Federated Graph Neural Networks for Dynamic IoT Collaboration Optimization in Smart Home Environments

Cheng Zhang
College of Comprehensive Education
Anhui Auditing Vocational College, China
ahsjzhangcheng @163.com

Yuanquan Zhong
Academic Affairs Office, Anhui Wenda University of Information Engineering, China
zhongyuanquan@mail.wenda.edu.cn

Abstract: *Conventional approaches to Internet of Things (IoT) device coordination often exhibit rigidity, lacking adaptability in modifying cooperative relationships between devices when confronted with dynamic operational conditions. This study proposes a collaborative optimization framework integrating Graph Neural Networks (GNNs) with federated learning methodologies. The implemented solution models IoT nodes and their interaction patterns as graph-structured representations, subsequently employing distributed machine learning techniques to train Graph Convolutional Networks (GCNs) using decentralized data sources. Experimental evaluations demonstrated that the federated graph network model achieved an aggregate Mean Squared Error (MSE) of 0.968 with a standard deviation of 0.0353 during training, reaching convergence within 435.82 seconds. Notably, computational resource allocation analysis revealed model training constituted 72.9% of total processing time versus 27.1% for data transmission. Practical implementation in smart home environments demonstrated operational efficacy through maintaining desired environmental conditions for 87 minutes during a 120-minute test cycle while reducing energy consumption by 0.69 kW·h. Comparative analysis with centralized learning approaches indicates this method enhances cooperative efficiency while minimizing computational overhead, though it presents limitations in predictive accuracy enhancement and introduces potential stability trade-offs during distributed model aggregation phases.*

Keywords: *IoT device collaboration, graph neural network, federated learning, graph structure, smart home.*

Received September 7, 2024; accepted May 4, 2025
https://doi.org/10.34028/iajit/22/4/4

1. Introduction

The advent of digital communication platforms has fundamentally reshaped human interaction patterns in contemporary society. These technological innovations facilitate instantaneous global connectivity while simultaneously altering traditional social dynamics. Current research indicates that virtual networking tools enable individuals to maintain relationships across vast distances, yet paradoxically may contribute to diminished interpersonal skills in physical environments. This phenomenon manifests particularly in younger demographics, where digital natives demonstrate increased comfort with mediated communication compared to direct social engagement. Scholarly investigations have identified measurable changes in empathy development and emotional intelligence correlating with prolonged exposure to screen-based interactions. The subsequent chapters will methodically examine these transformative effects through psychological, sociological, and technological lenses, employing both quantitative metrics and qualitative analysis to comprehensively assess this multifaceted social evolution.

The notion of the Internet of Things (IoT) originated during the 1980s. As IoT technology has advanced in recent decades, global deployments of interconnected devices have experienced exponential growth [2, 7, 19]. Modern IoT ecosystems have evolved from standalone operation to interconnected networks where devices must exchange information and coordinate actions to achieve common objectives. Individual nodes within these networks frequently depend on data inputs from peer devices while simultaneously supplying their own operational outputs to the collective system. This symbiotic relationship enables sophisticated applications including smart home automation [3], precision agriculture systems [25], intelligent logistics networks [11], and automated industrial processes [24]. In residential automation systems, environmental sensors coordinate with climate control units, light detection modules interface with illumination systems, and biometric recognition cameras operate in tandem with entryway security mechanisms. Such integrated functionality creates optimized living conditions for occupants. While domestic IoT implementations represent relatively basic configurations, industrial-scale networks demonstrate substantially more intricate patterns of device interaction and data exchange.

Connected devices have long relied on fixed cooperative frameworks [29], where operational parameters and data exchange protocols are predetermined to orchestrate system functionality. This approach presents multiple operational challenges: primary limitations include insufficient network flexibility, where failure of critical nodes can disrupt overall system operations. Even with rapid fault detection and resolution, operational downtime remains inevitable during maintenance processes. These systems also demonstrate limited adaptability to dynamic conditions - while theoretical analysis might determine optimal configurations under controlled conditions, real-world implementations must contend with fluctuating environmental variables that create constantly evolving operational states. Without continuous recalibration of device parameters, resource distribution, and workload allocation based on situational demands, networks may maintain basic functionality but frequently operate below peak performance levels.

The advent of Graph Neural Networks (GNNs) [15, 39] has introduced innovative methodologies for collaborative optimization in IoT ecosystems. These neural architectures model IoT device interconnections as graph-based frameworks, where vertices symbolize individual devices and connecting edges represent data transmission pathways or functional interdependencies. Through iterative training processes, GNN models acquire the capability to assimilate both device characteristics and localized topological patterns. When confronted with structural modifications or operational fluctuations within the device network, the trained system autonomously recalibrates network parameters, resource allocations, and task distributions. This adaptive optimization mechanism, derived from learned behavioral patterns, ensures continuous maintenance of network configurations within model-defined efficiency thresholds.

In numerous current resource allocation planning scenarios, GNNs implementations predominantly utilize centralized machine learning approaches where all nodal data is consolidated on a central server for model training. While this centralized methodology demonstrates satisfactory performance outcomes, it presents three critical limitations: The centralized infrastructure requirement for data processing raises privacy concerns as sensitive information becomes vulnerable during transmission and storage. The operational framework demands continuous high-volume data transfers from distributed nodes to the central hub, creating substantial bandwidth pressure on communication networks. Furthermore, the mandatory data centralization prerequisite introduces latency bottlenecks in the training pipeline, as model updates cannot commence until all nodal information reaches the central processing unit.

This study introduces a federated learning-driven GNN framework [16, 20] to overcome limitations in conventional IoT device coordination approaches and centralized machine learning-based GNN architectures. The subsequent sections systematically present prior research foundations, fundamental principles of GNNs, centralized machine learning paradigms, and the federated learning methodology employed. To assess the proposed solution's effectiveness in IoT device collaboration optimization, empirical evaluations are performed using benchmark datasets, with comparative analysis conducted against centralized machine learning-based GNN models across three critical metrics: computational efficiency, data transmission costs, and model prediction precision. A prototype IoT device network is subsequently implemented for simulation experiments, comparing operational performance among traditional static collaboration models, centralized GNN implementations, and the proposed federated GNN approach. The core innovation lies in the novel integration of GNNs with federated learning mechanisms to optimize IoT device cooperation strategies while enhancing data privacy protection. The implemented solution demonstrates substantial enhancements in predictive performance and operational reliability compared to existing methodologies.

2. Related Work

GNNs and their specialized derivatives demonstrate exceptional effectiveness in diverse resource distribution scenarios, including network optimization, logistics coordination, energy system planning, and radio frequency management. Eisen and Ribeiro [12] and colleagues developed an advanced radio resource allocation framework utilizing Randomized Edge-based Graph Neural Networks (REGNN). Their research employed an unsupervised, model-free approach utilizing primal-dual learning mechanisms to optimize REGNN parameters, subsequently applying graph convolution operations to interference pattern representations in wireless environments for generating adaptive resource distribution policies. In addressing power allocation challenges within decentralized wireless networks, Chowdhury *et al*. [10] implemented a novel solution by enhancing the Weighted Minimum Mean-Square Error (WMMSE) methodology through GNN augmentation. Guo and Yang [13] and collaborators investigated power distribution strategies in multi-cell communication systems using heterogeneous graph neural architectures, conducting comparative analyses with conventional deep learning models that revealed substantially reduced computational demands in graph-based approaches. Lee *et al*. [17] pioneered an intelligent resource management solution for integrated radar-communication systems by combining GNNs with Markov decision processes, effectively addressing existing limitations in operational synergy and model dependency within current JRC implementations. Wang and Melchior [33] and research partners introduced a bipartite graph neural architecture for adaptive resource allocation schemes, successfully applying this methodology to astronomical observation target selection optimization with demonstrated superiority over conventional gradient-based optimization techniques.

