




Article

An Improved Big Data Analytics Architecture Using Federated Learning for IoT-Enabled Urban Intelligent Transportation Systems

Sarah Kaleem ^{1,*} , Adnan Sohail ¹, Muhammad Usman Tariq ²  and Muhammad Asim ³ 

¹ Department of Computing and Technology, Iqra University, Islamabad 44000, Pakistan; adnan.sohail@iqraisb.edu.pk

² Department of Marketing, Operations, and Information Systems, Abu Dhabi University, Abu Dhabi P.O. Box 59911, United Arab Emirates; muhammad.kazi@adu.ac.ae

³ EIAS Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; masim@psu.edu.sa

* Correspondence: sarahkaleem33887@iqraisb.edu.pk

Abstract: The exponential growth of the Internet of Things has precipitated a revolution in Intelligent Transportation Systems, notably in urban environments. An ITS leverages advancements in communication technologies and data analytics to enhance the efficiency and intelligence of transport networks. At the same time, these IoT-enabled ITSs generate a vast array of complex data classified as Big Data. Traditional data analytics frameworks need help to efficiently process these Big Data due to its sheer volume, velocity, variety, and significant data privacy concerns. Federated Learning, known for its privacy-preserving attributes, is a promising technology for implementation within ITSs for IoT-generated Big Data. Nevertheless, the system faces challenges due to the variable nature of devices, the heterogeneity of data, and the dynamic conditions in which ITS operates. Recent efforts to mitigate these challenges focus on the practical selection of an averaging mechanism during the server's aggregation phase and practical dynamic client training. Despite these efforts, existing research still relies on personalized FL with personalized averaging and client training. This paper presents a personalized architecture, including an optimized Federated Averaging strategy that leverages FL for efficient and real-time Big Data analytics in IoT-enabled ITSs. Various personalization methods are applied to enhance the traditional averaging algorithm. Local fine-tuning and weighted averaging tailor the global model to individual client data. Custom learning rates are utilized to boost the performance further. Regular evaluations are advised to maintain model efficacy. The proposed architecture addresses critical challenges like real-life federated environment settings, data integration, and significant data privacy, offering a comprehensive solution for modern urban transportation systems using Big Data. Using the Udacity Self-Driving Car Dataset for vehicle detection, we apply the proposed approaches to demonstrate the efficacy of our model. Our empirical findings validate the superiority of our architecture in terms of scalability, real-time decision-making capabilities, and data privacy preservation. We attained accuracy levels of 93.27%, 92.89%, and 92.96% for our proposed model in a Federated Learning architecture with 10 nodes, 20 nodes, and 30 nodes, respectively.

Keywords: big data analytics; federated learning; internet of things; smart transportation; intelligent transportation systems



Citation: Kaleem, S.; Sohail, A.; Tariq, M.U.; Asim, M. An Improved Big Data Analytics Architecture Using Federated Learning for IoT-Enabled Urban Intelligent Transportation Systems. *Sustainability* **2023**, *15*, 15333. <https://doi.org/10.3390/su152115333>

Academic Editors: Eduardo Mojica-Nava and Juan Manuel Rey

Received: 26 September 2023

Revised: 14 October 2023

Accepted: 25 October 2023

Published: 26 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things has become a cornerstone in the evolution of a digitally connected world, enabling various sectors to collect and analyze data in real time [1]. By embedding sensors and software in physical objects, IoT technologies allow for unprecedented levels of monitoring and automation, paving the way for more innovative and

efficient systems [2]. One of the most impactful applications of IoT is in the domain of Intelligent Transportation Systems. An IoT-enabled ITS aims to optimize traffic flow, improve road safety, and enhance the overall transportation experience for individuals and logistics providers [3] through interconnected sensors, vehicles, and traffic management tools. These systems are becoming particularly crucial in urban environments, where managing complex, congested networks is a growing challenge [4]. One of the most formidable challenges and opportunities posed by IoT-enabled Intelligent Transportation Systems is generating voluminous and highly complex data, often called Big Data [5]. These systems employ interconnected sensors, vehicles, traffic lights, and other IoT devices that continuously collect and transmit real-time data. The data can range from vehicle speed and location to weather conditions, road quality, and driver behavior [6]. The diversity of data types, including structured, semi-structured, and unstructured data, adds another layer of complexity.

