Decentralized IoT-Edge Computing: An LSTM-based Federated Learning Framework for Personalized Task Failure Prediction

Nawaz Ali
Department of Computer Modeling Electronic, and System Engineering
University of Calabria, Rende, Italy
nawaz.ali@dimes.unical.it

Mir Hassan
Department of Information Engineering and Computer Science
University of Trento, Italy
mir.hassan@unitn.it

Ali Hassan Sodhro
Department of Computer Science
Kristianstad University, Sweden
ali.hassan_sodhro@hkr.se

Gianluca Aloi
Department of Computer Modeling, Electronic, and System Engineering
University of Calabria, Rende, Italy
gianluca.aloi@unical.it

Raffaele Gravina
Department of Computer Modeling, Electronic, and System Engineering
University of Calabria, Rende, Italy
r.gravina@dimes.unical.it

Claudio Savaglio
Department of Computer Modeling, Electronic, and System Engineering
University of Calabria, Rende, Italy
csavaglio@dimes.unical.it

Giovanni Iacca
Department of Information Engineering and Computer Science
University of Trento, Italy
giovanni.iacca@unitn.it

Giancarlo Fortino
Department of Computer Modeling,
Electronic, and System Engineering
University of Calabria, Rende, Italy
giancarlo.fortino@unical.it

***Abstract*—Task failures in decentralized Internet of Things (IoT)-edge computing environments not only lead to inefficiencies, increased latency, and resource wastage but can also introduce system instability and cause application malfunctions. These failures may arise due to network disruptions, resource constraints, or inefficient task scheduling, ultimately affecting the overall reliability and performance of IoT-edge systems. This study presents a novel Long Short-Term Memory (LSTM)-based Federated Learning (FL) framework for proactive task failure prediction, ensuring adaptive scheduling and efficient resource utilization. Unlike existing conventional methods, our approach personalizes failure prediction per device, addressing heterogeneous execution characteristics while preserving data privacy. By integrating LSTM with FL, we improve the failure detection accuracy and reduce unnecessary task executions. We first trained all models using Federated Learning (FL) and then conducted a comparative analysis of Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and LSTM. Our findings show that LSTM achieves the highest accuracy and F1 score, while CNN excels in recall and energy efficiency. These insights validate the effectiveness of our FL-based failure prediction framework and highlight the advantages of model personalization for dynamic decentralized IoT-edge environments.**

***Index Terms*—Decentralized, Internet of Things, Federated Learning, Edge Computing, Task Failure Prediction, Adaptive Scheduling, Privacy Preservation**

I. Introduction

IoT and Edge computing have emerged as a fundamental paradigm in modern decentralized and distributed systems, enabling low-latency data processing for applications such as autonomous vehicles, smart cities, and industrial automation. Federated Learning (FL) offers a privacy-preserving approach to decentralized model training, allowing edge devices to collaboratively refine models without exposing raw data. However, despite its advantages, FL-based IoT-edge computing systems suffer from task execution failures, which arise due to limited computational resources, unpredictable network conditions, and fluctuating energy availability [1]. These failures lead to inefficiencies, increased latency, and suboptimal resource utilization, necessitating a proactive and energy-efficient failure prediction mechanism. Existing FL methodologies focus mainly on optimizing resource allocation and client selection [2], [3], but lack real-time predictive mechanisms for task failures. Conventional task scheduling frameworks either assume perfect resource availability [4] or react to failures after they occur, resulting in inefficient system performance. Additionally, traditional approaches do not account for heterogeneous device constraints, leading to generalized models that fail to capture device-specific failure risks [5]. A personalized FL approach that accurately predicts task failures before execution is critical to improving system reliability, reducing energy consumption, and improving decision-making in dynamic decentralized IoT-edge environments.

A. Problem Statement

Despite advancements in Federated Edge Learning (FEEL), the absence of real-time task failure prediction leads to excessive resource wastage and increased system instability. Existing methods emphasize secure model aggregation [6], communication-efficient scheduling [7], or hierarchical learning architectures [8], but the investigation of an integrated solution (i.e., combining failure prediction, privacy preservation, and energy-aware scheduling) is overlooked and has not been yet examined by researchers.