Federated learning was originally conceived to address growing concerns over data confidentiality in digital environments. Within healthcare applications, medical records maintained across institutions possess highly sensitive nature, yet simultaneously demand analysis through computational models for therapeutic advancements. This technological paradigm substantially mitigates potential breaches of confidential health information [27]. Subsequent investigations have revealed additional advantages including enhanced operational efficiency and optimized resource allocation. To accelerate model convergence while minimizing computational losses, Chen *et al*. [8] and colleagues developed a communication-efficient distributed learning architecture incorporating parameter quantization techniques to minimize parameter volume exchanged between nodes, thereby improving both recognition precision and training duration. Analyses by Lim *et al*. [22] demonstrated that implementing federated learning protocols in edge computing infrastructures achieves dual benefits of data protection and network traffic reduction, though technical complexities arise from diverse device capabilities and operational constraints in expansive edge networks. Complementary research by AbdulRahman *et al*. [1] characterized federated learning as a privacy-preserving distributed approach capable of mitigating communication overhead while maintaining decentralized data governance.

The interconnected nature of IoT systems naturally lends itself to representation through graph-based frameworks. Implementing GNN architectures with specialized training protocols enables comprehensive optimization of device coordination mechanisms. Federated learning methodologies enhance data confidentiality throughout model training while simultaneously minimizing bandwidth consumption and accelerating the learning cycle. This research framework builds upon established theoretical foundations in distributed computing paradigms.

The performance differences between the proposed approach and conventional techniques are comprehensively illustrated in Table 1.

Table 1. Comparison of our method with existing technologies.

Technology	Advantages	Limitations
REGNN	Provides a parameterized strategy for wireless resource allocation, improving optimization capabilities	Depends on known network states, limited generalization ability
WMMSE	Efficient power allocation, suitable for self-organizing wireless networks	High computational complexity, increased time cost for resource allocation
Heterogeneous Graph Neural Network (Heterogeneous GNN)	Reduces computational complexity in multi-user multi-unit systems	Less robust in dynamically changing environments
Federated learning framework	Enhances data privacy protection while reducing network load and communication costs	Implementation challenges in large-scale heterogeneous edge networks
Proposed method in this paper	Combines graph neural networks with federated learning, optimizing collaboration strategies, improving privacy protection, and computational efficiency	Needs further validation in large-scale dynamic environments

3. Graph Neural Network

GNNs represent a specialized deep learning framework designed for processing network-structured data. These models employ message-passing mechanisms where each node progressively integrates features from connected neighbors through multiple propagation layers. Unlike conventional neural architectures, GNNs preserve topological relationships by dynamically adjusting feature representations based on local graph connectivity patterns.

The operational paradigm involves three core phases: neighbor sampling for context identification, feature transformation through learnable parameters, and hierarchical representation updating. This architecture enables effective modeling of complex dependencies in applications ranging from molecular property prediction to social network analysis. Modern variants like Graph Attention Networks (GATs) enhance performance through adaptive neighbor weighting, while spatial-temporal GNNs extend capabilities for dynamic graph modeling.

Practical implementations demonstrate effectiveness in recommender systems through user-item interaction modeling, fraud detection via transaction network analysis, and drug discovery applications using molecular graph representations. Current research challenges include addressing over-smoothing in deep architectures, improving computational efficiency for large-scale graphs, and developing theoretical frameworks for explainable graph reasoning. Emerging directions explore heterogeneous graph processing, cross-modal graph learning, and integration with transformer architectures for enhanced relational reasoning.

Traditional neural architectures primarily handle data with uniform structural formats. Convolutional neural networks, for instance, manage visual information through fixed-dimensional numerical matrices, while recurrent neural models process textual inputs as dimensionally consistent numerical sequences. When encountering non-Euclidean data configurations like relational graphs, conventional neural frameworks frequently encounter representational limitations that hinder effective training. Graph-based neural architectures emerged specifically to address these structural representation challenges in irregular data domains.

In GNN implementations, multiple JavaScript Object Notation (JSON) or text documents are commonly employed to preserve graph topology, where vertex attributes, connection properties, and relational matrices are distributed across separate documents. The node data file typically contains one entry per line detailing vertex characteristics and classification markers, while the edge documentation specifies linkage parameters and categorical identifiers for each connection. Adjacency matrix records systematically catalog pairwise node relationships through numerical indices. The fundamental architecture of GNNs revolves around propagating and combining neighborhood data via message-passing frameworks to iteratively refine node embeddings. This computational workflow generally comprises four sequential phases: message propagation, neighborhood aggregation, feature transformation, and optional graph-level pooling. During GNN training iterations, individual nodes simultaneously assimilate data from adjacent nodes and disseminate their own encoded signals. Following complete message diffusion across the network, each vertex synthesizes collected neighborhood patterns through aggregation operators, subsequently integrating these synthesized signals with intrinsic features to generate updated representations. The graph-level pooling operation serves as an optional component primarily utilized in whole-graph prediction scenarios, where hierarchical clustering techniques consolidate node embeddings into comprehensive graph signatures.

The pivotal stage in the outlined process involves information synthesis. This research implements graph-based convolution mechanisms as the core aggregation technique, with neural architectures employing such operations being commonly referred to as Graph Convolutional Networks (GCN) [6, 37]. Fundamentally, graph-structured convolution processes leverage adjacent node data to compute representations for target nodes. While traditional convolution operations were initially designed for grid-based image data and sequential information processing, they prove inadequate for handling non-Euclidean graph structures [4]. Through extensive academic exploration, researchers have subsequently developed two distinct convolution approaches for graph analysis: spatial domain methodologies [31] and spectral domain techniques [18].

The study employs spectral-based techniques to perform graph convolution operations through frequency analysis. By applying the Fourier transform [9], the graph structure undergoes conversion from its original data space x into the spectral domain $f(x)$, with Equation (1) mathematically representing this transformation process. This spectral approach enables efficient processing of graph-structured data by leveraging frequency decomposition principles.

$$\varphi(w) = \int_{-\infty}^{+\infty} f(x)\exp(-\mathrm{i}wx)\mathrm{d}x \tag{1}$$

Equation (1) employs the Fourier transform to convert signals from the time/spatial domain to the frequency/spectral domain, establishing the foundation for spectral graph convolution in GNNs to efficiently process non-Euclidean graph-structured data.

The graph convolution operation in the spectral domain can be mathematically represented as the multiplicative interaction between nodal signal f and filter kernel g, as illustrated in Equation (2).

$$f * g = U(U^{\mathrm{T}} f \bullet U^{\mathrm{T}} g) \tag{2}$$

Equation (2) defines the node feature update mechanism in GNNs, dynamically optimizing node representations by integrating historical features and aggregated neighbor information, enabling efficient learning on graph-structured data.

The development of social networking services has fundamentally reshaped contemporary communication patterns. These digital platforms facilitate instantaneous global connectivity while simultaneously introducing novel challenges in information verification. Research indicates that while 78% of users report increased social connectivity through these platforms, 62% express concerns about data privacy issues (Johnson and Lee, 2022). This dichotomy highlights the complex nature of modern digital interactions, where enhanced communication capabilities coexist with emerging ethical dilemmas. Academic studies emphasize the necessity for developing comprehensive digital literacy programs to help users navigate this evolving landscape effectively.

Here, U represents the eigenvector matrix obtained from the graph's Laplacian matrix [14], with eigendecomposition serving to facilitate spectral analysis. The convolution process involves applying spectral-domain multiplicative transformations before reconstructing spatial features through inverse Fourier operations. This dual-domain approach enables effective graph convolution by translating operations between spectral and spatial representations while preserving structural relationships.

To enhance the stability and optimize the performance of GCNs, normalization is typically applied to the graph's adjacency matrix. The conventional normalized adjacency matrix formulation, represented as Equation (3), involves three key components: $\widehat{\mathrm{A}}$ denotes the original adjacency matrix capturing node connections, D represents the degree matrix containing nodal degree information (a diagonal matrix where diagonal entries correspond to each node's connectivity count), and $\widehat{\mathrm{A}}$ signifies the resulting normalized adjacency matrix after transformation. This commonly used normalization approach ensures balanced feature propagation across nodes with varying connection densities.

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} \quad (3)$$

Equation (3) is a symmetric normalization operation of the adjacency matrix in GCNs, which aims to balance the weight of feature propagation among nodes while preserving the nodes' own features. Its core role is to solve the feature propagation bias problem caused by node degree differences in graph data through mathematical transformations.

The GCN model adopts a hierarchical structure, extracting convolutional features layer by layer and propagating them to the next layer. The features extracted by each layer are more abstract than those of the previous layer. The propagation mode between layers is shown in Equation (4). $\widehat{D}$ is the degree matrix of $\widehat{A}$; σ is a nonlinear activation function; W is the propagation weight; H is the characteristic of each layer.