The data is generated at an unprecedented velocity, requiring rapid processing for actionable insights. Given the velocity, volume, and variety, which are the three Vs of Big Data, it becomes evident that traditional data processing systems must be equipped to handle the complexities of data flow and analytics in an IoT-enabled ITS [7]. This enormous scale and complexity of data not only necessitate more advanced Big Data analytics but also makes it imperative to address challenges related to data storage, privacy, integration, and real-time processing [8]. IoT-enabled ITSs inherently generate colossal amounts of data due to the continuous real-time collection and transmission of various types of information [9]. This ever-growing mountain of data falls under the category of Big Data, characterized by its high velocity, volume, and variety [10]. While Big Data provides opportunities for deep analytics and insights, it also presents many challenges. Traditional data processing frameworks often need to be revised to handle this data's sheer scale and complexity, which requires real-time analysis for actionable insights [11]. Furthermore, integrating disparate data types and sources adds a layer of complication to the analytical processes. One of the most pressing challenges in dealing with Big Data from IoT-enabled ITSs is the issue of data privacy.

Since individual vehicles and devices contribute sensitive information, these data's centralized collection and processing raise significant privacy and security concerns [12]. Federated Learning (FL) offers a novel approach to tackling data privacy challenges. In a federated model, machine learning algorithms are trained across multiple decentralized devices or servers holding local data samples without exchanging them [13]. This allows for practical model training and ensures that the data remain on the local device, thereby maintaining individual privacy. FL is a machine learning paradigm where multiple decentralized devices or servers collaboratively train a shared model while keeping their data locally stored. Unlike the traditional centralized machine learning approach, FL ensures data privacy by transmitting only model updates, rather than raw data, between participating entities. This decentralized approach addresses significant privacy and security concerns, especially in domains with sensitive data. Its core benefit lies in enabling machine learning on edge devices, preserving data ownership, and minimizing data transmission overheads.

Big Data analytics using FL offers a transformative approach that addresses critical challenges like data privacy, scalability, and real-time analysis. By allowing machine learning models to be trained across multiple decentralized devices or servers, FL eliminates the need to move data to a central location. This ensures privacy compliance, optimizes resource usage, and enhances model robustness. Moreover, FL can handle real-time data analytics, non-IID data distributions, and data imbalance and heterogeneity, making it a promising solution for future Big Data analytics [14]. FL emerges as a potent solution for handling Big Data analytics in the context of Big Data produced in the IoT-enabled ITS environment. ITS produces a wealth of data from various IoT sensors embedded in the transportation infrastructure and vehicles. FL offers a decentralized approach to model

training, allowing these devices to perform localized analytics without sending sensitive or voluminous data to a central server [15].

This addresses data privacy concerns and adds a layer of efficiency and real-time responsiveness crucial for transportation systems. The localized analytics provided by FL can lead to more accurate and personalized models, better traffic management, and enhanced safety measures, positioning FL as a critical enabler for more innovative and more secure IoT-based ITSs. One well-known constraint in Federated Learning is the network bandwidth that limits the rate at which local updates from different organizations can be combined in the cloud. To mitigate this, Fedavg uses local data for gradient descent optimization before conducting a weighted average aggregation of the models uploaded by each client. The algorithm proceeds iteratively, updating the global model in each training round based on the contributions from participating organizations. Given the challenges mentioned earlier and the opportunities, this paper proposes an improved Big Data analytics architecture incorporating FL for IoT-enabled ITS. Using FL, our architecture aims to provide robust, real-time analytics while preserving user privacy. This approach will lead to more efficient data management in ITS, providing a scalable and effective solution for modern urban transportation systems. The contributions of this work include:

1. We introduce a Big Data analytics architecture that synergizes FL and IoT-enabled ITS to address critical issues such as data integration, data processing, and privacy, offering comprehensive solutions within the architecture.
2. Diverging from conventional Federated Averaging techniques, we introduce a more personalized algorithm.
3. Various personalization methods are introduced to enhance the FedAvg algorithm, including local fine-tuning and weighted averaging to tailor the global model to individual client data; custom learning rates are utilized to boost the performance further, and regular evaluations are advised to maintain model efficacy.
4. In personalized Federated Averaging, individual contributions from clients are weighted based on their data volume to utilize the Big Data features and model performance. We improve the FedProx with FedAvg, which is used for robust aggregation, accounting for system heterogeneity and stragglers. We deploy advanced adaptive aggregation techniques that factor in the attributes of client updates for a better-informed global update.
5. We execute a broad range of tests using real-world data to prove the efficacy of our suggested strategies.

The remainder of this paper is structured as follows: Section 2 reviews related works in ITS, Big Data analytics, and Federated Learning. Section 3 describes the proposed architecture. Section 4 presents our empirical findings. Section 5 provides a discussion on findings, and Section 6 concludes the paper while providing directions for future research.

2. Literature Review

Big Data analytics involves examining, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision making. With the increasing importance of data privacy and distributed data sources, FL is emerging as a powerful tool that complements traditional Big Data analytics. Utilizing FL techniques in Big Data analytics allows for decentralized model training across a myriad of data sources without the need for central data aggregation. This provides an efficient and privacy-preserving mechanism for harnessing insights from vast amounts of data scattered across multiple locations or organizations. Unlike traditional machine learning, FL enables model training across multiple decentralized nodes without requiring raw data to be shared centrally, thus ensuring data privacy and reducing data movement [16]. One of the main challenges in Big Data analytics is data privacy. FL stands out as a privacy-preserving method since it enables model training without requiring raw data to be transferred to a central server, aligning with privacy regulations like GDPR and HIPAA [17]. Big Data is often characterized by its enormous volume and the speed at which it is generated. The

scalability of FL allows it to handle the challenges of Big Data efficiently by facilitating decentralized training across multiple nodes.

The architecture of FL enables real-time data analytics as data is analyzed at the source, and no latency is involved in sending the data to a centralized location for processing. This feature is crucial for applications requiring immediate insights [18]. Traditional Big Data analytics often requires the assumption that data are independently and identically distributed (IID). FL can handle non-IID data distributions, enabling more personalized and accurate model training. Data transfer over the network is resource-intensive [19]. FL alleviates this issue by localizing the data and reducing the need to send data over the network. Instead, model updates are the only information exchanged, conserving computational resources [20]. In Big Data analytics, one of the goals is to generalize findings across diverse and complex datasets. FL contributes to model robustness by aggregating learning from diverse data sources. The issue of imbalanced and heterogeneous data is also present in Big Data analytics. FL can adapt to these challenges due to its flexible and distributed architecture. Federated Learning presents a promising avenue for tackling the challenges of Big Data analytics, offering solutions for data privacy, scalability, real-time analysis, and more [21]. Its features complement the goals of Big Data analytics, paving the way for more secure and efficient data analysis techniques.

The exponential growth of the IoT has precipitated a revolution in ITS, notably in urban environments. IoT has been a driving force behind significant advancements in ITS, especially within urban settings. ITS leverages advancements in communication technologies and data analytics to enhance the efficiency and intelligence of transport networks. This fusion aims to elevate the intelligence and efficiency of transportation networks, making them more responsive to the needs of modern urban environments. One of the groundbreaking integrations in ITS is the incorporation of FL. By leveraging FL, ITS can enable vehicles and transportation infrastructure to engage in collaborative learning. This collaboration is pivotal in optimizing traffic flow, enhancing safety protocols, and improving the overall efficiency of travel routes, as cited in [22]. A standout feature of this approach is its emphasis on data privacy. Unlike traditional systems, FL ensures that data generated by individual vehicles or sensors is not required to be sent to a central repository. Instead, learning and model improvements occur at the edge, ensuring data remains decentralized. All this is accomplished while ensuring data privacy, as the data generated by individual vehicles and sensors does not have to be centrally collected to build and improve the predictive models. The ITS model envisages a network of interconnected vehicles that communicate with each other and intelligent infrastructure [23]. The envisioned model for ITS is a highly interconnected network where vehicles are not isolated entities. They are part of a larger ecosystem, communicating continuously with each other and with smart infrastructure components. However, implementing such a vision is not without challenges. These challenges can be broadly categorized into four main areas:

1. **System complexity:** The intricate nature of ITSs, with multiple components interacting simultaneously, adds layers of complexity to the system.
2. **Model performance:** The dynamic and ever-changing environment of ITSs presents challenges in maintaining consistent model performance. Models that rely solely on static local intelligence often find adapting to these dynamic changes challenging, resulting in performance degradation [23].
3. **Privacy concerns:** Ensuring user and data privacy becomes paramount with the increasing interconnectivity and data sharing. The dynamic nature of ITSs further amplifies these concerns. These obstacles are primarily clustered into four critical areas: system complexity, model performance, privacy concerns, and data management. The dynamic nature of ITS environments poses a significant hurdle regarding privacy concerns [24].
4. **Data management:** As the ITS network expands, so does the number of nodes capable of processing data. This growth necessitates efficient data management strategies, especially given the constraints of roadside units. Traditional machine learning tech-

niques might face difficulties when applied in such scenarios, particularly during the training phase. The limited data storage capacities of roadside units can hamper the effectiveness of these techniques. Models built on static local intelligence need more flexibility to adapt to such changes, leading to a sharp decline in performance. The growing number of network nodes with data processing capabilities makes data management a significant concern [25]. Since roadside units have limited resource availability, special attention must be paid to efficient data storage strategies. Traditional localized ML techniques may be handicapped during the training phase due to the constraints in data storage capacity at the roadside units.

Related Work

FedGRU, an algorithm that combines Federated Learning with Gated Recurrent Unit (GRU) networks, is proposed for privacy-focused traffic flow prediction [26]. This approach excels in both preserving privacy and prediction accuracy while employing Federated Averaging to reduce communication overhead. In contrast, another study integrates Federated Learning and blockchain technology to maintain data privacy and integrity in Intelligent Transportation Systems (ITSs), using a blockchain-based smart contract to securely aggregate threat-detection models trained on individual vehicles securely [27]. However, this approach shows a slight trade-off with a 7.1% decrease in detection accuracy and precision. A survey offers a comprehensive overview of combining blockchain and Federated Learning to address data privacy and security in the Internet of Vehicles (IoVs), identifying key challenges and future research directions [28]. Similarly, a blockchain-based asynchronous Federated Learning scheme called DBAFL is introduced for intelligent public transportation systems [29]. This scheme balances efficiency, reliability, and learning performance using a committee-based consensus algorithm and a dynamic scaling factor.

A thorough review of Federated Learning applications in Connected and Automated Vehicles (CAVs) analyzes data modalities, evaluates various applications, and outlines future research directions [30]. Another study proposes a contextual client selection pipeline for Federated Learning in transportation systems, using Vehicle-to-Everything (V2X) messages to predict latency and select clients accordingly [31]. A Federated Learning framework designed for autonomous controllers in CAVs is introduced, presenting a novel algorithm called Dynamic Federated Proximal (DFP) that outperforms traditional machine learning solutions in various traffic scenarios [32]. Transformation of the Internet of Vehicles into Intelligent Transportation Systems through advancements like 5G networks is discussed, identifying key challenges such as scalability and data privacy while proposing Federated Learning as a solution [33]. A study addresses the non-identical data distribution across clients in Federated Learning systems, introducing a new FedOT scheme based on the Optimal Transport theory [34]. Lastly, communication challenges in Federated Learning within dynamic and dense vehicular networks are addressed, introducing a Communication Framework for Federated Learning (CF4FL) that reduces training convergence time by 39% [35].

