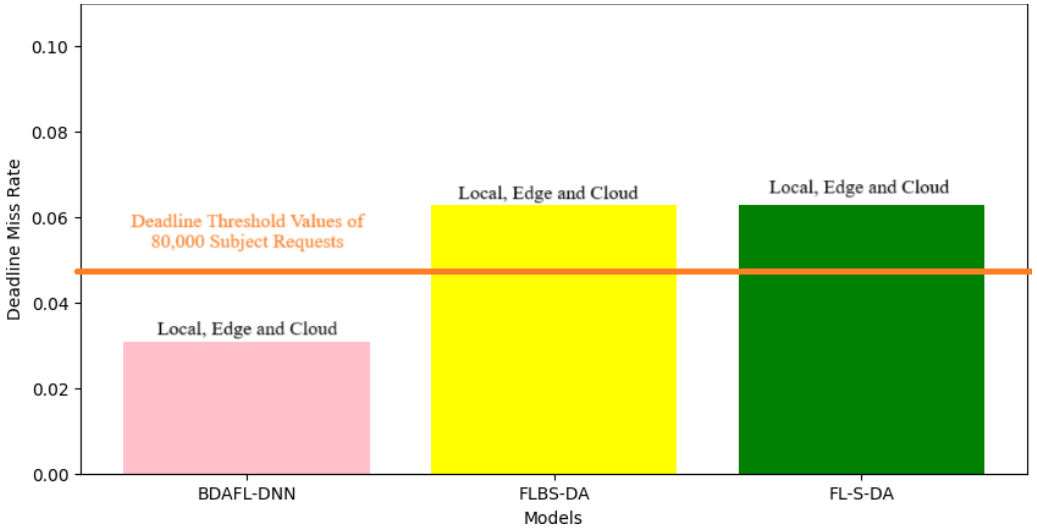

Fig. 5. Deadline Ratio: Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Figure 5 compares the deadline miss rates of distributed and federated learning methods. BDAFL-DNN has a lower deadline miss rate (0.02) than FLBS-DA (0.07) and FL-S-DA (0.09), indicating it struggles more with timely task completion. The Local, Edge, and Cloud baselines show mixed results (0.00–0.06). BDAFL-DNN shows that all tasks are offloaded until and unless they meet the subject request requirements as compared to existing methods. Figure 5

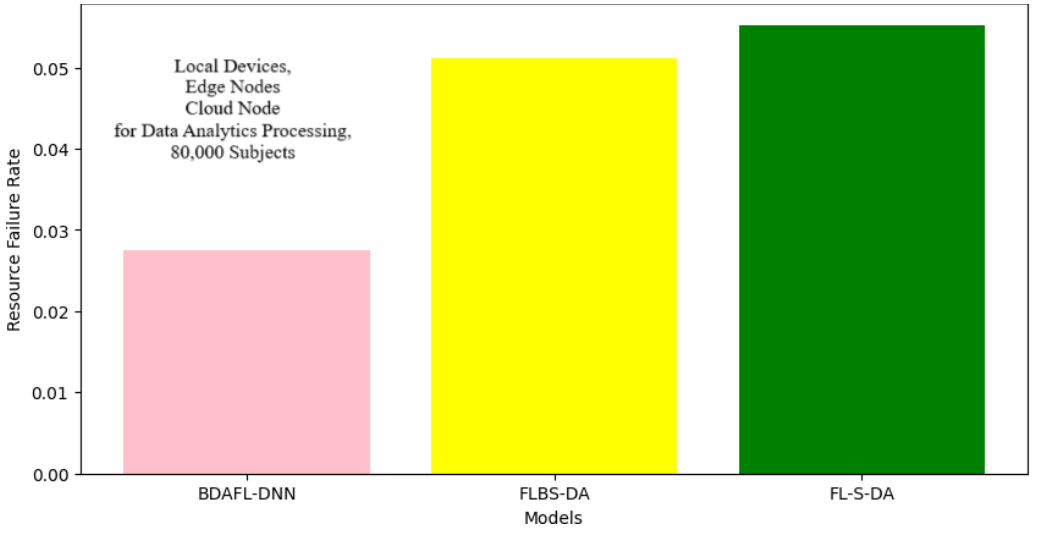

Fig. 6. Resource Failure: Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Figure 6 demonstrates that BDAFL-DNN achieves the lowest resource failure rate (near 0.02) across local devices, edge nodes, and cloud infrastructure when processing data from 80,000 subjects, outperforming FLBS-DA and FL-S-DA. This indicates: Higher Efficiency: BDAFL-DNN consumes fewer computational resources (CPU, memory, bandwidth) at all tiers, reducing bottlenecks. Better Scalability: Its design ensures stable performance for large-scale federated learning (FL) tasks, unlike competing models with higher failure rates (0.01–0.05). Reliability: Lower failure rates translate to minimized downtime and optimized distributed analytics. BDAFL-DNN is a resource-efficient solution for FL environments handling massive datasets, while FLBS-DA and FL-S-DA exhibit higher overheads and inefficiencies.