Current FL-based scheduling approaches optimize training efficiency but do not proactively anticipate task failures [9]. On the other hand, IoT-edge devices operate under strict energy budgets, making it necessary to integrate power-aware failure prediction [10]. Traditional task failure models require data centralization, introducing privacy concerns and communication bottlenecks [4]. Generic models do not account for device-specific execution conditions, leading to inaccurate failure estimations [2].

To address these gaps, we propose a Long Short-Term Memory (LSTM)-based FL framework for proactive task failure prediction, ensuring efficient scheduling and adaptive resource utilization in decentralized IoT-edge computing environments.

B. Contributions

In this paper, we introduce a novel FL-based task failure prediction framework that integrates LSTM networks to predict failures in real time and optimize energy-efficient task scheduling. Our contributions are summarized as follows:

- We develop an LSTM-based model that analyzes sequential execution data to estimate failure probabilities before task execution, preventing inefficiencies.
- We employ Federated Personalization to fine-tune failure prediction models per device, capturing heterogeneous execution characteristics. Our framework leverages Federated Personalization within a decentralized IoT-Edge computing setup, where LSTM-based Federated Learning adapts to individual device behaviors.
- Unlike traditional failure prediction techniques, our method enables on-device learning within a decentralized IoT-Edge computing framework. By leveraging LSTM-based Federated Learning, only encrypted model updates are shared, reducing communication overhead while ensuring personalized task failure prediction.
- Our approach integrates real-time failure prediction with adaptive scheduling, dynamically reallocating resources to prevent unnecessary failures.

Our manuscript is organized as follows. In Section II, we briefly overview the related works, covering the main existing methods and their limitations. We present our method in Section III. The experimental setup and results are presented in Section IV. Finally, we conclude the paper in Section V

II. Related Work

FL has been widely studied in edge computing environments for various applications, including task scheduling, energy efficiency, and security. However, most existing methods fail to address the problem of real-time task failure prediction, energy-aware scheduling, and personalized FL models. This section critically reviews the most relevant works and highlights their limitations.

Cao et al. [1] introduced a Multi-Task Asynchronous Federated Learning (MTAFL) framework that optimizes resource allocation and client scheduling to minimize system-wide energy consumption. While their approach improves learning efficiency, it does not incorporate real-time failure prediction, making it ineffective for dynamic environments where task failures must be anticipated. Jiang et al. [11] proposed a specialized FL scheme (CuFL) algorithm tailored to heterogeneous edge computing devices. Their work improves model adaptability but does not consider resource constraints or task failure prevention, which limits its applicability in energy-constrained systems. Kang et al. [10] explored an optimization-based task assignment mechanism using blockchain for FEEL. Their approach improves task reliability through device evaluation but lacks predictive failure mechanisms, assuming that optimal task allocation alone is sufficient to prevent execution failures. Sun et al. [2] proposed a dynamic scheduling framework for FEEL that optimizes energy constraints by predicting the computational and communication energy required for training. However, their method does not account for the effect of task failures on overall system performance, leading to inefficiencies in real-world applications. Zeng et al. [3] investigated energy-efficient radio resource allocation for FEEL by adjusting the bandwidth distribution. Although their study effectively reduces energy consumption, it does not address task execution failures, which remain a critical challenge in resource-constrained edge environments. Shi et al. [4] proposed a task scheduling mechanism in Mobile Edge Computing (MEC) that integrates FL with collaborative computing. Their method reduces task completion time but does not include predictive modeling for failures, which can lead to ineffective task execution strategies in high-variance workloads. Li et al. [5] developed a blockchain-based hierarchical FL system (PoFEL) for security and privacy preservation. While their approach ensures secure model aggregation, it does not account for real-time failure prediction, making it unsuitable for energy-sensitive edge applications that require proactive adaptation. Hu et al. [6] explored FL-based energy-efficient task scheduling with Lyapunov optimization for wireless networks. Their work optimizes scheduling but does not integrate personalized FL models, making it less adaptable to heterogeneous edge devices with varying failure risks. Deng et al. [7] presented a semi-asynchronous FL approach with trajectory prediction for vehicular edge computing. While their method accounts for dynamic behavior in mobile edge devices, it does not incorporate failure prediction, which is crucial for maintaining task reliability. Cai et al. [8] introduced an optimization framework for task scheduling in FL with energy-efficient strategies. However, their work does not personalize failure prediction models, leading to potential inefficiencies in device-specific training.

