Privacy Preserved Integrated Big Data Analytics Framework Using Federated Learning for Intelligent Transportation Systems

Sarah Kaleem
EIAS Data Science Lab
College of Computer and Information Sciences (CCIS)
Prince Sultan University (PSU)
Riyadh, Saudi Arabia
skaleem@psu.edu.sa

Awais Ahmad
Information Systems Department
College of Computer and Information Sciences
Imam Mohammad Ibn Saud Islamic University (IMSIU)
Riyadh, Saudi Arabia
aahmad.marwat@gmail.com

Muhammad Babar
Robotics and Internet of Things Lab, CCIS
Prince Sultan University, Riyadh, Saudi Arabia
mbabar@psu.edu.sa

*Goutham Reddy Alavalapati
Department of Computer Science
University of Illinois, Springfield, Illinois, USA 62703
galav@uis.edu

***Abstract*—This paper presents an integration of Federated Learning (FL) with Big Data Analytics (BDA) for Intelligent Transportation Systems (ITS). By leveraging the decentralized nature of FL, the framework enhances privacy, reduces latency, and improves scalability, addressing key limitations of traditional BDA approaches. This research demonstrates the potential of FL to revolutionize data analytics in ITS by enabling real-time applications and facilitating personalized insights. The key contributions of this research include the integration of FL with BDA to tackle traditional BDA challenges, the implementation of FL algorithms within the proposed integrated framework, and a comprehensive performance and scalability analysis. Additionally, the paper presents the development and validation of a specialized ITS dataset designed for FL environments. These contributions collectively highlight the transformative potential of FL in optimizing traffic management and public transportation systems through efficient and scalable data analytics. We demonstrate FL's capability to efficiently manage and analyze ITS data while maintaining user privacy and scalability. Our findings reveal that FedProx achieved the highest global accuracy at 79.61%, surpassing FedSGD at 79.10% and FedAvg at 78.01%.**

I. Introduction

The Internet of Things (IoT) has transformed many sectors, including Intelligent Transportation Systems (ITS), where connected devices generate vast amounts of Big Data related to transportation activities [1], [2]. This data can improve traffic management and enhance travel experiences, but it also presents significant challenges, particularly around privacy, and the scalability. Traditional methods struggle to manage the diverse and large-scale data generated by ITS, leading to inefficiencies, high latency, and privacy concerns [3]. Centralizing sensitive transportation data increases the risk of breaches, while the volume and variety of data complicate real-time decision-making and personalized insights. Transferring large data volumes also limits bandwidth, hindering timely model training and inference [4].

FL addresses these challenges by performing computations locally on devices and sending only model updates to a central server, reducing the need to transfer raw data [5]. This decentralized approach improves privacy, security, and efficiency, making FL a valuable alternative to centralized methods. FL allows data to remain on local devices while only sharing model updates, which enhances privacy, reduces latency, and better suits the distributed nature of ITS data sources [6], [7]. Algorithms like FedAvg, Fed-SGD, and FedProx address challenges related to non-identically distributed data, making them well-suited for managing Big Data in ITS environments [8], [9].

This paper integrates FL with BDA to overcome classical BDA challenges. It effectively utilizes FL to optimize traffic management and implements FL algorithms within an integrated BDA framework to enhance public transportation systems. Additionally, we contribute by developing and validating an ITS dataset for FL environments and conducting comprehensive performance and scalability analyses of FL algorithms to improve urban transportation efficiency and accessibility. This paper also evaluates the performance of various FL algorithms in handling ITS Big Data. By curating and preparing an ITS dataset for FL for collaborative model training, this study explores the impact of different FL algorithms, including Fed-SGD, FedAvg, and FedProx.

Workshop on Security and Privacy in Standardized IoT (SDIoTSec) 2025
24 February 2025, San Diego, CA, USA
ISBN 979-8-9919276-6-6
https://dx.doi.org/10.14722/sdiotsec.2025.23053
www.ndss-symposium.org

The findings demonstrate the potential of FL in modernizing transportation systems through real-time, scalable analytics. The key contributions of this research are:

- Integration of FL with BDA to Overcome Classical BDA Challenges
- Effective Utilization of Federated Learning
- Implementation of FL Algorithms in an Integrated BDA Framework
- Development and Validation of ITS Dataset for FL Environment
- Comprehensive Performance Analysis of FL Algorithms
- Scalability Analysis

II. Related Work

The IoT has transformed many industries by enabling billions of devices to communicate and share data in real time [10]. IoT is pivotal in improving traffic management, reducing accidents, and enhancing the overall travel experience for commuters [11]. IoT-enabled ITS generates vast amounts of Big Data, encompassing vehicle telemetry, traffic patterns, passenger behaviors, and environmental conditions, which can be leveraged to optimize transportation systems [13]. BDA in ITS is essential for enabling real-time decision-making, predictive modeling, and system optimization [14]. Big Data in ITS presents numerous challenges, especially concerning privacy and data management. The data collected is often personal and location-sensitive, raising serious privacy concerns about unauthorized access or misuse. Moreover, data heterogeneity from various sources and the need for real-time processing make it challenging to use ITS Big Data [15] efficiently. Traditional methods struggle with these vast, diverse datasets, leading to inefficiencies and high latency, making them unsuitable for real-time analytics in ITS [16].