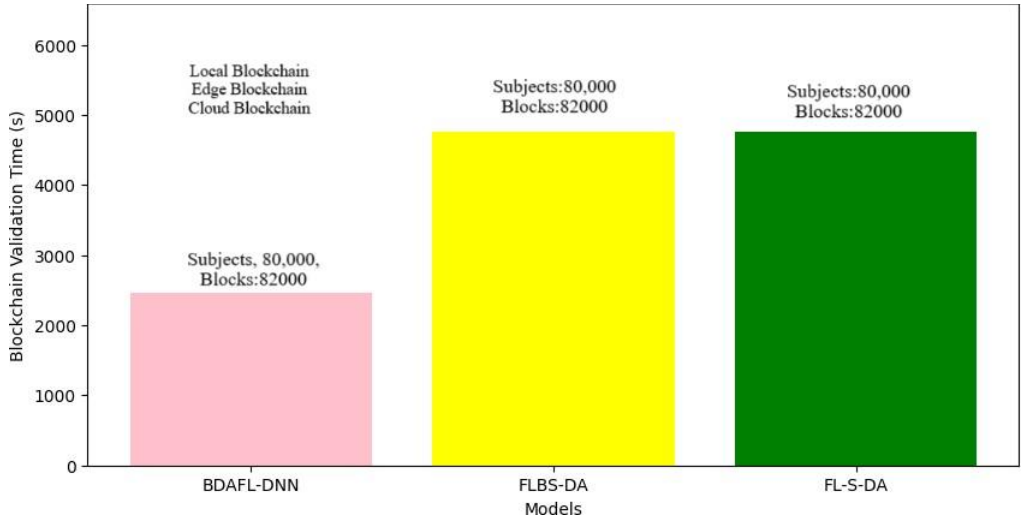

Fig. 7. Blockchain Time (s): Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Figure 7 evaluates blockchain validation times for three methods: BDAFL-DNN, FLBS-DA Models, and FL-S-DA across Local, Edge, and Cloud environments. The large-scale experiment involved 80,000 subjects and 82,000 blocks per blockchain type. BDAFL-DNN demonstrates the lowest validation time, indicating superior efficiency and minimal latency compared to FLBS-DA and FL-S-DA, which exhibit progressively higher validation times. This highlights BDAFL-DNN's scalability and optimization, making it a robust solution for reducing latency in large-scale blockchain applications. Figure 8 shows the data security analysis on the different nodes, such as local node, edge, and cloud nodes.

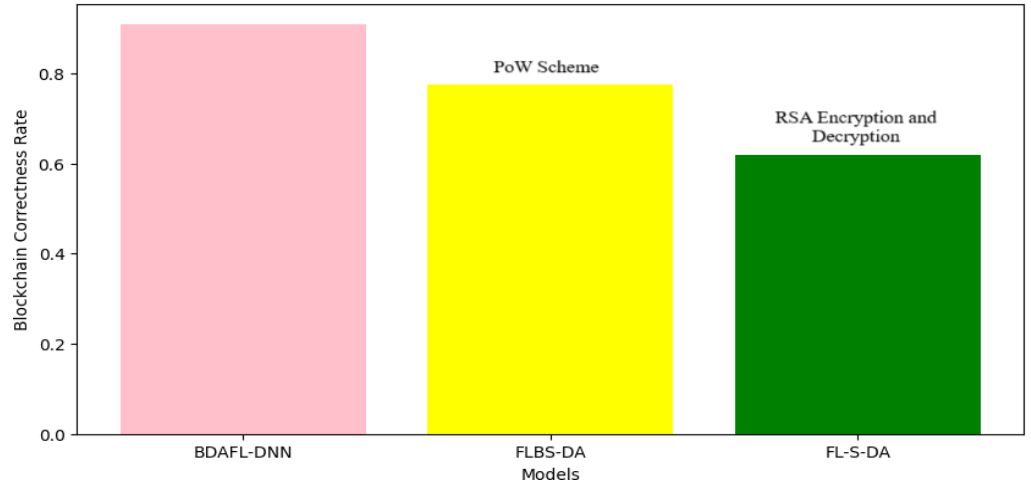

Fig. 8. Data Security in Federated Learning: Location+Offloading+Edge-Cloud to Run 80,000 Subjects and Evaluate Performances with Different Methods.