Our proposed approach addresses the limitations (i.e., absence of real-time task failure prediction, and lack of personalized FL models for heterogeneous edge devices) by integrating FL with real-time failure prediction using LSTM models, energy-efficient task scheduling, and optimization of federated personalization. By proactively predicting failures, our model ensures improved task reliability, reduced energy consumption, and adaptive learning for heterogeneous IoT-edge devices in decentralized systems.

A. Comparison with Existing Works

Table I provides a comparative analysis of our approach with existing methods.

TABLE I
COMPARISON OF FEDERATED LEARNING-BASED METHODS

Method	Failure Prediction	Energy Awareness	Personalization
Cao et al. [1]	×	✓	✓
Jiang et al. [11]	×	×	✓
Kang et al. [10]	×	✓	×
Sun et al. [2]	×	✓	×
Shi et al. [4]	×	✓	×
Li et al. [5]	×	×	✓
Deng et al. [7]	×	×	✓
Our Approach	✓	✓	✓

III. PROPOSED METHOD

In this section, we present our proposed methodology for task failure prediction using FL, ensuring decentralized training while maintaining data privacy.

A. Federated Learning Workflow

Our proposed approach employs FL to train a distributed LSTM model for task failure prediction while preserving user data privacy. The methodology follows these key steps:

1) **Local Training at Edge Devices**: Each edge device trains an LSTM model using historical task execution data, including CPU load, memory usage, energy consumption, and network conditions.
2) **Model Update Sharing**: Instead of transmitting raw data, edge devices send model weight updates to the aggregation server, reducing communication overhead while maintaining privacy.
3) **Federated Model Aggregation**: The server aggregates the received updates using Federated Averaging (FedAvg) [12], where updates from multiple edge devices are combined using a weighted averaging strategy to form an improved global model.
4) **Global Model Distribution**: The updated global model is redistributed to edge devices for further local training and refinement, ensuring continuous learning across the system.

To facilitate a better understanding of the proposed FedAvg-based task failure prediction, we present the FL process in Algorithm 1 and its corresponding workflow illustration in Figure 1.

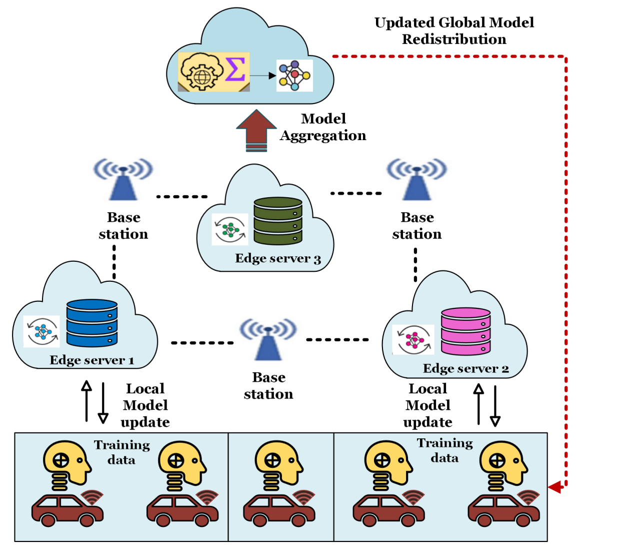

Fig. 1. Overview of the proposed Federated Learning Workflow for Task Failure Prediction.

Algorithm 1 Federated Learning for Task Failure Prediction

Initialize global LSTM model M_g
for each FL round $t = 1, 2, \ldots, T$ **do**
  Server selects a subset of edge devices $\mathcal{S}_t$ and distributes M_g to them
  for each device $i \in \mathcal{S}_t$ **in parallel do**
    Train local LSTM model M_i using task failure history
    Compute failure probability P^i_{fail}
    if $P^i_{fail} > T_{fail}$ **then**
      Trigger task rescheduling for device i
    end if
    Compute local model update $\Delta M_i = M_i - M_g$
    Send ΔM_i to the server
  end for
  FedAvg Aggregation:
    $M_g \leftarrow M_g + \eta \sum_{i \in \mathcal{S}_t} w_i \Delta M_i$ {Federated Averaging with learning rate η}
  Server updates and redistributes M_g
end for

B. Privacy-Preserving Training

FL ensures privacy preservation by keeping raw data on local IoT-edge devices in a decentralized system. To further reduce communication overhead, we implement an event-triggered update mechanism, which selectively transmits model updates only when significant changes are detected. The mechanism consists of the following steps:

- **Adaptive Update Triggering**: Model updates are transmitted only if a significant deviation in local training metrics is observed, preventing unnecessary communication.