FL has emerged as a promising solution to these issues. FL enables decentralized model training, allowing data to remain on local devices while sharing only model updates with a central server, preserving privacy and reducing the need for large-scale data transfers [17]. This approach aligns well with the distributed nature of ITS data sources, reducing latency and ensuring scalability as ITS networks grow [18]. FL provides a flexible framework that helps mitigate privacy risks and computational inefficiencies while ensuring timely decision-making in ITS. Several FL algorithms have been developed to address specific challenges. FedAvg is widely used for aggregating local model updates and training global models [19]. Its weighted and unweighted versions enable nuanced model updates based on the significance of individual data sources. FedSGD simplifies the process by using a single-step update per client. FedProx, on the other hand, is designed to handle non-identically distributed data, a common issue in real-world datasets [20]. The adaptability of FedAvg with various optimizer schedules further enhances its versatility, making it suitable for large-scale, complex ITS environments [21]. These algorithms collectively provide a powerful toolkit for managing Big Data challenges in IoT-enabled ITS, ensuring both scalability and efficiency.

III. Classical BDA Framework

BDA frameworks are designed to manage the processing, analysis, and storage of large datasets. This streamlined framework supports the entire BDA process from data collection to visualization, enabling effective decision-making. The key stages are highlighted in Table I. The stages include:

- **Data Collection**: Data is gathered from various sources like IoT devices, sensors, and online platforms.
- **Data Integration**: This step merges data from multiple sources into a unified format, involving data integration, cleaning, and transformation.
- **Data Storage**: Collected data is stored efficiently using Data Lakes or Warehouses with cloud storage and Hadoop often utilized for this purpose.
- **Data Preprocessing**: This stage involves cleaning and transforming data to remove inaccuracies and prepare it for analysis.
- **Data Processing and Analysis**: This phase extracts insights from the data using techniques like descriptive, predictive, and prescriptive analytics.
- **Data Visualization**: It involves presenting the analysis through visualizations like charts, graphs, and dashboards.

The data framework begins with the Data Collection phase, where data is gathered from a variety of sources such as IoT devices, sensors, and online platforms. This phase often faces challenges in managing the volume and variety of data, ensuring that the data is of high quality and suitable for further processing. Ensuring consistency and accuracy during collection is critical, as data that is not properly gathered can lead to complications in later stages. The data moves to the Data Integration phase after collection, where it is merged from multiple sources into a unified format. This step includes integrating, cleaning, and transforming the data to ensure consistency and compatibility across different data formats and sources. Tools such as ETL (Extract, Transform, Load) and middleware solutions play a crucial role here, helping to overcome challenges like format inconsistencies, data redundancy, and quality issues. Effective data integration ensures that the data is reliable and can be processed efficiently in subsequent steps.

Data Storage is used to stored data in systems like Data Lakes or Data Warehouses. These storage solutions must be scalable to handle large datasets and secure to protect sensitive information. Cloud storage solutions and distributed storage systems like Hadoop are frequently used to meet these demands. Proper storage infrastructure enables efficient access to data, allowing for smooth data processing. In the Data Preprocessing stage, the stored data is cleaned and transformed to prepare it for analysis. This involves handling missing or incomplete data, removing inaccuracies, and converting the data into a usable format. Python and R are standard tools used during this phase to automate and streamline data-cleaning tasks.

In Processing and Analysis, insights are extracted from the data using various analytical techniques. Analyzing large datasets requires specialized skills to apply these algorithms effectively, making this stage one of the most critical for deriving actionable insights. Analysts must choose the proper methods and algorithms to interpret the data accurately. The final stage is Data Visualization, where the insights gained from the analysis are presented in a visually accessible format, such as charts, graphs, and dashboards. This is essential in communicating findings to non-technical stakeholders who may need to become more familiar with complex data analysis. Effective data visualization turns raw data into a story that can guide business strategies and actions.

A. Classical BDA Challenges

Big Data presents technical, ethical, and organizational challenges. Ensuring data quality is a primary hurdle, as the unstructured and diverse data from multiple sources often contains inaccuracies, making analysis difficult and leading to potential misinformed decisions [22]. Data integration across various formats and systems is essential yet challenging for accurate insights. Despite reduced storage expenses, managing costs remains significant when considering the entire infrastructure required [23]. Technological advancements necessitate organizations to rapidly adapt to new tools and methods for processing growing data volumes. Privacy is a significant concern, especially with the risks of data breaches in centralized systems [24]. Bias and discrimination are also concerns, as unrepresentative data can lead to skewed analytics, impacting decisions in areas like hiring or law enforcement.