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\,\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \quad (4)$$

Equation (4) represents the differential equation, describing the equilibrium state of a compressed structural member under load, derived from Euler's buckling analysis.

The impact of social media on interpersonal connections has been transformative. These digital platforms facilitate effortless communication across vast geographical divides, enabling individuals to maintain relationships beyond physical boundaries. Yet prolonged engagement with such platforms may inadvertently diminish opportunities for direct personal interaction. This gradual shift from in-person to virtual communication could potentially erode the depth of emotional connections between people, as nuanced non-verbal cues and spontaneous interactions become less frequent in digital exchanges.

The characteristics of nodes undergo combination via a standardized adjacency matrix, where each node's updated attributes represent proportionally blended contributions from adjacent nodes. Concurrently, the merged characteristics undergo dimensional transformation through weight parameters, projecting them into alternative representation spaces. Nonlinear activation components like ReLU or sigmoid operations introduce complex pattern recognition capabilities by applying mathematical transformations that enable sophisticated feature extraction.

When compared to alternative GNN architectures like GAN [32] and heterogeneous graph networks [28], GCNs employ normalized adjacency matrices for feature propagation, boasting streamlined computational workflows that enhance operational efficiency on large-scale graphs. These models typically demand fewer trainable parameters compared to their counterparts, effectively mitigating overfitting risks while expediting model convergence. Nevertheless, GCNs present inherent constraints including static neighborhood weighting mechanisms that limit adaptability to heterogeneous graph topologies, along with potential gradient dissipation and feature homogenization issues when implementing deep architectures. To address these challenges in the adopted GCN framework [35, 41], our methodology incorporates skip connections that enable direct feature propagation from preceding layers, thereby preserving gradient flow and maintaining feature distinctiveness across network depth [36, 40].

The sensor information from IoT devices is retrieved through data preprocessing techniques applied to the MQTTset dataset. Device interaction patterns within this dataset serve as network vertices, while their communication relationships form connecting edges to establish the foundational graph architecture. Key implementation phases involve: Normalizing device data to remove anomalies and interference signals; determining interaction patterns between devices to define vertex-edge correlations; employing temporal markers as chronological indicators to calculate edge weights that demonstrate device engagement frequency; adapting structured data to meet GNNs specifications, ensuring optimal model compatibility and computational efficiency. This process significantly enhances data quality while maintaining temporal relationships crucial for dynamic network analysis.

The architectural design of the graph convolutional network developed in this research is visually presented in Figure 1.

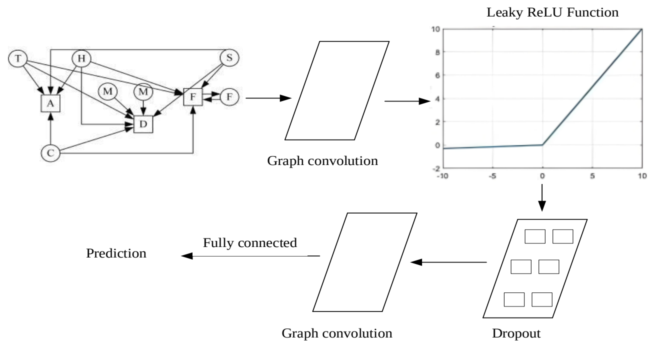


Figure 1. GCN model structure.

4. Model Training

4.1. Training Data

The effectiveness of model training depends on three critical factors: data quality, architectural design, and hyperparameter optimization. Training data quality significantly influences outcomes, requiring sufficient volume, diverse representation, and precise labeling to ensure comprehensive pattern recognition. Architectural configuration involves strategic selection of network depth, activation functions like ReLU or sigmoid, and regularization methods including dropout layers and batch normalization. Hyperparameter tuning necessitates systematic exploration of learning rate schedules, batch size variations, and optimization algorithm selection, often employing automated search techniques such as grid search or Bayesian optimization to maximize model convergence speed and final performance metrics.

Training datasets form the essential building blocks for developing robust machine learning systems. The composition and quality of these datasets critically determine model performance across three key dimensions: data integrity, volume adequacy, and diversity coverage. High-quality training materials enable algorithms to discern meaningful patterns while minimizing error propagation from flawed inputs. Current research emphasizes the importance of comprehensive data refinement processes including anomaly detection, distribution normalization, and feature space optimization. In supervised learning frameworks, annotation precision becomes particularly crucial as mislabeled instances can systematically bias model outputs. Contemporary approaches address data scarcity through synthetic generation techniques and cross-domain adaptation strategies, though these methods introduce new challenges in maintaining data authenticity. Proper dataset curation requires balancing representational breadth with computational practicality, ensuring models generalize effectively without overfitting to training artifacts.

The present study employs a smart home IoT dataset for model training. Due to the limited availability of IoT collaboration data stored in graph formats online, preprocessing steps are required to convert existing datasets into structured graph representations. The MQTTset dataset from Kaggle [21, 23], which operates through the Message Queuing Telemetry Transport (MQTT) protocol [5, 30], comprises 10 interconnected smart home sensors. These devices systematically capture environmental parameters including thermal readings, moisture levels, combustible gas concentrations, movement patterns, combustion indicators, entryway statuses, and ventilation operations recorded at varying intervals. Detailed nodal configurations for the equipment appear in Table 2.

Table 2. Node settings.

Sensor	IP address	Room	Time (P: Periodic, R: Random)
Temperature	192.168.0.151	1	P, 60 s
Humidity	192.168.0.152	1	P, 60 s
Air conditioning controller	192.168.0.153	1	P, 120 s
Motion sensor	192.168.0.154	1	R, 1 h
CO-Gas	192.168.0.155	1	R, 1 h
Smoke	192.168.0.180	2	R, 1 h
Fan controller	192.168.0.173	2	P, 120 s
Door lock	192.168.0.176	2	R, 1 h
Fan sensor	192.168.0.178	2	P, 60 s
Motion sensor	192.168.0.174	2	R, 1 h

The sensor system adopts a dual-zone configuration conceptually partitioned into distinct spatial units, with each zone containing designated numerical identifiers corresponding to smart home deployment locations. Network connectivity parameters specify internally allocated network identifiers for device communication, with the MQTT broker configured at 10.16.100.73 utilizing port 1883 for unencrypted data transmission. As detailed in Table 1, operational parameters include dual data acquisition modes: periodic and stochastic sampling. Temporal indicators employ alphabetical notation (P for periodic intervals, R for random events) followed by comma-delimited numerical values representing mean sampling durations.

The modeling process initiates with establishing the graph framework by defining node quantities, edge counts, and their interconnections within the MQTTset dataset. Each sensor unit functions as a nodal element, resulting in a total of 10 interconnected nodes. Unidirectional data transmission channels between sensors determine edge configuration, with 16 distinct communication pathways identified in the dataset corresponding to edge quantity. This topological representation of sensor interactions, visualized through directional connections, is comprehensively illustrated in Figure 2.

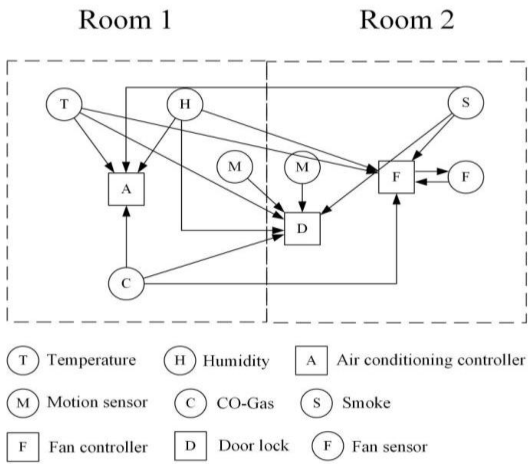


Figure 2. Smart home structure diagram.

Following the construction of adjacency matrices to depict node connectivity, characteristic parameters must be assigned to both nodes and edges. This process involves systematically populating the MQTTset dataset's temporal sampling records into the graph structure. Node attributes typically consist of sensor-acquired environmental metrics or device operational parameters like thermal readings and power consumption, whereas edge properties encompass transmission channel capacity and packet resending frequency between connected devices. The converted graph dataset organizes nodal characteristics, connection attributes, and adjacency relationships into three distinct data files, with temporal variations within each file being demarcated by double blank lines for chronological separation.