- **Dynamic Client Selection**: Clients are selected dynamically based on computational capacity and network availability, ensuring load balancing and efficiency.

The computed failure probability is then compared against a predefined threshold (T_{fail}). If $P^i_{fail} > T_{fail}$, the system triggers task rescheduling to mitigate potential execution failures.

IV. Experimental Setup and Result Evaluation

To evaluate our proposed approach, we conducted experiments on a Linux workstation equipped with an NVIDIA TITAN Xp GPU, 62 GB of memory, and CUDA v11.7. The framework was implemented in Python, utilizing PyTorch for model training and the Flower framework to simulate the FL environment.

Each experiment involved a total of five clients, where each client received unique data subsets for local training. The hyperparameters used in our experiments are summarized in Table II.

TABLE II
HYPERPARAMETER CONFIGURATION USED IN THE EXPERIMENTS.

Hyperparameter	**Value**
Learning rate	0.0005
Number of communication rounds	50
Local epochs per client	30
Batch size	64
Input size	12
Hidden size	64
Number of layers	2
Output size	12

This setup ensured an effective evaluation of our method in a FL scenario, leveraging distributed data across multiple clients while optimizing model performance.

A. Dataset

The simulation environment uses EdgecloudSim to simulate computing, network and vehicle mobility, as discussed in our previous work [13], [14]. Training data are created using a random orchestration policy in various load scenarios, following [13], [14].

EdgecloudSim simulates computing and networking resources for a transportation edge computing system, while Simulation of Urban MObility (SUMO) generates vehicle traces.

WLAN, MAN, WAN, and GSM-based cellular networks are simulated in EdgeCloudSim using a single server queuing model. Using the MMP/M/1 queuing model [15], which considers Poisson arrival rate and task size, a wrapper class manages these communication standards' task size and transmission latency.

It should be noted that the current EdgeCloudSim GSM model oversimplifies real-world GSM networks. Instead of the simple GSM model, a 5G model with the Closed-IN (CI) path loss model was used to simulate frequency, distance, and a realistic path loss exponent. For further details, see the [13], [14]. In addition, the more details on the relationship between energy-efficiency and relaibility can be found in this work [16].

We determine the nearest edge data center based on signal strength within one Kilometer (km) and choose the one with the strongest signal. We simulated one car moving into the SUMO map and edge data centers, assigning it to a base station ID based on signal strength.

This simulation, EdgeCloudSim, generates a comprehensive dataset on energy consumption across mobile devices, edge servers, and cloud computing infrastructure [14]. To assess the efficiency of edge computing, EdgeCloudSim systematically monitors power usage trends.

A key aspect of this dataset is the tracking of CPU energy consumption, capturing power usage during both active processing and idle states. Task energy consumption is determined based on computational workload, measured in Million Instructions Per Second (MIPS), effectively representing the dynamic power demands of processor activities.

B. Baselines

To validate the effectiveness of our proposed FedAvg-based LSTM model for task failure prediction, we compare it against two widely used deep learning architectures: Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU), which are tested with FL under the same setup as LSTM. These models serve as strong baselines due to their effectiveness in pattern recognition and sequential data processing. We select CNN as a baseline because it excels at feature extraction and has been successfully applied in various classification tasks. However, CNN is not optimized for capturing temporal dependencies, which are crucial for predicting task failures in dynamic edge computing environments. GRU, on the other hand, is a kind of recurrent neural network that efficiently models sequential dependencies while reducing computational overhead compared to LSTM. It provides a balance between performance and efficiency, making it an ideal baseline for comparison.