In ITS, the complexity of managing large-scale data and ensuring the privacy of sensitive travel information is a notable challenge. Processing this data in real-time demands significant infrastructure investment. Moreover, protecting individual privacy becomes even more critical as ITS expands, particularly with the advent of connected and autonomous vehicles, contributing to smart cities and sustainable urban mobility. Federated Learning (FL) offers a promising solution to many challenges by decentralizing data processing, addressing privacy concerns, and improving scalability, latency, and bandwidth use. FL allows models to be trained on diverse datasets without compromising data integrity, fostering a secure and efficient Big Data ecosystem that supports innovation in analytics. The key challenges are highlighted in Table II. Key challenges include:

- **Privacy Concerns**: Centralized data raises risks of breaches, especially in sectors like ITS.
- **High Latency**: Centralized processing leads to delays in real-time applications, such as ITS.
- **Limited Scalability**: Scaling machine learning across many devices is complex without FL.
- **Lack of Personalization**: Traditional centralized models can compromise privacy in personalized applications.
- **Homogeneity of Data Sources**: Centralized approaches may fail to handle diverse data sources effectively.
- **Bandwidth Constraints**: Transferring large datasets without FL consumes significant bandwidth.
- **Delayed Real-time Learning**: Centralized models can slow down dynamic decision-making.
- **Inefficient Resource Utilization**: Without FL, computational resources are often underutilized.
- **Limited Data Inclusivity**: Traditional models may miss data from remote devices, affecting model performance.
- **Stagnation in Innovation**: Without collaborative frameworks like FL, innovation in Big Data analytics can be limited.

IV. Federated Learning for BDA in ITS

The integration of FL with IoT-enabled ITS addresses key Big Data challenges in these systems [25], as shown in Figure 1. Due to bandwidth limitations and privacy concerns, the immense volume and variety of data generated by IoT devices and ITS make traditional centralized data processing impractical. IoT devices and ITS infrastructure, like traffic sensors and vehicular systems, produce continuous data streams. FL offers a solution by enabling on-device learning, where local models are trained, and only aggregated model updates are shared with a central server, reducing data transmission and preserving privacy [26]. FL can facilitate real-time traffic management in ITS by processing data from various local traffic sources, allowing timely predictions and adjustments. Similarly, in IoT, FL enables devices to adapt in real-time while minimizing large-scale data transfers and protecting user privacy [27]. As IoT and ITS ecosystems expand, FL will be crucial in improving efficiency and addressing privacy concerns [28].

This study explores the use of large-scale data in ITS, focusing on privacy protection and scalability. With increasing privacy concerns and the complexity of Big Data, FL emerges as a practical solution [29]. Our research evaluates the performance of various FL algorithms applied to ITS data, aiming to balance privacy, efficiency, and scalability. Using a dataset of 15,000 high-quality images from Udacity and Roboflow optimized for the YOLO object detection system, we conduct experiments in collaborative learning with transportation imagery [30]. The study highlights the effectiveness of FL algorithms in processing ITS Big Data through collaborative learning rounds, showing promising scalability. We assess algorithms such as Fed-SGD, FedAvg, and FedProx, examining their performance with different optimizations. The study provides insights into how FL can be applied to ITS environments, focusing on privacy and real-time processing needs.

The main contributions include:

- **Big Data Preparation for FL:** Curated and tailored an ITS dataset specifically for FL experiments.
- **Decentralized Data Processing:** Demonstrated the effectiveness of FL for privacy-preserving, decentralized data processing.
- **Collaborative Learning:** Showed promising scalability and adaptability of FL across communication rounds.

TABLE I
CLASSICAL BDA FRAMEWORK

Stage	Description
Data Collection	Data is gathered from various sources like IoT devices, sensors, and online platforms. The main challenge is managing the volume and variety of data while ensuring quality.
Data Integration	This step merges data from multiple sources into a unified format, involving data integration, cleaning, and transformation.
Data Storage	Collected data is stored efficiently using Data Lakes or Warehouses. Challenges include ensuring scalability, security, and integrity.
Data Preprocessing	This stage involves cleaning and transforming data to remove inaccuracies and prepare it for analysis. Python and R are commonly used to handle missing or incomplete data.
Data Processing and Analysis	This phase extracts insights from the data using techniques like descriptive, predictive, and prescriptive analytics and machine learning.
Data Visualization	The final step involves presenting the analysis through visualizations like charts, graphs, and dashboards, making insights accessible to non-technical stakeholders.

TABLE II
CLASSICAL BDA CHALLENGES

Concern	Description
Privacy Concerns	Centralized data raises risks of breaches, especially in sectors like ITS.
High Latency	Centralized processing leads to delays in real-time applications, such as ITS.
Limited Scalability	Scaling machine learning across many devices is complex without Federated Learning (FL).
Lack of Personalization	Traditional centralized models can compromise privacy in personalized applications.
Homogeneity of Data Sources	Centralized approaches may fail to handle diverse data sources effectively.
Bandwidth Constraints	Transferring large datasets without FL consumes significant bandwidth.
Delayed Real-time Learning	Centralized models can slow down dynamic decision-making.
Inefficient Resource Utilization	Without FL, computational resources are often underutilized.
Limited Data Inclusivity	Traditional models may miss data from remote devices, affecting model performance.
Stagnation in Innovation	Without collaborative frameworks like FL, innovation in Big Data analytics can be limited.