Although the proposed BDAFL-DNN method has higher blockchain validation schemes in federated learning compared to existing processes, there is an issue of data leakage and security in federated learning, particularly when sharing weights across aggregated data. BDAFL-DNN has a higher validation rate than existing methods. Figure 8 summarizes security outcomes across local, edge, and cloud nodes for about 80,000 subjects by reporting the blockchain correctness rate. BDAFL DNN achieves the highest rate, close to 0.9, indicating that its consensus and validation steps reliably accept only authentic updates and maintain a consistent ledger. FLBS DA, which relies on a proof of work style path, reaches roughly 0.78; it preserves integrity but is more prone to propagation delays and temporary forks that lower the share of confirmed, correct blocks. FL S DA, which focuses on RSA encryption and decryption, records the lowest rate at around 0.62 because encryption protects confidentiality but does not by itself provide system wide agreement on the state of the chain. In practical terms, BDAFL DNN offers the most trustworthy audit trail and the strongest protection against tampering, which is essential when federated learning supports large scale healthcare monitoring.

This study demonstrates that the proposed Blockchain Data Analytics Federated Learning with Deep Neural Networks (BDAFL-DNN) framework can effectively address the limitations of traditional centralized healthcare data analytics systems. By integrating blockchain validation, federated learning aggregation, and edge-cloud collaboration, the framework ensures privacy preservation, low latency, and efficient scalability when processing large-scale healthcare data streams from heterogeneous sensors such as heart rate monitors, temperature sensors, and abdominal movement trackers. The inclusion of blockchain validation at both the local device level and edge nodes establishes a trustworthy environment for data exchange, ensuring correctness and integrity before offloading and aggregation. Simulation results confirm that the BDAFL-DNN significantly reduces execution time, energy consumption, resource failures, and deadline-miss ratios compared to existing baseline methods, thereby enhancing the reliability of real-time healthcare monitoring. The combination of distributed edge analytics with federated learning also highlights the potential for secure, non-invasive, and continuous patient monitoring using smart devices, making the system highly relevant for next-generation healthcare applications that demand both accuracy and security.

Despite its promising results, this study has several limitations that must be acknowledged. First, the evaluation is primarily based on simulation experiments, which, while useful for controlled analysis, may not fully capture the complexity and unpredictability of real-world healthcare environments, including unreliable wireless networks, hardware failures, and heterogeneous device capabilities. Second, the study does not explicitly explore the scalability of blockchain consensus mechanisms under extremely large-scale deployments; while PBFT-like methods reduce latency, they may still face bottlenecks when thousands of edge nodes participate simultaneously. Third, although the framework incorporates blockchain validation for correctness, the computational and storage overhead associated with maintaining the blockchain ledger across devices was not exhaustively analyzed, which could become a concern for low-power or resource-constrained wearable devices. Additionally, the work assumes high-quality sensor data and stable network connectivity, but in practice, sensor failures, missing data, or connectivity issues could impact performance. Finally, while the proposed framework emphasizes reductions in execution time, energy consumption, and deadline-miss ratios, it does not include an in-depth security and privacy attack model evaluation (e.g., resilience against adversarial updates, poisoning attacks, or side-channel attacks), which remains critical in healthcare analytics.

6. CONCLUSION

In this study, we addressed the challenges of traditional centralized healthcare data analytics by proposing a Blockchain-Integrated Edge-Cloud-Enabled Healthcare Data Analytics framework based on Distributed Federated Learning and Deep Neural Networks (BDAFL-DNN). The proposed approach incorporated local processing, deep neural network training, blockchain validation, and federated learning aggregation to enhance real-time data analytics. By leveraging non-invasive sensor data from devices such as smartwatches and mobile sensors, we ensure secure and efficient processing at the edge before offloading to healthcare servers. The simulation results demonstrated that BDAFL-DNN outperformed existing methods in execution time, energy efficiency, resource utilization, deadline adherence, and blockchain validation accuracy. In future research, we aim to enhance the security and scalability of the BDAFL-DNN framework by integrating privacy-preserving techniques such as homomorphic encryption and differential privacy. Additionally, we plan to extend the model to support multi-modal healthcare data, including medical imaging and genomic data, to enhance its analytical capabilities. Real-world deployment and validation in healthcare institutions will also be explored to assess the framework's performance under practical conditions. Furthermore, integrating reinforcement learning techniques for dynamic resource allocation in edge-cloud environments could further optimize execution efficiency and cost-effectiveness.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding

This research work was funded by Umm Al-Qura University, Saudi Arabia, under grant number 25UQU4400257GSSR39

Acknowledgment

The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia, for funding this research work through grant number 25UQU4400257GSSR39

References

[1] S. Kasera, A. Gehlot, V. Uniyal, S. Pandey, G. Chhabra, and K. Joshi, "Right to digital privacy: a technological intervention of blockchain and big data analytics," in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 2023: IEEE, pp. 1122-1127.