C. Experimental Results

The results of our comparative analysis across the different neural architectures considered are presented in Table III. The evaluation metrics include accuracy, precision, recall, F1 score, energy consumption based on the dataset from [14], and task completion time, ensuring a comprehensive assessment of each method's performance. Additionally, we analyze task failure rates across different numbers of vehicles and the number of connected devices per edge server over time.

The LSTM model achieves the highest accuracy (**84.04%**) and F1 score (**79.33%**), indicating its superior overall predictive capability. GRU, on the other hand, demonstrates the highest precision (**88.38%**), suggesting that it makes fewer false-positive predictions. CNN significantly outperforms the other models in recall (**85.65%**), meaning it is less likely to miss positive instances. Regarding energy consumption, CNN consumes the least power (**74.25J**), whereas LSTM performs best in terms of task completion time (**2.63s**).

TABLE III
SUMMARY OF THE NUMERICAL RESULTS.

Method	Accuracy	Precision	Recall	F1 Score	Energy Consumption (J)	Task Completion (s)
CNN-FL (Baseline1)	64.10 $\pm$ 3.32	54.94 $\pm$ 3.25	**85.65 $\pm$ 3.36**	66.85 $\pm$ 2.31	**74.25 $\pm$ 11.31**	2.94 $\pm$ 1.20
GRU-FL (Baseline2)	83.91 $\pm$ 2.28	**88.38 $\pm$ 3.57**	70.63 $\pm$ 4.42	79.13 $\pm$ 3.48	69.93 $\pm$ 15.86	3.63 $\pm$ 1.04
LSTM-FL	**84.04 $\pm$ 2.36**	87.54 $\pm$ 4.23	72.57 $\pm$ 2.78	**79.33 $\pm$ 2.96**	81.67 $\pm$ 15.60	**2.63 $\pm$ 1.09**

The best accuracy of LSTM can be attributed to its ability to retain long-term dependencies through its gating mechanism, which allows it to capture sequential information more effectively. This makes it well-suited for tasks requiring temporal dependencies. GRU's higher precision suggests that it is more selective in its positive predictions, possibly due to its simplified gating mechanism, which reduces the likelihood of overfitting compared to LSTM. CNN achieves the best recall, likely because it focuses on feature extraction rather than temporal dependencies, making it better at identifying positive instances even if some are false positives.

To further understand model performance, we analyze the confusion matrices for all three architectures, as shown in Figure 2. LSTM exhibits the highest true positive rate, maintaining a balance between sensitivity and specificity. GRU, although competitive, shows slightly higher misclassification rates compared to LSTM. CNN struggles with distinguishing between classes, particularly in scenarios where precision is critical. The misclassification in CNN likely arises due to its inability to capture sequential relationships, leading to errors when distinguishing between temporally similar instances.

Figure 3 illustrates the task failure rate as a function of the number of vehicles. LSTM and CNN models maintain relatively lower failure rates across varying vehicle counts, whereas GRU exhibits more fluctuations, indicating inconsistencies in handling task load variations. The results highlight that LSTM offers a stable and reliable performance in dynamic conditions. The fluctuations in GRU can be attributed to its sensitivity to parameter tuning, which affects its learning stability over different workload conditions. CNN's stability in failure rate suggests that it relies on spatial feature extraction rather than sequential dependencies, making it less susceptible to variations in workload.

Figure 4 illustrates how energy consumption varies across processing levels as the number of vehicles increases. The device-level curve rises most steeply, indicating that local processing becomes substantially more energy-demanding with a larger fleet. In contrast, energy consumption at the edge increases at a moderate rate, demonstrating the benefit of offloading computation to nearby servers. The cloud-level curve, with its lower slope, confirms that centralized processing is the most energy-efficient, which is likely due to its optimized resource utilization. Overall, these results reveal a clear trade-off: while device-level processing minimizes latency, it incurs high energy costs, whereas cloud processing, though more efficient, may introduce additional latency. Edge computing thus serves as a balanced alternative.

The Figure 5 shows the relationship between edge server training load over time and time in Federated Learning for edge vehicular networks is typically non-linear and dynamic, influenced by vehicle density, mobility patterns, communication delays, and computational constraints at the edge.