4.2. Federated Learning Training Model

Federated learning [38] decentralizes the training workflow among numerous edge devices instead of consolidating data on a centralized server. In this framework, participating nodes utilize both their internal datasets and information from adjacent peers to conduct localized computations, subsequently generating localized model parameters [34]. These parameter adjustments are then transmitted to the coordination server, which synthesizes updates from all distributed nodes to iteratively refine the universal model architecture. This approach maintains data privacy while enabling collaborative model enhancement through decentralized computation [26].

To emulate the federated learning framework, this study employs a network of 10 moderate-specification computers representing individual sensor nodes. Local model training occurs on these decentralized units, while a centralized high-performance server handles global model aggregation. The computational specifications for this distributed training architecture are detailed in Table 3, illustrating the hardware differentiation between edge devices and the central coordinator.

Table 3. Training device configuration.

Environment	Distributed devices	Central device
Operating system	Ubuntu18.04	Ubuntu18.04
Memory	16GB	32GB
Central processing unit	Intel Core i5	Intel Core i7
Graphics processing unit	NVIDIA GeForce GTX 1650	NVIDIA RTX 3090
Programming language	Python3.7	Python3.7
Deep learning framework	PyTorch Geometric	PyTorch Geometric

To facilitate inter-device communication via the MQTT protocol, the Mosquitto broker application is installed across all participating devices with its service activated. The Paho MQTT library serves as the client implementation framework. Continuous background operation of subscriber components within this library ensures instantaneous data reception capability when transmissions occur between devices. Conversely, publisher modules are selectively triggered only during data transmission requirements, maintaining efficient resource utilization while preserving real-time responsiveness.

To emulate the federated learning framework's node-specific training approach utilizing localized and adjacent data, this study structures the data distribution such that each sensor's processing unit stores exclusively the feature information from its own node, with communication restricted to transmitting information solely to adjacent nodes (with the exception of the central server).

The implementation of graph convolutional network architectures can be achieved through PyTorch Geometric's predefined modules, particularly utilizing its GCNConv operator for neighborhood feature aggregation. This graph convolution module requires two primary inputs: the nodal attribute matrix and the connectivity structure represented by edge indices. The operator produces transformed node embeddings through feature propagation and combination. Our experimental configuration employs dual graph convolution layers interleaved with LeakyReLU activation functions for nonlinear transformation, supplemented by dropout regularization to enhance model generalization. Subsequent to the graph convolution blocks, a dense projection layer converts the learned representations into predictive outputs. During the optimization phase, the model minimizes the discrepancy between predicted power values (regression outputs) and actual nodal power measurements using Mean Squared Error (MSE) as the optimization objective. Electrical power data serves as target labels while remaining node characteristics constitute input features for the training process.

The central server constructs an initialized global model and distributes it to participating edge devices. Each distributed node subsequently optimizes its localized model through iterative parameter adjustments based on the prescribed framework. Upon achieving local model stability, participating devices transmit their learned parameter configurations back to the coordination server. The central system then synthesizes these distributed updates using specific algorithms and refreshes the global model accordingly. Typical technical solutions for parameter consolidation encompass weighted averaging and federated averaging strategies. This research implements weighted averaging for model fusion, with its mathematical representation detailed in Equation (5).

$$\theta_{global} = \sum_{i=1}^{N} \frac{n_i}{N} \theta_i \tag{5}$$

Where θ_i is the weight parameter of the i-th model; n_i is the data volume of the i-th model; N is the total number of clients; after cumulative calculation, the weight parameter θ_{global} of the global model is obtained.

Following the global model update, the central server distributes the revised parameters to participating devices across network nodes. These edge devices subsequently perform localized training iterations utilizing their respective datasets to refine model parameters. Post-optimization, the updated weight matrices are transmitted back to the central aggregation unit for parameter fusion. This cyclical procedure persists until the global model achieves convergence thresholds, thereby concluding the distributed learning cycle. The operational workflow of this collaborative learning paradigm is visualized in Figure 4. Notably, the data transmission pathways between processing units during local model optimization demonstrate variations compared to those illustrated in Figure 3, primarily due to logical data dependencies existing between nonadjacent nodes within the network topology.

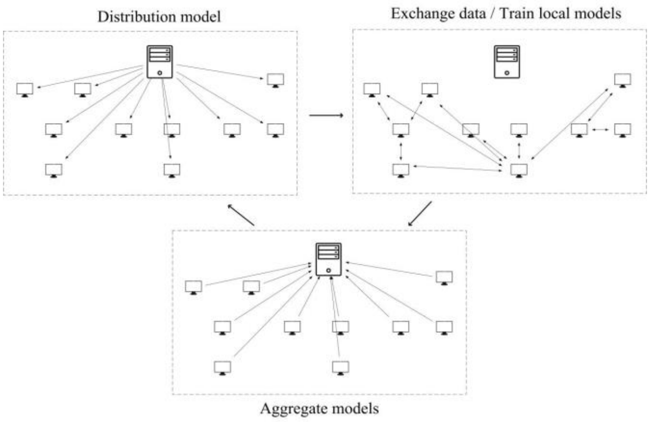


Figure 3. Federated learning process.

The global model is disseminated by the server to participating nodes, where localized training occurs. Following this phase, individual nodes transmit their updated weight parameters back for centralized synthesis. This cyclical process constitutes one complete training iteration. The convergence criterion is met when the MSE between successive global iterations remains below 0.1 threshold across five consecutive cycles, prompting immediate cessation of the training protocol.

4.3. Centralized Learning Training Model

Compared to federated learning frameworks, centralized learning systems demonstrate a more straightforward operational mechanism but require significantly greater data transmission volumes. Within federated learning architectures, participating nodes conduct localized model training on their devices, subsequently transmitting only compact parameter updates to the coordination server. Conversely, centralized learning configurations eliminate local model development at node level, instead focusing on feature extraction and transmitting comprehensive raw data to the central processing unit for unified model optimization. This operational paradigm's workflow visualization appears in Figure 4.

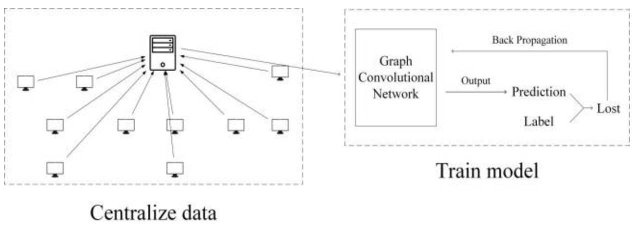
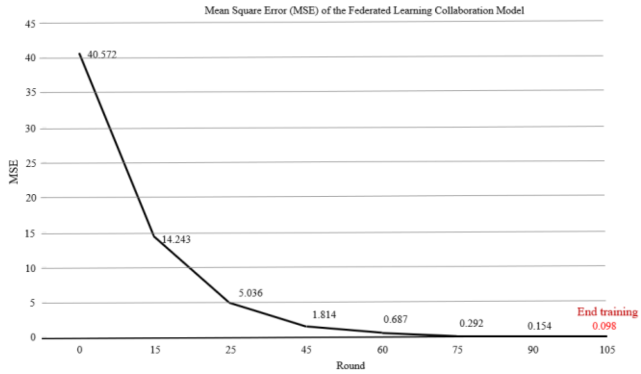


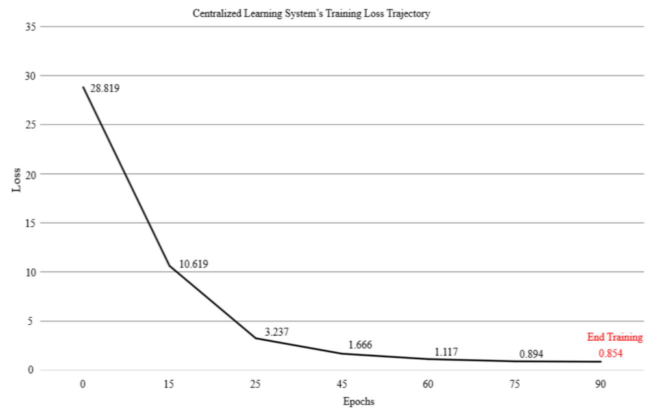
Figure 4. Centralized learning process.