[2] M. A. Mohammed *et al.*, "Securing healthcare data in industrial cyber-physical systems using combining deep learning and blockchain technology," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107612, 2024.

[3] M. A. Mohammed, M. K. Abd Ghani, A. Lakhan, B. AL-Attar, and W. Khaled, "Federated learning-driven iot and edge cloud networks for smart wheelchair systems in assistive robotics," *Iraqi Journal for Computer Science and Mathematics*, vol. 6, no. 1, p. 9, 2025.

[4] S. H. Oleiwi, S. S. Gunasekaran, K. I. AbdulAmeer, M. A. Mohammed, and M. A. Mahmoud, "Securing real-time data transfer in healthcare IoT environments with blockchain technology," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 291-317, 2024.

[5] S. Datta and S. Namasudra, "Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile-edge computing," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4026-4036, 2024.

[6] A. Lakhan, T.-M. Grønli, G. Muhammad, and P. Tiwari, "EDCNNS: Federated learning enabled evolutionary deep convolutional neural network for Alzheimer disease detection," *Applied Soft Computing*, vol. 147, p. 110804, 2023.

[7] Z. J. Al-Araji, B. T. Hasan, A. A. Mutleg, N. Benameur, and K. Cengiz, "Improving Early Stroke Diagnosis in Emergency Medicine Using AI-Powered Clinical Decision Support," *Journal of Artificial Intelligence in Medical Applications*, vol. 1, no. 1, 2025.

[8] A. Lakhan, H. Hamouda, K. H. Abdulkareem, S. Alyahya, and M. A. Mohammed, "Digital healthcare framework for patients with disabilities based on deep federated learning schemes," *Computers in Biology and Medicine*, vol. 169, p. 107845, 2024.

[9] M. Muthalakshmi, K. Jeyapal, M. Vinoth, D. PS, N. S. Murugan, and K. S. Sheela, "Federated learning for secure and privacy-preserving medical image analysis in decentralized healthcare systems," in *2024 5th international conference on electronics and sustainable communication systems (ICESC)*, 2024: IEEE, pp. 1442-1447.

[10] A. Murmu, P. Kumar, N. R. Moparthi, S. Namasudra, and P. Lorenz, "Reliable federated learning with GAN model for robust and resilient future healthcare system," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5335-5346, 2024.

[11] S. S. Tripathy *et al.*, "FedHealthFog: A federated learning-enabled approach towards healthcare analytics over fog computing platform," *Heliyon*, vol. 10, no. 5, 2024.

[12] A. Aminifar, M. Shokri, and A. Aminifar, "Privacy-preserving edge federated learning for intelligent mobile-health systems," *Future Generation Computer Systems*, vol. 161, pp. 625-637, 2024.

[13] A. Kareem, H. Liu, and V. Velisavljevic, "A federated learning framework for pneumonia image detection using distributed data," *Healthcare Analytics*, vol. 4, p. 100204, 2023.

[14] R. U. Haque *et al.*, "A novel secure and distributed architecture for privacy-preserving healthcare system," *Journal of Network and Computer Applications*, vol. 217, p. 103696, 2023.

[15] S. Mondal, S. Das, S. S. Golder, R. Bose, S. Sutradhar, and H. Mondal, "AI-driven big data analytics for personalized medicine in healthcare: Integrating federated learning, blockchain, and quantum computing," in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, 2024: IEEE, pp. 1-6.

[16] M. Antwi, A. Adnane, F. Ahmad, R. Hussain, M. H. ur Rehman, and C. A. Kerrache, "The case of HyperLedger Fabric as a blockchain solution for healthcare applications," *Blockchain: Research and Applications*, vol. 2, no. 1, p. 100012, 2021.

[17] E. Barka, S. Dahmane, C. A. Kerrache, M. Khayat, and F. Sallabi, "STHM: A secured and trusted healthcare monitoring architecture using SDN and Blockchain," *Electronics*, vol. 10, no. 15, p. 1787, 2021.

[18] B. Benattia, N. E. H. Maicha, C. Kerrache, G. Rathee, and C. Calafate, *Implementing a Blockchain Based System for Healthcare Applications Using Digital Twin*. 2023, pp. 1-6.

[19] S. Memon, A. Lakhan, and Q. U. A. Mastoi, "AQ-ResCon: Adaptive quantum-resistant lattice-based key agreement protocol for secure distributed container orchestration in edge cloud environments," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 3, pp. 377-389, 2025.