Finally, Figure 6 presents the number of connected devices per edge server over time. Fluctuation patterns indicate that edge servers experience dynamic connectivity demands. The LSTM-based system exhibits a more balanced allocation of connected devices across servers, whereas GRU and CNN exhibit sharp transitions, leading to potential load imbalance issues. The sharp transitions in GRU and CNN may be due to their inability to smooth variations over time, resulting in abrupt changes in device allocation. LSTM's ability to handle sequential dependencies helps regulate the distribution of connected devices over time, preventing extreme imbalances. However, the observed variations suggest that rather than maintaining a strictly uniform distribution, the model dynamically adapts to changes in device connectivity patterns.

Overall, the results demonstrate that the LSTM is the most reliable model for achieving high accuracy, stability, and task efficiency. Although GRU performs well in precision, it suffers from higher fluctuations in task failure rates. CNN, despite having the best recall and the lowest energy consumption, struggles with misclassification errors. These insights guide the selection of models for real-time and energy-efficient applications in FL and edge computing scenarios. The findings highlight the importance of selecting the appropriate model based on the specific requirements of the task, balancing precision, stability, and computational efficiency.

D. Discussion and Analysis

Our preliminary experimental results demonstrate the potential of FL for task failure prediction in IoT-edge computing for decentralized systems. By leveraging deep learning models, including LSTM, GRU, and CNN, we enhance task execution reliability while maintaining energy efficiency and privacy preservation. However, several challenges remain.

Traditional FL approaches optimize resource allocation but lack real-time failure prediction, leading to inefficiencies [10], [11]. While recent studies introduce adaptive scheduling [1], they do not incorporate predictive modeling. Our framework integrates personalized FL with predictive modeling to address dynamic edge environments [7]. Unlike conventional FL models that assume homogeneous device capabilities [17], our approach adapts to heterogeneous execution conditions, improving accuracy and reducing computational overhead.

Despite its advantages, our framework has limitations. The accuracy of deep learning-based predictions depends on the

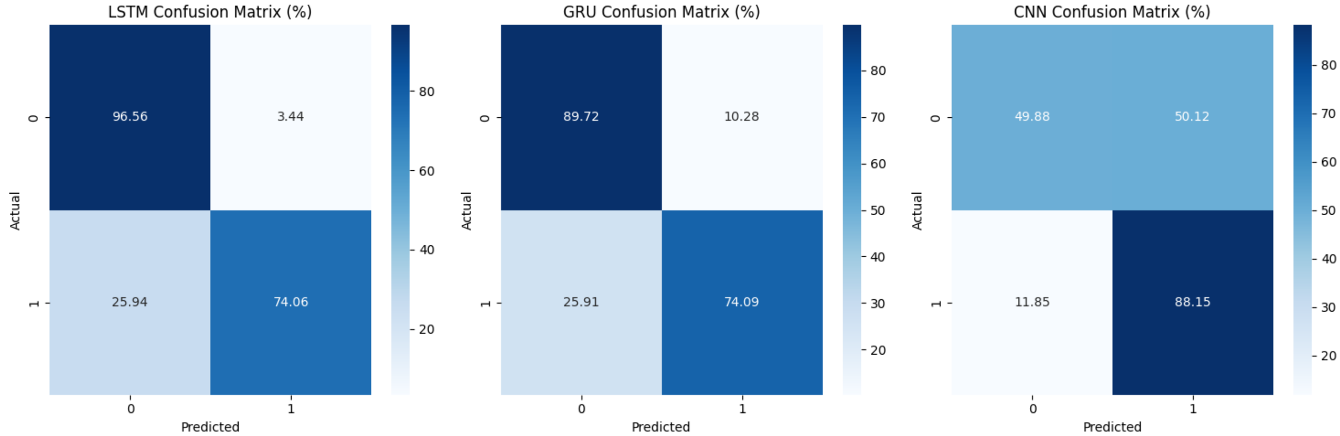

Fig. 2. Confusion matrices for LSTM, GRU, and CNN models. The LSTM model exhibits superior classification performance, whereas CNN suffers from a higher misclassification rate.