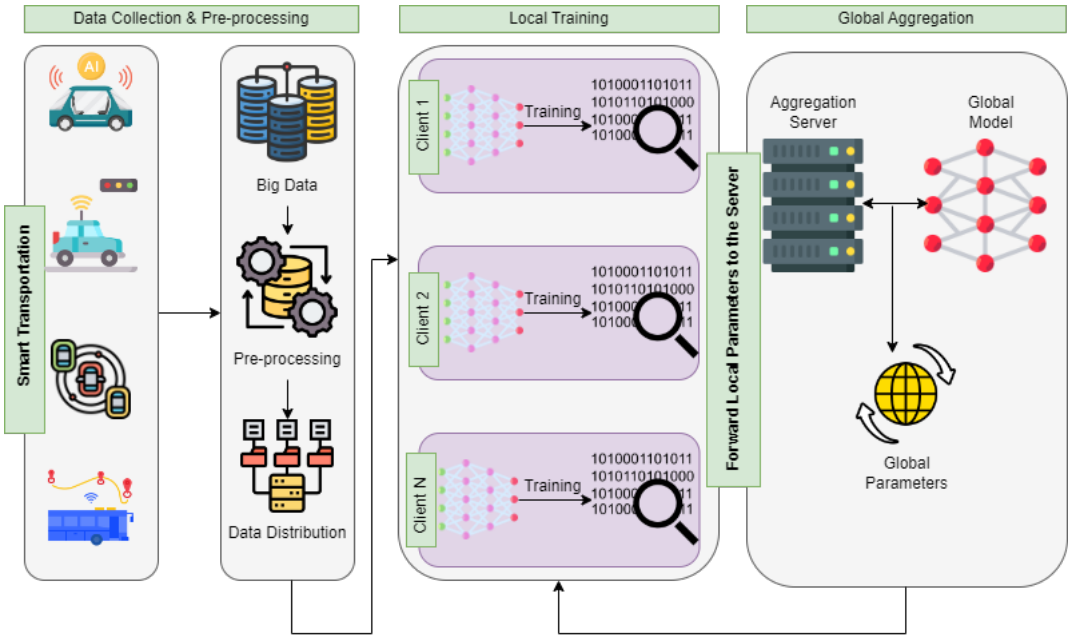

Fig. 1. Federated Learning on Big Data for Intelligent Transportation Systems

- **Exploration of FL Algorithms:** Investigated various FL algorithms, including Fed-SGD, FedAvg, and FedProx, with different optimization techniques.
- **Evaluation of Collaborative Learning:** Provided an evaluation of learning processes, examining the impact of collaborators and epochs on FL performance in ITS.

V. METHODOLOGY

In this study, each device initially trains models locally using Deep Learning techniques, including Multi-Layer Perceptrons (MLP) and Convolutional Neural Networks (CNN). CNNs were particularly effective in identifying complex patterns in the ITS dataset. The central server then aggregates

the locally trained model parameters using Federated Learning (FL) algorithms like Fed-SGD, FedAvg, and FedProx. These aggregated updates are returned to the devices for further training rounds, ensuring continuous model improvement. The workflow is depicted in Figure 2.

A. Data Preparation for FL

We prepared the ITS dataset by organizing and partitioning it for efficient use in FL. Meta-data files were generated to map the vast data, and the dataset was divided into "shards" to simulate decentralized environments. This structure allowed efficient data distribution and training across client devices, ensuring balanced data access and consistent classification results.

B. Client Setup for Collaborative Learning

We established multiple virtual client nodes, each representing a real-world data distribution. Data was allocated to nodes using an adaptive algorithm to ensure diversity. The clients processed the data in parallel, mimicking real-world FL setups. Post-training, model updates were aggregated, refining the overall global model.

Algorithm 1 Federated Learning with Multiple Clients

Input: Number of clients K, initial global model W_0, total communication rounds T, dataset $\mathcal{D}$
Output: Final global model W_T
Setup:
Prepare the dataset $\mathcal{D}$ and corresponding meta-information for FL
Divide $\mathcal{D}$ into K distinct subsets, one for each client
Initialize the global model W_0
each round $t = 1, \ldots, T$
Client-Side Local Training: each client $k = 1, \ldots, K$ **(in parallel)**
Receive the global model W_{t-1}
Train local model W_k^t on client k's data using MLP/CNN
Send local model update ΔW_k^t to the central server
Server-Side Model Aggregation:
Aggregate the local updates ΔW_k^t using a method such as FedAvg or FedProx to update the global model W_t
Distribute the updated global model W_t to all clients
Return: Final global model W_T =0

C. Training with Deep Learning Models

Both MLPs and CNNs were employed for training across client nodes. MLPs served as foundational models, while CNNs were used for more complex image recognition tasks. CNNs, with their convolutional and pooling layers, excelled at extracting features from transportation imagery, and dropout layers helped mitigate overfitting. This approach allowed for practical evaluation of the FL process in real-world scenarios.

D. Global Aggregation Using Federated Learning Algorithms

In Federated Learning (FL), global model aggregation combines updates from multiple clients to create a unified global model. This process aims to represent the data distribution across clients while accounting for the variations in their data, ensuring balanced training. Various techniques, from basic to advanced, are used to achieve this. Simple aggregation treats all client updates equally, while weighted averaging gives more significance to updates based on the importance of each device's data and scales them accordingly. A more refined method, such as Fed aggregation, adjusts the weights based on the accuracy of local updates and the data volume at each node. The global aggregation process using FL algorithms is as follows:

1) A central server initializes a global model.
2) The model is distributed to a subset of devices or nodes for local training.
3) Each device updates the model using its local data.
4) The updated local models are sent back to the server.
5) The server averages these updates.
6) The global model is then updated with the aggregated data.
7) Steps 2-6 are repeated iteratively over several rounds.