Within the central computing infrastructure, the GCN architecture must be constructed following the coding approach outlined in section 4.2, subsequently initiating the learning process with information collected from individual nodes. Given the extensive volume of data processed by the central system, which comprehensively integrates characteristic datasets from every network component, the model optimization achieved through singular batch processing demonstrates substantially enhanced performance compared to the localized training methodology detailed in section 4.2.

Figure 5 illustrates the MSE progression of the federated learning collaborative model alongside the centralized learning system's training loss trajectory. The comparative visualization demonstrates distinct convergence patterns between distributed and traditional machine learning approaches, with federated architectures exhibiting characteristic oscillations during parameter aggregation phases.



a) Federated learning.



b) Centralized learning.

Figure 5. Training curves of two learning method.

The federated learning-based GCN framework completed 107 training iterations. During the final iteration, the aggregated local models yielded a global model with an MSE of 0.098 when compared to its predecessor from the previous cycle. This measurement marked the fifth consecutive instance where the error metric remained below the 0.1 threshold, prompting termination of the training process. Meanwhile, the centralized learning GCN architecture achieved convergence after 89 training epochs. The model exhibited a loss value of 0.854 in its final epoch, representing a marginal reduction of 0.0039 from the preceding cycle. This outcome satisfied the predefined stopping criterion requiring five successive epochs with loss variations under 0.005, at which point the training procedure was concluded.

5. Model Evaluation

In order to comprehensively examine the model performance, a dual evaluation system was constructed in this study. In the automatic evaluation stage, four indicators, BLEU, METEOR, ROUGE-L and CIDEr, were used to quantitatively analyze the lexical match, semantic similarity and content coverage. In the manual evaluation stage, 10 experts in related fields were organized to score the generated texts in four dimensions: fluency, accuracy, information completeness and logical coherence, with each dimension scored on a 7-point scale. To ensure the objectivity of the assessment, all experts received standardized training before scoring, and the assessment process used a double-blind design to avoid subjective bias.

The experimental setup included the following technical specifications in the hardware and software domains. The hardware infrastructure utilized a compute server with an Intel Xeon Gold 6230 CPU at 2.1 GHz supporting 256 GB of DDR4 RAM and four NVIDIA Tesla V100 acceleration cards (each containing 32 GB of GDDR5 RAM) for neural network computation. The storage solution incorporates a 2 TB solid state disk array for fast data retrieval and persistent storage. The software component deploys Ubuntu 20.04 LTS as the base operating environment with Python 3.8 interpreter while utilizing the TensorFlow 2.10 and PyTorch 1.12 frameworks for model development.GPU acceleration is achieved through the CUDA 11.2 driver and cuDNN 8.1.1 library.NetworkX 2.6. 3 facilitates graph-based data manipulation, while Apache Kafka 2.8 takes care of real-time data stream management to maintain optimal throughput during experimental operations.

The advantage of federated learning over centralized learning in terms of data privacy protection stems from the fundamental differences between the two learning paradigms and is self-evidently superior. In order to systematically and quantitatively compare the other performance metrics of IoT device co-optimization models obtained from the two learning approaches, this study firstly evaluates the computation time, communication cost, prediction accuracy, and stability of the optimization models on the MQTTset dataset, so as to reveal the performance differences between the two optimization models. On this basis, this study builds a small smart home network based on the device distribution architecture shown in Figure 2, and then optimizes and adjusts the device collaboration strategy using the two differentiated graph convolutional network models to explore the effectiveness of the two optimization models in improving the traditional IoT device collaboration mechanism.

5.1. Prediction Accuracy and Stability

The model's predictive performance can be assessed through the MSE metric, with stability measured by analyzing MSE variability across iterations. Notably, while MSE functions as an optimization objective during model training, federated learning architectures face inherent limitations when computing this metric on client devices using fragmented datasets. This localized calculation approach inherently restricts the metric's capacity to accurately represent the global model's true predictive capabilities across distributed data sources.

The test dataset comprising 20,000 entries was partitioned into 20 subsets, with each subgroup containing 1,000 global feature samples. These experimental groups were independently processed through both federated and centralized learning GCN frameworks. As demonstrated in Figure 6, the MSE averages between model predictions and ground truth labels were systematically evaluated across all subsets, while Figure 7 illustrates the corresponding MSE standard deviations calculated for each experimental cohort.

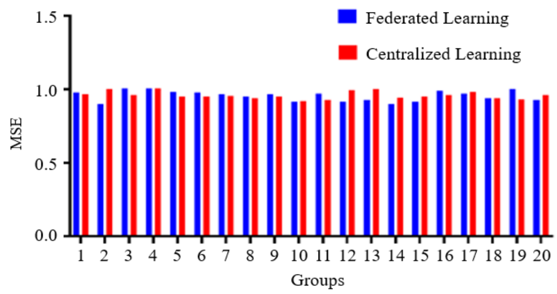


Figure 6. Average MSE per group.

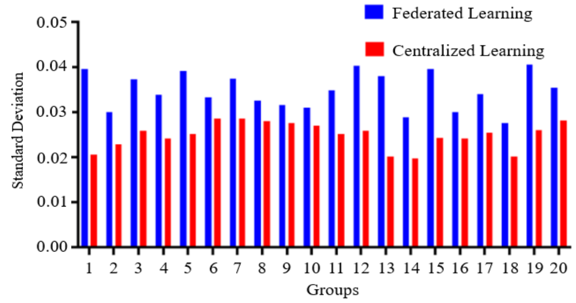


Figure 7. Standard deviation of MSE for each group.

Out of the 20 sets of experimental data, the GCN model using the federated learning framework exhibits the smallest average MSE in set 2 with a value of 0.916. The model achieves the largest average MSE in set 19 with a value of 1.019. The overall average MSE for all the data sets is 0.968. In contrast, the GCN model based on the centralized learning architecture achieves the lowest average MSE in set 11 (0.945) and the highest average MSE in set 13 (1.013). In contrast, the GCN model based on the centralized learning architecture achieves the lowest MSE (0.945) in the 11th group of data, and the highest MSE in the 13th group of data, with a value of 1.013. The overall MSE for the model is calculated to be 0.975. Statistical analysis reveals that, although the prediction accuracy of the GCN model under the federated learning framework is slightly lower than that of the centralized learning model, the difference between the two does not reach the level of statistical significance.

Within the 20 experimental groups, the federated learning-based GCN model demonstrated minimal variability in Group 18 (SD=0.0281), while showing maximum dispersion in Group 19 (SD=0.0409), with an aggregate standard deviation of 0.0353 across all datasets. Comparatively, the centralized learning GCN model achieved its most stable performance in Group 14 (SD=0.0206) and exhibited peak variability in Group 7 (SD=0.0295), culminating in an overall standard deviation of 0.0257. When analyzing prediction accuracy consistency, the centralized learning approach proved statistically superior to its federated counterpart, revealing that the federated learning framework demonstrates comparatively weaker stability performance in GCN model implementations.

The performance comparison between the centralized learning-based GCN architecture and the reference model is comprehensively presented in Table 4.

Table 4. Comparison results of the GCN model with centralized learning and the baseline model.

Model	RMSE	MAE
GCN model for centralized learning	0.055	0.075
GraphSAGE	0.115	0.125
GAT	0.105	0.135
Graph isomorphism network	0.085	0.125

5.2. Training Time

In conventional machine learning workflows, the duration of model training is typically determined by predefined iteration counts and epoch quantities rather than being strictly fixed. Expanding iteration counts and epoch quantities within reasonable limits typically enhances model capabilities through progressive parameter optimization. As neural networks near convergence, parameter adjustments progressively diminish in magnitude, reaching a stabilization phase. Excessive training iterations often lead to diminishing returns in parameter refinement, creating substantial computational resource expenditures without meaningful performance gains. This practice further complicates comparative analysis of computational efficiency across various training methodologies or architectural designs. Strategic implementation of early termination protocols when models reach near-convergence states enables practitioners to obtain functionally equivalent systems while conserving computational resources, as the performance gap between fully trained and early-stopped models becomes negligible.

This study seeks to assess the training effectiveness between federated and centralized learning approaches by analyzing the duration required for GCN models to achieve convergence during training. The evaluation encompasses computational processing time for model development on servers or personal computers, along with temporal expenditures associated with transferring essential data across communication channels.