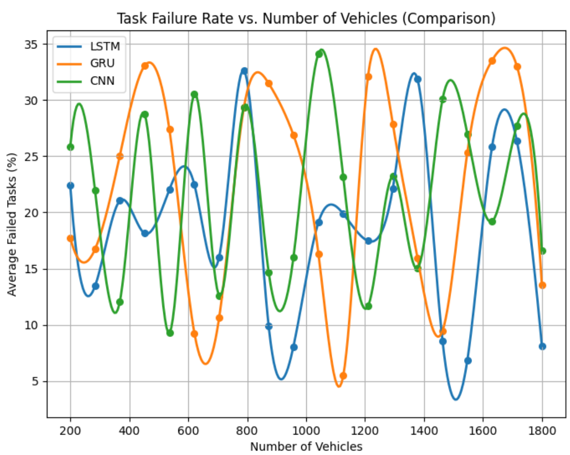

Fig. 3. Task failure rate as a function of the number of vehicles for LSTM, GRU, and CNN. LSTM demonstrates comparatively smoother variations, indicating better stability in handling task failures over different numbers of vehicles, while GRU shows fluctuating performance.

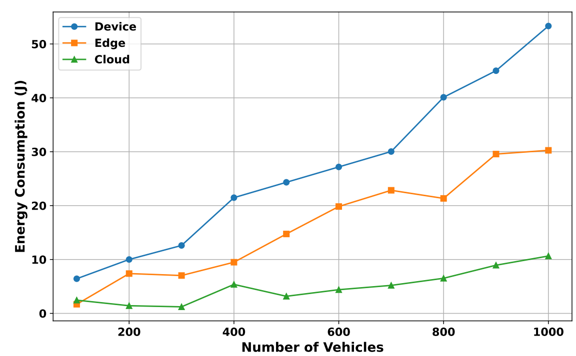

Fig. 4. Simulated energy consumption (in arbitrary units) as the number of vehicles increases from 200 to 1000. The device-level energy consumption rises most steeply, followed by the edge level, while the cloud level exhibits the lowest increase in energy consumption.

quality and quantity of local training data, making performance inconsistent between devices. Future research should explore transfer learning for better generalization [18]. Additionally, FedAvg does not account for device reliability variations. Alternative aggregation methods, such as hierarchical aggregation [1] or reputation-based worker selection [10], could improve the robustness of the model.

Further improvements include adaptive client selection using reputation-based or reinforcement learning approaches [7], optimizing energy efficiency while maintaining model precision [1], and implementing privacy-enhancing mechanisms such as homomorphic encryption or secure multiparty computation [18]. Performing the deployment of the model on real-world edge computing infrastructures [19] would validate its practical performance [17]. Addressing these challenges

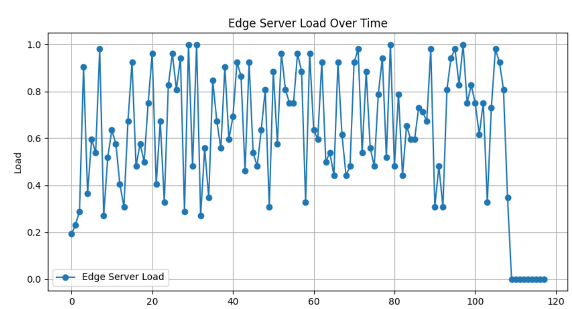

Fig. 5. Edge server training load over time.

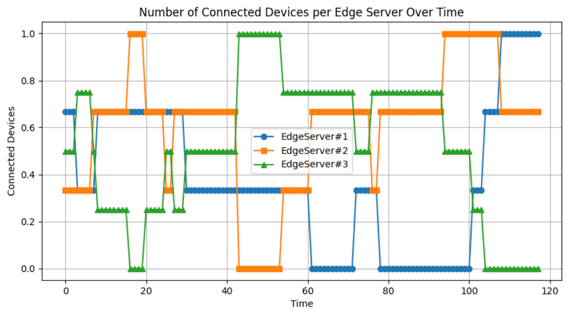

Fig. 6. Number of connected devices per edge server over time, highlighting the dynamic allocation patterns of LSTM, GRU, and CNN models.

will enhance the scalability and reliability of FL-based task failure prediction in edge computing.

V. Conclusion and Future Recommendations

This research presented a FL-based framework integrating multiple deep learning models, including LSTM, GRU, and CNN, for proactive prediction of task failure in IoT edge computing for decentralized systems. By combining real-time failure prediction with adaptive scheduling, the proposed approach improves system reliability, reduces resource wastage, and improves energy efficiency. Our experimental results demonstrate that personalized federated models outperform conventional methods by capturing device-specific execution characteristics while maintaining privacy. In the near future, we will test the proposed approach in smart city applications.