VI. Results and Discussion

Our experiments utilized the Udacity Self Driving Car Dataset [29], [30], containing 15,000 high-resolution images annotated across 11 categories, formatted for YOLO object detection. This dataset, shared by Roboflow, is widely recognized in ITS research due to its comprehensiveness and precision, making it an ideal choice for vehicle recognition in our federated learning experiments. Our experiments were conducted on an Intel Core i7 processor, 32 GB RAM, and TensorFlow Federated (TFF) as the main framework for implementing FL algorithms. The Kaggle API facilitated smooth data integration into our Google Colab environment, with Python code running in Jupyter Notebooks.

Analyzing the performance of three FL algorithms, including FedAvg, FedSGD, and FedProx, across 40 epochs in a 10-node simulation environment reveals distinct learning behaviors and efficiencies. The 10-node environment is preferred due to resource limitations. It can be increased to more nodes if we have more resources. The experiments in this paper used the Udacity Self-Driving Car dataset, valuable for controlled testing of FL algorithms. However, real-world ITS testing introduces complexities, including data diversity, large-scale deployment, and practical constraints. Real-world ITS data is more diverse, involving various vehicle types, infrastructure, and environmental conditions, requiring datasets from multiple sources. Large-scale deployments in ITS systems, with thousands of nodes, may cause latency and bandwidth issues.

The comparative analysis is done in the context of global accuracy, achieved by the global aggregation of all the nodes. Initially, FedAvg has the highest global accuracy in the early epochs, as shown in Figure 3. For example, at epoch 1,

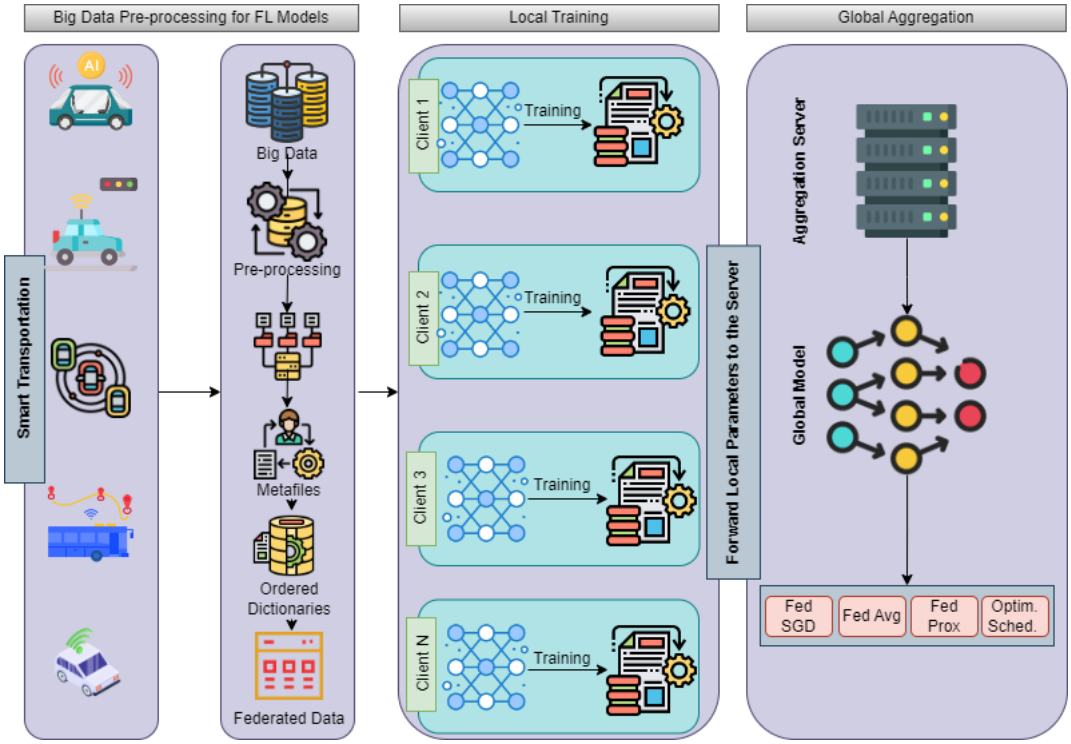

Fig. 2. Methodology

FedAvg achieves an accuracy of 85.46%, surpassing both FedSGD (83.74%) and FedProx (81.60%). This suggests that FedAvg quickly adapts to the dataset, which is crucial in FL applications requiring rapid initial learning and adaptation across distributed nodes. FedSGD shows steady improvement as training progresses, gradually closing the accuracy gap with FedAvg. By epoch 3, FedSGD reaches an accuracy of 86.82%, slightly outperforming FedAvg's 86.27%, highlighting its capacity for consistent learning over time in a federated environment.