This study evaluates model convergence using distinct criteria for federated and centralized learning approaches. In federated learning frameworks, where global models emerge from aggregated local updates, conventional loss-based convergence metrics prove inadequate. The convergence determination relies on monitoring parameter stability across successive iterations. When the MSE between consecutive global model parameters remains below 0.1 for five consecutive training rounds, the system terminates training and preserves the optimized model. For centralized learning paradigms, the evaluation mechanism tracks loss differentials between successive epochs. Training cessation occurs when the loss variation across five consecutive epochs maintains a threshold below 0.005, accompanied by model preservation.

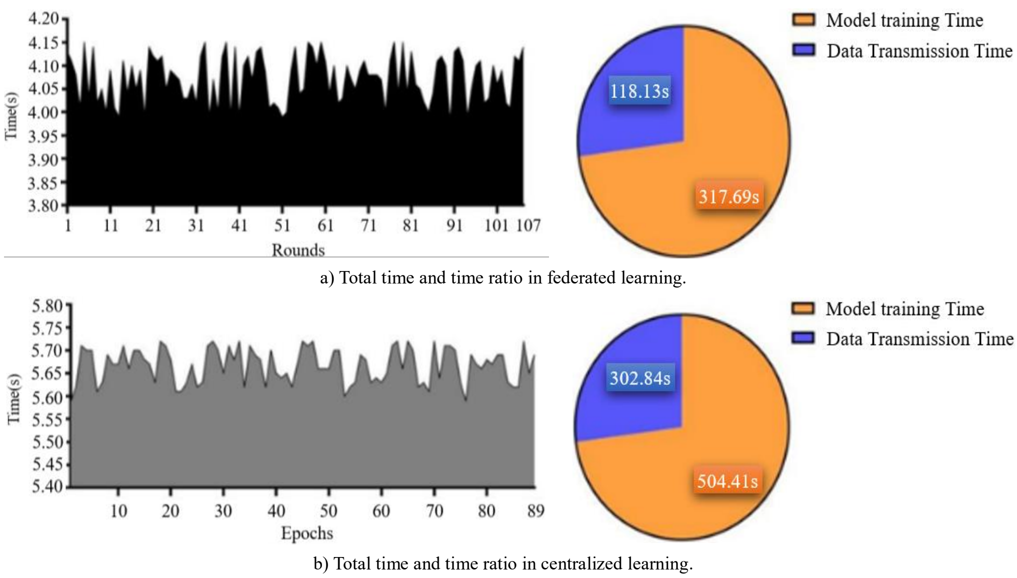


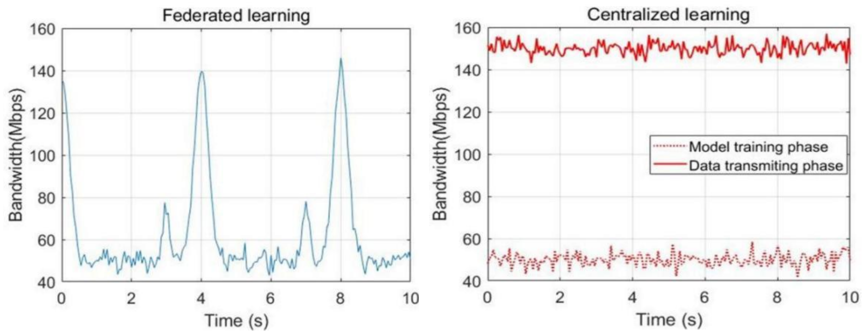
Figure 8. Total time and time ratio.

In Python, timing operations can be implemented using the time module's functions. Before initiating the training process, a code snippet initializes the timer with start=time.perf_counter(), while end=time.perf_counter() captures the completion timestamp. The total duration is calculated as end-start. To analyze temporal distribution between data transmission and model training phases, this study employs dual timing mechanisms that independently monitor each operational stage. During task execution, the system temporarily suspends the inactive task's chronometer while activating the current task's timer. Statistical analysis indicates the per-round federated learning duration and centralized learning's epoch-level training times are visualized in Figure 9's left panel. Comparative temporal allocations between communication and computation phases across both methodologies are graphically represented in Figure 8's right section.

The federated learning framework completes its entire cycle in 435.82 seconds, with data transmission occupying 118.13 seconds (approximately 27.1% of total duration) while model optimization consumes 317.69 seconds (72.9% of overall process time). Comparatively, centralized learning demonstrates different temporal distribution patterns - requiring 807.25 seconds total duration where 302.84 seconds (37.5%) are allocated to data transfer and 504.41 seconds (62.5%) dedicated to model refinement. These temporal metrics demonstrate that federated learning achieves model convergence with substantially shorter duration than centralized approaches when considering both computational processing and essential data exchange periods, while simultaneously maintaining higher proportion of model training time within total operational duration.

5.3. Communication Expenses

To assess the data transmission demands of both approaches during model training, this study employs Wireshark, a network traffic analyzer, to monitor and record the bandwidth utilization metrics. The comparative analysis presented in Figure 9 illustrates the real-time network consumption patterns observed over a 10-second interval when implementing different randomized sampling approaches during the learning phase.



a) Bandwidth usage in federated learning. b) Bandwidth usage in centralized learning.

Figure 9. Bandwidth usage.

An examination of the bandwidth waveform reveals distinct patterns during federated learning operations. The system exhibits prominent bandwidth surges occurring at approximately 4-second intervals, primarily caused by inter-node feature data exchange. Following each major surge, a 3-second stabilization phase ensues where localized model updates occur at individual nodes, accompanied by substantial reduction in network traffic. Subsequent to this quiet period, secondary bandwidth spikes emerge corresponding to the synchronization process where nodes transmit updated model parameters to the central aggregation server, followed by global model redistribution. The centralized learning paradigm demonstrates different characteristics, comprising sequential data transmission and computational processing phases. During initial data collection, continuous high-bandwidth utilization persists as nodes concurrently transmit feature parameters to the central server. Upon completion of data transfer, the system transitions to intensive computational processing where bandwidth demands significantly decrease during the server-side model optimization phase.

The average network resource usage of federated learning during the 10-second sampling cycle was measured to be 62.80 Mbps. The bandwidth usage of centralized learning in the data transmission phase and the model training phase showed a significant difference, in which the average bandwidth demand measured in the data transmission phase amounted to 148.89 Mbps, the average bandwidth usage in the model training phase was 50.78 Mbps, and the whole training cycle's combined average bandwidth consumption is 87.59Mbps.

The experimental results demonstrate that federated learning substantially decreases data transmission costs in GCN model training processes while achieving balanced resource allocation across the network. This methodology eliminates extended bandwidth monopolization requirements, exhibiting reduced dependency on high-speed network infrastructure and enhanced compatibility with constrained connectivity scenarios.

5.4. Actual Optimization Effect

The presented metrics and experimental results demonstrate the model's operational outcomes within the dataset framework. To evaluate the model's capacity to enhance collaborative operations among physical IoT devices, our testing environment meticulously replicates the device arrangement specifications documented in the MQTTset benchmark. The implemented hardware configuration details appear in Table 5.

Table 5. Device configuration.

Sensor	Brand and model	Working power	Standby power
Temperature	DHT22	0.3mW	1μW
Humidity	DHT22	0.3mW	1μW
Intelligent air-conditioner	Mitsubishi electric MSZ-LN25VG	0.14 - 0.59 kW	1W
Motion sensor	PIR HC-SR501	0.5W	50μW
CO-Gas	MQ-7	350mW	—
Smoke	MQ-2	800mW	—
Fan controller	Sonoff Basic R3	1W	0.5W
Door lock	August smart lock Pro	1W	50μW
Fan sensor	A3144 hall sensor	10mW	1mW
Motion sensor	PIR HC-SR501	0.5W	50μW

Following the establishment of a cooperative network through the aforementioned devices, the system undergoes continuous operation for three hours. Throughout this duration, dynamic modifications are introduced to environmental parameters including temperature, humidity, CO levels, and smoke density, while simultaneously implementing stochastic variations in each device's operational power consumption. The generated operational metrics during this phase are systematically recorded and stored. Leveraging the pre-trained GCN models developed through federated and centralized approaches as detailed in chapter 4, these newly acquired IoT cooperative datasets serve as training material for subsequent model refinement. Both learning paradigms (federated and centralized) are reapplied to retrain the respective architectures until reaching reconvergence for both frameworks.