Acknowledgments

This work has been funded by the Horizon Europe research and innovation program of the European Union, under grant agreement no. 101092912, project MLSysOps, and under grant agreement no. 101071179, project SUSTAIN, and Kristianstad University, 29188 Kristianstad, Sweden. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EISMEA. Neither the European Union nor the granting authority can be held responsible for them. For the purpose of open access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

References

[1] X. Cao, O. Tao, K. Zhao, Y. Li, and X. Chen, "Efficient multi-task asynchronous federated learning in edge computing," *IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2024.

[2] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE Journal on Selected Areas in Communications*, vol. 40, pp. 227–242, 2021.

[3] Q. Zeng, Y. Du, K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2019.

[4] T. Shi, H. Tian, T. Zhang, J. Loo, J. Ou, C. Fan, and D. Yang, "Task scheduling with collaborative computing of mec system based on federated learning," in *IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, pp. 1–5, 2022.

[5] S.-N. Li, Q. Hu, and Z. Wang, "Pofel: Energy-efficient consensus for blockchain-based hierarchical federated learning," *ArXiv*, vol. abs/2308.07840, 2023.

[6] C.-H. Hu, Z. Chen, and E. G. Larsson, "Energy-efficient federated edge learning with streaming data: A lyapunov optimization approach," *ArXiv*, vol. abs/2405.12046, 2024.

[7] Y. Deng, X. Li, C. Sun, Q. Fan, X. Wang, and V. C. M. Leung, "Semi-asynchronous federated learning with trajectory prediction for vehicular edge computing," in *IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2024.

[8] W. Cai and F. Duan, "Task scheduling for federated learning in edge cloud computing environments by using adaptive-greedy dingo optimization algorithm and binary salp swarm algorithm," in *Future Internet*, 2023.

[9] V. C, J. S, and R. B, "Energy-aware federated learning for aqi pollutants forecasting in edge networks," *IEEE Transactions on Network Science and Engineering*, vol. 11, pp. 4146–4157, 2024.

[10] J. Kang, Z. Xiong, X. Li, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Optimizing task assignment for reliable blockchain-empowered federated edge learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1910–1923, 2021.

[11] H. Jiang, M. Liu, B. Yang, Q. Liu, J. Li, and X. Guo, "Customized federated learning for accelerated edge computing with heterogeneous task targets," *Computer Networks*, vol. 183, p. 107569, 2020.

[12] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2022.

[13] N. Ali, F. D. Rango, G. Caliciuri, G. Aloi, R. Gravina, and G. Fortino, "Customized edgecloudsim: Enhanced mobility and network models for urban vehicular edge computing," in *Proceedings of the 28th International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2024)*, October 2024.

[14] N. Ali, G. Aloia, F. D. Rango, C. Savaglio, and R. Gravina, "Edge-cloud continuum driven industry 4.0," in *Proceedings of the 6th International Conference on Industry 4.0 and Smart Manufacturing (ISM)*, 2024.

[15] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018.

[16] A. H. Sodhro, S. Pirbhulal, G. H. Sodhro, M. Muzammal, L. Zongwei, and A. Gurtov, "Towards 5g-enabled self adaptive green and reliable communication in intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 5223–5231, August 2021.

[17] Y. Zhang, X. Zhang, and Y. Cai, "Multi-task federated learning based on client scheduling in mobile edge computing," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 185–190, IEEE, 2022.

[18] Y. Liu, Y. Qu, C. Xu, Z. Hao, and B. Gu, "Blockchain-enabled asynchronous federated learning in edge computing," *Sensors*, vol. 21, no. 10, p. 3335, 2021.

[19] M. Hassan, L. Leonardo, S. Yildirim, G. Iacca, *et al.*, "Fededge: Federated learning with docker and kubernetes for scalable and efficient edge computing," in *INTERNATIONAL CONFERENCE ON EMBEDDED WIRELESS SYSTEMS AND NETWORKS (EWSN)...*, pp. 339–344, ACM, 2023.