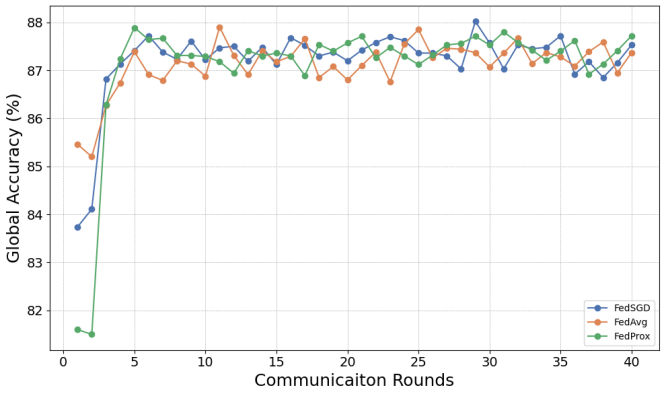

Fig. 3. Performance of FL Algorithms using 10 Nodes.

Despite starting with lower accuracy, FedProx outperforms FedAvg and FedSGD in the later epochs. By epoch 5, FedProx reaches an accuracy of 87.89%, the highest among the three algorithms. This improvement can be attributed to FedProx's design, which is particularly effective at handling non-IID data. A key observation from the loss data is FedProx's initially high loss value of 0.814 at epoch 1, significantly higher than FedSGD's 0.517 and FedAvg's 0.514, as shown in Figure 4. This aligns with its lower initial accuracy, suggesting that FedProx requires more iterations to adapt effectively in the early stages. However, FedProx significantly improves as training progresses, reducing its loss to 0.495 by epoch 5. Despite early challenges, FedProx is highly effective at optimizing its learning process.

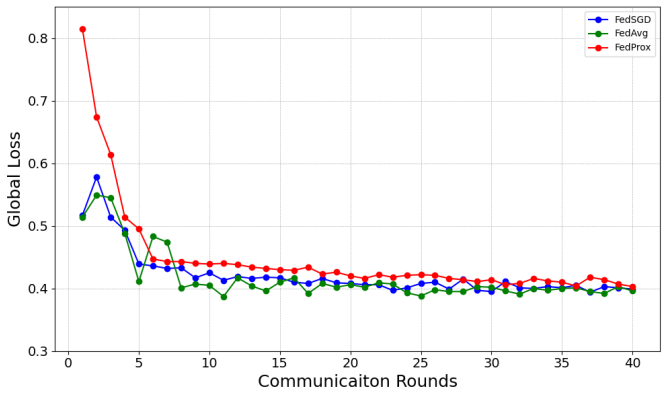

Fig. 4. Loss using 10 Nodes.

In contrast, FedAvg and FedSGD start with relatively lower loss values, with FedAvg maintaining a slight advantage over

FedSGD. For example, at epoch 5, FedAvg records a loss of 0.411 compared to FedSGD's 0.439, further supporting its higher early accuracy. While FedAvg performs well in the initial epochs, combining these insights with the accuracy data reveals that FedProx's ability to overcome its early limitations and improve significantly makes it a strong contender, especially in scenarios that benefit from extended training. FedSGD, though consistent, does not reach the optimization level of the other two algorithms, making it less ideal in settings where rapid adaptability and efficiency are crucial.

The results of our experiments highlight the performance of three FL algorithms: FedAvg, FedSGD, and FedProx. FedProx outperforms the others in later epochs, offering superior accuracy due to its proximal term stabilizing the learning process. This makes it ideal for heterogeneous data in ITS, where data from various vehicles and infrastructures differ. FedAvg, however, excels in early epochs, quickly adapting to emerging traffic patterns, making it suitable for real-time ITS scenarios. FedSGD shows steady improvement over time, offering consistency, which is useful in long-term traffic management and large, diverse datasets. A hybrid approach combining FedAvg's early adaptation and FedProx's stable convergence could benefit dynamic traffic conditions. In conclusion, the choice of FL algorithm depends on specific ITS needs—whether quick adaptation, stability, or high accuracy is required. These insights will guide practitioners in selecting the right algorithm for balancing immediate response and long-term system stability in ITS deployments.

Scalability is a critical challenge for implementing FL in large-scale ITS. As the number of nodes and dataset size grows, issues like cost, computational resources, and latency emerge. High infrastructure costs can be reduced using edge computing and model compression, where data is processed locally and model updates are compressed. FL can address computational limitations using lightweight models or offloading tasks to more powerful edge devices or the cloud. Latency and bandwidth problems in real-time ITS can be mitigated with a hybrid approach, combining local training with central computation. Adaptive learning rates can prioritize faster devices in early training to manage load imbalances and scale work to slower devices later. These strategies can help effectively scale FL in real-world ITS systems.

VII. Comparative Analysis

Table III highlights the advantages of the proposed BDA framework integrated with FL compared to traditional BDA methods. The proposed framework offers a distributed approach that significantly enhances privacy by keeping data on local devices, reducing the risk of central data breaches. Additionally, it lowers latency by processing data locally, making it ideal for real-time applications. In contrast, traditional systems face higher latency and privacy risks due to data centralization. Another key advantage of the proposed framework is its scalability, which allows it to handle increasing data volumes across many devices without centralizing data.