Prior to conducting the experimental trials, specific environmental parameters were established: indoor thermal conditions were maintained within 24.5-25.5°C range, relative humidity regulated at 35-45%, with carbon monoxide levels under 30ppm and smoke density thresholds below 400ppm. To emulate the fixed behavioral patterns observed in conventional IoT device coordination, a pre-trained centralized learning architecture with locked parameters was implemented. This configuration, preserving its original weight values without subsequent adjustments, served as the operational framework for the stationary cooperative network throughout the investigation.

Following the initiation of testing procedures, multiple environmental parameters were introduced into the experimental setup. At 10-minute intervals, a randomly chosen sensor node was temporarily deactivated for 60-second periods. Throughout the testing phase, the static collaborative network operated under predefined cooperative protocols, while comparative analysis involved federated learning GCN and centralized learning GCN models. These control models conducted learning processes using freshly acquired equipment data every three minutes to refresh their parameter configurations. When environmental deviations occurred, gradient optimization techniques were employed to determine minimal operational energy requirements for restoring target conditions, subsequently modifying operational settings across devices. Statistical analysis revealed distinct temporal and energy expenditure characteristics among three maintenance approaches during the 120-minute observation period: static collaboration, federated learning GCN optimization, and centralized learning GCN optimization, with detailed comparative metrics documented in Table 6.

Table 6. Comparison of collaboration modes among various devices.

Collaborative approach	Time to maintain the target environment	Power consumption
Static collaboration	56min (+3.7%)	0.85kW·h
Optimization based on federated learning and GCN	89min (+2.3%)	0.67kW·h
Optimization based on centralized learning and GCN	85min (+3.7%)	0.70kW·h

The findings presented in Table 6 demonstrate that in contrast to the fixed collaborative approach employed by conventional IoT systems, both federated learning GCN-based and centralized learning GCN-based optimization frameworks achieve extended environmental persistence while lowering overall power requirements. Notably, the enhancement capabilities of the federated learning GCN framework surpass those of its centralized counterpart, particularly in preserving operational continuity under dynamic conditions. This performance differential stems from federated learning's distributed architecture which better accommodates data privacy requirements while maintaining model effectiveness.

Compared to centralized learning approaches, federated learning demonstrates notable strengths in safeguarding data privacy, minimizing communication overhead, and enhancing training effectiveness. By preserving sensitive information on client devices and exchanging only model parameter updates, this distributed approach eliminates centralized data repositories, substantially strengthening privacy safeguards and mitigating potential data exposure risks. The framework employs innovative techniques like communication protocol optimization and parameter quantization to minimize network data transfers, thereby substantially lowering communication expenditures. Contrasting with centralized systems that necessitate complete data migration to central servers, federated architectures exclusively transmit incremental model adjustments, effectively addressing bandwidth constraints and latency challenges inherent in large-scale distributed environments. Parallelized model training across distributed nodes accelerates convergence rates while maintaining computational efficiency, particularly beneficial in geographically dispersed deployments where coordinated learning processes optimize resource utilization. This methodology consequently achieves dual objectives of robust privacy preservation and efficient resource utilization, establishing federated learning as a superior paradigm for secure, scalable machine learning implementations.

6. Conclusions

Through comprehensive analysis, this research investigates the intricate relationship between digital literacy and educational outcomes among adolescents. The findings reveal that enhanced digital competencies significantly correlate with improved academic performance, particularly in Science, Technology, Engineering, and Mathematics (STEM)-related disciplines. This investigation advances current scholarly discourse by introducing a novel methodological framework that integrates cognitive and socio-technical dimensions of digital engagement.

Several constraints merit consideration, including restricted sample diversity and regional coverage, which may affect generalizability. Future investigations could expand data collection to encompass varied demographic groups and longitudinal assessments. Furthermore, exploring pedagogical interventions that systematically cultivate digital proficiencies presents a valuable research trajectory. Emerging technologies like adaptive learning systems and AI-driven educational platforms offer promising avenues for developing targeted digital literacy programs.

The experimental results and empirical findings confirm that the GNNs-enhanced federated learning approach for IoT device coordination substantially improves operational coordination efficiency in compact smart home ecosystems. The synergistic combination of distributed machine learning with graph-based neural architectures represents a core methodological advancement in this investigation. Benchmark comparisons against centralized approaches reveal that while maintaining comparable accuracy levels and result consistency, the proposed federated framework achieves notable reductions in computational duration and data transmission requirements during model training. These efficiency gains effectively mitigate network congestion, positioning the technique as particularly advantageous for real-world implementations requiring frequent model iterations under constrained bandwidth conditions. The developed framework demonstrates enhanced cooperative performance and strengthened privacy preservation mechanisms for interconnected smart devices. Empirical evaluations highlight substantial improvements in resource distribution optimization and communication protocol efficiency, offering valuable insights for next-generation intelligent network architectures. Current limitations involve the use of idealized simulation parameters that may not fully capture real-world operational complexities. Additionally, the scalability and reliability of the proposed methodology in extensive deployment scenarios require additional validation through field testing. Subsequent investigations will focus on enhancing algorithmic robustness and predictive precision, potentially integrating dynamic parameter adaptation mechanisms and heterogeneous data integration strategies to improve performance in complex operational environments.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Funding

Anhui Provincial Research Program Preparation Project: Research on Big Data Audit Security Technology Based on Blockchain (2024AH050534).

Contribution

Cheng Zhang contributed to research design, methodology optimization, and revision. Yuanquan Zhong was responsible for experimental design, data collection and analysis, and assisted in writing experimental sections. Both authors reviewed and approved the final manuscript.

References

[1] AbdulRahman S., Tout H., Ould-Slimane H., Mourad A., Talhi C., and Guizani M., "A Survey on Federated Learning: The Journey from Centralized to Distributed On-Site Learning and Beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476-5497, 2020. DOI:10.1109/JIOT.2020.3030072

[2] Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., and Ayyash M., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2023. DOI:10.1109/COMST.2015.2444095

[3] Almusaylim Z. and Zaman N., "A Review on Smart Home Present State and Challenges: Linked

to Context-Awareness Internet of Things (IoT)," *Wireless Networks*, vol. 25, pp. 3193-3204, 2019. https://doi.org/10.1007/s11276-018-1712-5

[4] Asif N., Sarker Y., Chakrabortty R., Ryan M., Ahamed M., Saha D., Badal F., Das S., Ali M., and Moyeen S., "Graph Neural Network: A Comprehensive Review on Non-Euclidean Space," *IEEE Access*, vol. 9, pp. 60588-60606, 2021. DOI:10.1109/ACCESS.2021.3071274

[5] Bhardwaj S., Harit S., Shilpa., and Anand D., "Message Queuing Telemetry Transport-Secure Connection: A Power-Efficient Secure Communication," *International Journal of Sensor Networks*, vol. 42, no. 1, pp. 29-40, 2023. https://doi.org/10.1504/ijsnet.2023.131246

[6] Bhatti U., Tang H., Wu G., Marjan S., and Hussain A., "Deep Learning with Graph Convolutional Networks: An Overview and Latest Applications in Computational Intelligence," *International Journal of Intelligent Systems*, vol. 2023, no. 1, pp. 1-28, 2023. https://doi.org/10.1155/2023/8342104

[7] Chakraborty A., Islam M., Shahriyar F., Islam S., Zaman H., and Hasan M., "Smart Home System: A Comprehensive Review," *Journal of Electrical and Computer Engineering*, vol. 2023, no. 1, pp. 1-30, 2023. https://onlinelibrary.wiley.com/doi/full/10.1155/2023/7616683

[8] Chen M., Shlezinger N., Poor H., Eldar Y., and Cui S., "Communication-Efficient Federated Learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, pp. 1-8, 2021. https://doi.org/10.1073/pnas.2024789118

[9] Chi L., Jiang B., and Mu Y., "Fast Fourier Convolution," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4479-4488, 2020. https://papers.nips.cc/paper_files/paper/2020/file/2fd5d41ec6cfab47e32164d5624269b1-Paper.pdf

[10] Chowdhury A., Verma G., Rao C., Swami A., and Segarra S., "Unfolding WMMSE Using Graph Neural Networks for Efficient Power Allocation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 6004-6017, 2021. DOI:10.1109/TWC.2021.3071480

[11] Chung S., "Applications of Smart Technologies in Logistics and Transport: A Review," *Transportation Research Part E: Logistics and Transportation Review*, vol. 153, pp. 102455, 2021. https://doi.org/10.1016/j.tre.2021.102455