This proposed distributed approach enables efficient bandwidth usage, as only model updates, not raw data, are transmitted. Traditional systems, in comparison, struggle with scaling and bandwidth efficiency due to the need to transfer large datasets. The proposed framework also excels in personalization, allowing tailored applications while maintaining privacy, a challenge for traditional methods. It draws from diverse data sources, making models more robust than conventional systems that rely on homogeneous data. Local processing enables real-time learning, providing faster insights critical for dynamic environments. Furthermore, the proposed system optimizes computational resource use and ensures inclusivity by integrating data from various devices, even remote areas. By fostering innovation through secure collaboration, the proposed framework addresses the limitations of traditional BDA, offering a more efficient and scalable solution for modern data analytics.

VIII. Conclusion

This study underscores the transformative role of FL in addressing the challenges of managing vast, decentralized data in ITS. By integrating FL with BDA, privacy is preserved as the data resides on the nodes, and only parameters are passed for global training purposes using FL. Using the Udacity Self-Driving Car Dataset, we highlighted the efficiency of FL in optimizing data across diverse sources without sacrificing user privacy or system performance. Our results suggest that FL can become a cornerstone of future ITS architectures, particularly in environments that demand collaboration and data security. However, real-world deployments will inevitably introduce new challenges. While our framework performed well in a controlled, simulated environment, scaling FL to real-world ITS scenarios requires further exploration, especially under varying network conditions and hardware limitations. Future research should focus on testing with larger, more heterogeneous datasets, exploring novel FL algorithms, and extending the integration to edge computing and emerging technologies such as 6G. These advancements could unlock even greater potential for FL, driving innovations in traffic management, autonomous vehicles, and urban mobility systems.

Acknowledgments

This work was supported by the research grant [SEED-CCIS-2024-166]; Prince Sultan University, Riyadh, Saudi Arabia. The authors would like to thank Prince Sultan University for their support.

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (Grant Number: IMSIU-RP23008).

This work was supported by the faculty research funds of the University of Illinois at Springfield, USA.

References

[1] Yalli, Jameel S., Mohd H. Hasan, and Aisha Badawi. "Internet of things (iot): Origin, embedded technologies, smart applications and its growth in the last decade." IEEE Access (2024).

TABLE III
COMPARATIVE ANALYSIS.

Aspect	Proposed BDA	Traditional BDA
Privacy	Enhanced due to decentralized approach. Data stays on local devices, reducing central data breach risks.	Higher risk as data is centralized, increasing breach risks.
Latency	Reduced, as data is processed locally. Ideal for real-time applications.	Higher, due to central processing of data.
Scalability	High, as models can be scaled across many devices without centralizing data.	Limited, as scaling involves centralizing large data volumes.
Personalization	Facilitated while maintaining privacy, suitable for tailored applications.	Challenging to achieve without compromising privacy.
Data Source Diversity	Improved model robustness from diverse datasets across devices.	Models may lack generalizability due to homogeneous data sources.
Bandwidth Efficiency	Higher, as only model parameters or gradients are communicated, not raw data.	Lower, due to the transfer of large data volumes.
Real-time Learning	Enabled, crucial for dynamic environments where timely insights are vital.	Slower, affecting decision-making in dynamic environments.
Resource Optimization	Efficient utilization of computational resources across devices.	Potential underutilization of resources and higher costs.
Data Inclusivity	Higher, as it includes data from a vast network of devices, even from remote areas.	Limited, potentially excluding data from less accessible areas.
Innovation	Encourages collaborative model training, fostering innovation without sharing sensitive data.	May hinder innovation due to reluctance in data sharing.

[2] Kaleem, Sarah, Adnan Sohail, Muhammad Usman Tariq, and Muhammad Asim. "An improved big data analytics architecture using federated learning for IoT-enabled urban intelligent transportation systems." Sustainability 15, no. 21 (2023): 15333.

[3] Aouedi, Ons, Thai-Hoc Vu, Alessio Sacco, Dinh C. Nguyen, Kandaraj Piamrat, Guido Marchetto, and Quoc-Viet Pham. "A survey on intelligent Internet of Things: applications, security, privacy, and future directions." IEEE Communications Surveys Tutorials (2024).

[4] Babar, Muhammad, Basit Qureshi, and Anis Koubaa. "Review on Federated Learning for digital transformation in healthcare through big data analytics." Future Generation Computer Systems (2024).

[5] Zhang, Yifei, Dun Zeng, Jinglong Luo, Xinyu Fu, Guanzhong Chen, Zenglin Xu, and Irwin King. "A Survey of Trustworthy Federated Learning: Issues, Solutions, and Challenges." ACM Transactions on Intelligent Systems and Technology (2024).

[6] Sánchez, Pedro Miguel Sánchez, Alberto Huertas Celdrán, Ning Xie, Gérôme Bovet, Gregorio Martínez Pérez, and Burkhard Stiller. "Federatedtrust: A solution for trustworthy federated learning." Future Generation Computer Systems 152 (2024): 83-98.

[7] Chai, Di, Leye Wang, Liu Yang, Junxue Zhang, Kai Chen, and Qiang Yang. "A survey for federated learning evaluations: Goals and measures." IEEE Transactions on Knowledge and Data Engineering (2024).