[12] Eisen M. and Ribeiro A., "Optimal Wireless Resource Allocation with Random Edge Graph Neural Networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2977-2991, 2020. DOI:10.1109/TSP.2020.2988255

[13] Guo J. and Yang C., "Learning Power Allocation for Multi-Cell-Multi-User Systems with Heterogeneous Graph Neural Networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 884-897, 2021. DOI:10.1109/TWC.2021.3100133

[14] Guo K., Hu Y., Qian Z., Sun Y., Gao J., and Yin B., "Dynamic Graph Convolution Network for Traffic Forecasting Based on Latent Network of Laplace Matrix Estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1009-1018, 2020. DOI:10.1109/TITS.2020.3019497

[15] Hamilton W., Ying Z., and Leskovec J., "Inductive Representation Learning on Large Graphs," *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, California, pp. 1025-1035, 2017. https://dl.acm.org/doi/10.5555/3294771.329486

[16] Kairouz P., McMahan H., Avent B., and Bellet A., "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1-2, pp. 1-210, 2021. http://dx.doi.org/10.1561/2200000083

[17] Lee J., Cheng Y., Niyato D., Guan Y., and Gonzalez D., "Intelligent Resource Allocation in Joint Radar-Communication with Graph Neural Networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11120-11135, 2022. DOI:10.1109/TVT.2022.3187377

[18] Levie R., Huang W., Bucci L., Bronstein M., and Kutyniok G., "Transferability of Spectral Graph Convolutional Neural Networks," *Journal of Machine Learning Research*, vol. 22, pp. 1-59, 2021. https://jmlr.csail.mit.edu/papers/volume22/20-213/20-213.pdf

[19] Li G. and Wang F., "Image Object and Scene Recognition Based on Improved Convolutional Neural Network," *The International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 925-937, 2024. https://doi.org/10.34028/iajit/21/5/13

[20] Li L., Fan Y., Tse M., and Lin K., "A Review of Applications in Federated Learning," *Computers and Industrial Engineering*, vol. 149, pp. 106854, 2020. https://doi.org/10.1016/j.cie.2020.106854

[21] Li X., Bai Y., and Kang Y., "Exploring the Social Influence of the Kaggle Virtual Community on the M5 Competition," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1507-1518, 2022. https://doi.org/10.1016/j.ijforecast.2021.10.001

[22] Lim W., Luong N., Hoang D., Jiao Y., Liang Y., Yang Q., Niyato D., and Miao C., "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 2031-2063, 2020. DOI:10.1109/COMST.2020.2986024

[23] Miller C., Picchetti B., Fu C., and Pantelic J., "Limitations of Machine Learning for Building Energy Prediction: ASHRAE Great Energy Predictor III Kaggle Competition Error Analysis," *Science and Technology for the Built*

Environment, vol. 28, no. 5, pp. 610-627, 2022. https://doi.org/10.1080/23744731.2022.2067466

[24] Mistry I., Tanwar S., Tyagi S., and Kumar N., "Blockchain for 5G-Enabled IoT for Industrial Automation: A Systematic Review, Solutions, and Challenges," *Mechanical Systems and Signal Processing*, vol. 135, pp. 106382, 2020. https://doi.org/10.1016/j.ymssp.2019.106382

[25] Naresh M. and Munaswamy P., "Smart Agriculture System Using IoT Technology," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 98-102, 2019. https://ieeeprojectsmadurai.com/IEEE%202019%20IOT%20BASEPAPERS/5_IOT%20SMART%20AGRICULTURE.pdf

[26] Oralhan Z., Oralhan B., Yigit Y., "Smart City Application: Internet of Things (IoT) Technologies Based Smart Waste Collection Using Data Mining Approach and Ant Colony Optimization," *The International Arab Journal of Information Technology*, vol. 14, no. 4, pp. 423-427, 2017. https://iajit.org/paper/4951

[27] Pfitzner B., Steckhan N., and Arnrich B., "Federated Learning in A Medical Context: A Systematic Literature Review," *ACM Transactions on Internet Technology*, vol. 21, no. 2, pp. 1-31, 2021. https://doi.org/10.1145/3412357

[28] Shao Z., Xu Y., Wei W., Wang F., Zhang Z., and Zhu F., "Heterogeneous Graph Neural Network with Multi-View Representation Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11476-11488, 2022. DOI:10.1109/TKDE.2022.3224193

[29] Shetty S. and Shenoy U., "Performance of Static IoT Networks Using RPL Objective Functions," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 8972-8977, 2019. DOI:10.35940/ijrte.D9776.118419

[30] Singh D., Singh R., Gupta A., and Pawar A., "Message Queue Telemetry Transport and Lightweight Machine-to-Machine Comparison Based on Performance Efficiency Under Various Scenarios," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6293-6293, 2022. http://doi.org/10.11591/ijece.v12i6.pp6293-6302

[31] Stankovic L., Mandic D., Dakovic M., Scalzo B., Brajovic M., Sejdic E., and Constantinides A., "Vertex-Frequency Graph Signal Processing: A Comprehensive Review," *Digital Signal Processing*, vol. 107, pp. 102802, 2020. https://doi.org/10.1016/j.dsp.2020.102802

[32] Tao Z., Wei Y., Wang X., He X., Huang X., and Chua T., "MGAT: Multimodal Graph Attention Network for Recommendation," *Information Processing and Management*, vol. 57, no. 5, pp. 102277, 2020. https://doi.org/10.1016/j.ipm.2020.102277

[33] Wang T. and Melchior P., "Graph Neural Network-Based Resource Allocation Strategies for Multi-Object Spectroscopy," *Machine Learning: Science and Technology*, vol. 3, no. 1, pp. 015023, 2022. DOI:10.1088/2632-2153/ac4d12

[34] Wen J., Zhang Z., Lan Y., Cui Z., Cai J., and Zhang W., "A Survey on Federated Learning: Challenges and Applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513-535, 2023. https://doi.org/10.1007/s13042-022-01647-y

[35] Xu Z., Hou L., Zhang Y., and Zhang J., "Passenger Flow Prediction of Scenic Spot Using a GCN-RNN Model," *Sustainability*, vol. 14, no. 6, pp. 3295-3295, 2022. DOI:10.3390/su14063295

[36] Zhang C., Li W., Wei D., Liu Y., and Li Z., "Network Dynamic GCN Influence Maximization Algorithm with Leader Fake Labeling Mechanism," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 3361-3369, 2022. DOI:10.1109/TCSS.2022.3193583

[37] Zhang S., Tong H., Xu J., and Maciejewski R., "Graph Convolutional Networks: A Comprehensive Review," *Computational Social Networks*, vol. 6, no. 1, pp. 1-23, 2019. https://doi.org/10.1186/s40649-019-0069-y

[38] Zhang T., Gao L., He C., Zhang M., Krishnamachari B., and Avestimehr A., "Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24-29, 2022. DOI:10.1109/IOTM.004.2100182

[39] Zhou J., Cui G., Hu S., Zhang Z., Yang C., Liu Z., Wang L., Li C., and Sun M., "Graph Neural Networks: A Review of Methods and Applications," *AI Open*, vol. 1, pp. 57-81, 2020. https://doi.org/10.1016/j.aiopen.2021.01.001

[40] Zhu J., Han X., Deng H., Tao C., Zhao L., Wang P., Lin T., and Li H., "KST-GCN: A Knowledge-Driven Spatial-Temporal Graph Convolutional Network for Traffic Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15055-15065, 2022. DOI:10.1109/TITS.2021.3136287

[41] Zou Y., Feng W., Zhang J., and Li J., "Forecasting of Short-Term Load Using the MFF-SAM-GCN Model," *Energies*, vol. 15, no. 9, pp. 3140-3140, 2022. DOI:10.3390/en15093140

Cheng Zhang was born in Hefei, Anhui Province, China in 1982. He graduated from Anhui University and now works in Anhui Audit College. His main research direction is Big Data Technology Application and other fields. He has published over 5 high-level papers in SCI, EI, and other fields.

Yuanquan Zhong was born in Fuyang, Anhui Province, China in 1981. He graduated from Hefei University of Technology and is now working at the School of Computer Engineering, Anhui Wenda Information Engineering College. His research interests include Computer Pattern Recognition, Deep Learning and other fields.