[8] Huang, Wenke, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. "Federated learning for generalization, robustness, fairness: A survey and benchmark." IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[9] Zhang, Rongqing, Jingxin Mao, Hanqiu Wang, Bing Li, Xiang Cheng, and Liuqing Yang. "A survey on federated learning in intelligent transportation systems." IEEE Transactions on Intelligent Vehicles (2024).

[10] Kaleem, Sarah, Adnan Sohail, Muhammad Babar, Awais Ahmad, and Muhammad Usman Tariq. "A hybrid model for energy-efficient Green Internet of Things enabled intelligent transportation systems using federated learning." Internet of Things 25 (2024): 101038.

[11] Zhang, Shiying, Jun Li, Long Shi, Ming Ding, Dinh C. Nguyen, Wuzheng Tan, Jian Weng, and Zhu Han. "Federated learning in intelligent transportation systems: Recent applications and open problems." IEEE Transactions on Intelligent Transportation Systems (2023).

[12] Anbukkarasi, S., and C. R. Dhivyaa. "AI Techniques for Future Smart Transportation." In Artificial Intelligence for Future Intelligent Transportation, pp. 243-268. Apple Academic Press, 2024.

[13] Laraib, Areeba, and Raja Majid Ali Ujjan. "Intelligent Transportation Systems (ITS) Opportunities and Security Challenges." Cybersecurity in the Transportation Industry (2024): 117-141.

[14] Zhu, L., Yu, F. R., Wang, Y., Ning, B., & Tang, T. (2018). Big data analytics in intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 20(1), 383-398.

[15] Shoman, Wasim, Sonia Yeh, Frances Sprei, Jonathan Köhler, Patrick Plötz, Yancho Todorov, Seppo Rantala, and Daniel Speth. "A review of big data in road freight transport modeling: gaps and potentials." Data Science for Transportation 5, no. 1 (2023).

[16] Gadekallu, Thippa Reddy, Quoc-Viet Pham, Thien Huynh-The, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Madhusanka Liyanage. "Federated learning for big data: A survey on opportunities, applications, and future directions." arXiv preprint arXiv:2110.04160 (2021).

[17] Babar, Muhammad, Basit Qureshi, and Anis Koubaa. "Investigating the impact of data heterogeneity on the performance of federated learning algorithm using medical imaging." Plos one 19, no. 5 (2024): e0302539.

[18] Li, Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. "A review of applications in federated learning." Computers & Industrial Engineering 149 (2020): 106854.

[19] Su, Lili, Jiaming Xu, and Pengkun Yang. "A non-parametric view of FedAvg and FedProx: beyond stationary points." Journal of Machine Learning Research 24, no. 203 (2023): 1-48.

[20] Yuan, Xiaotong, and Ping Li. "On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond." Advances in Neural Information Processing Systems 35 (2022): 10752-10765.

[21] Chen, Junbin, Jipu Li, Ruyi Huang, Ke Yue, Zhuyun Chen, and Weihua Li. "Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging." IEEE Transactions on Instrumentation and Measurement 71 (2022): 1-11.

[22] Prakash, Andrea, Narem Navya, and Jayapandian Natarajan. "Big data preprocessing for modern world: opportunities and challenges." In International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018, pp. 335-343. Springer International Publishing, 2019.

[23] Srivastava, Gautam, Christoph M. Flath, Jerry Chun-Wei Lin, and Yu-Dong Zhang. "Challenges and Outcomes Using Big Data as a Service." Business and Information Systems Engineering 66, no. 1 (2024): 1-2.

[24] Demirol, Doygun, Resul Das, and Davut Hanbay. "A key review on security and privacy of big data: issues, challenges, and future research directions." Signal, Image and Video Processing 17, no. 4 (2023): 1335-1343.

[25] Nguyen, Hoang Phuong, Phuoc Quy Phong Nguyen, and Viet Duc Bui. "Applications of big data analytics in traffic management in intelligent transportation systems." JOIV: International Journal on Informatics Visualization 6, no. 1-2 (2022): 177-187.

[26] Zhao, Jianxin, Xinyu Chang, Yanhao Feng, Chi Harold Liu, and Ningbo Liu. "Participant selection for federated learning with heterogeneous data in intelligent transport system." IEEE transactions on intelligent transportation systems 24, no. 1 (2022): 1106-1115.

[27] Posner, Jason, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh.

”Federated learning in vehicular networks: Opportunities and solutions.” IEEE Network 35, no. 2 (2021): 152-159.
[28] Wang, Xiaoding, Wenxin Liu, Hui Lin, Jia Hu, Kuljeet Kaur, and M. Shamim Hossain. ”AI-empowered trajectory anomaly detection for intelligent transportation systems: A hierarchical federated learning approach.” IEEE Transactions on Intelligent Transportation Systems 24, no. 4 (2022): 4631-4640.
[29] sshikamaru, “Udacity self-driving car dataset,” https://www.kaggle.com/datasets/sshikamaru/udacity-self-driving-car-dataset, 2023, accessed: 2023-11-06.
[30] https://public.roboflow.com/object-detection/self-driving-car