Federated Optimal Transport (FedOT) is introduced to address data distribution issues in Federated Learning, validated through numerical tests [36]. Selective Federated Reinforcement Learning (SFRL) aims to improve the efficiency and adaptability of Connected Autonomous Vehicles through a unique selection process, confirmed by extensive simulations [37]. FedSup employs Bayesian Convolutional Neural Networks for fatigue detection in the Internet of Vehicles, showcasing reduced communication costs and improved training [38]. Federated Transfer-Ordered-Personalized Learning (FedTOP) is tailored for driver monitoring, demonstrating improved accuracy, efficiency, and scalability across two real-world datasets [39,40]. A Hybrid Federated and Centralized Learning (HFCL) framework merges the advantages of federated and centralized learning, achieving up to 20% higher accuracy and 50% less communication overhead [41]. Driver Activity Recognition (DAR) is explored through a Federated Learning model, showing competitive performance

in centralized and decentralized settings while considering data privacy and computational resources [42].

While the existing body of literature extensively covers various aspects of Big Data using FL in IoT-enabled ITS, it primarily focuses on privacy and data distribution. However, a notable gap remains in exploring FL systems' real-time adaptability and resilience to dynamic changes in performance and data distribution in vehicular settings. It generally needs to offer an integrated, IoT-enabled Big Data architecture that addresses data integration and real-time processing while maintaining data privacy. Moreover, current studies often rely on generic Federated Averaging techniques, needing a personalized approach tailored to the unique data characteristics of individual clients in a vehicular network. Our work fills these critical gaps by introducing a Big Data analytics architecture that synergizes FL and IoT technologies for a more robust ITS. We diverge from conventional Federated Averaging techniques by introducing a personalized algorithm enhanced by local fine-tuning, weighted averaging, and custom learning rates. Custom learning rates refer to adjusting the learning rate during training rather than using a fixed rate. The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. FL involves training on multiple decentralized devices or servers (clients) and aggregating the updates on a central server. The learning rate can be crucial in both the client and server updates. Additionally, we employ transfer and ensemble learning strategies to optimize pre-existing models for specialized tasks, thereby improving prediction accuracy. These contributions are empirically validated through a comprehensive suite of tests using real-world data, thereby advancing the field by addressing these unmet needs.

3. Proposed Framework

Our proposed architecture is designed to seamlessly equip Big Data analytics with FL in an IoT-enabled ITS. The proposed approach is a personalized FL approach used to tailor the global aggregation and averaging for improved performance. Various personalization methods are utilized to enhance the Federated Averaging (FedAvg) algorithm. Local fine-tuning and weighted averaging tailor the global model to individual client data. Custom learning rates are utilized to boost the performance further. Regular evaluations are advised to maintain model efficacy. Overall, these approaches offer a robust strategy for personalizing FedAvg. The architecture comprises five major modules or layers, each addressing specific requirements to ensure robust, real-time analytics while preserving user privacy. The architecture leverages ensemble techniques to enhance model performance. The proposed model is depicted in Figure 1.

3.1. Big Data Preprocessing

The first layer of our architecture serves a crucial role in preprocessing the extensive volume of Big Data generated by IoT-enabled ITSs. In real-world scenarios, data often comes with a lot of 'noise' that can adversely affect the performance of machine learning models. Hence, this step is crucial for maintaining the dataset's integrity. It is worth noting that these preprocessing techniques were tailored explicitly for our dataset, which contained numerous missing values and needed modification to suit the problem of vehicle detection. These comprehensive preprocessing steps have been vital for preparing the dataset for further analytics, ensuring quality and making it amenable to solving complex problems like vehicle detection. This layer consists of four integral sub-modules designed to address specific challenges:

1. The Missing Values Management sub-module employs a sophisticated imputation algorithm to address the issue of data gaps. Given that our dataset had many missing values, this sub-module ensures that the dataset remains comprehensive and reliable for further analysis.
2. The Data Reduction sub-module comes into play to make the dataset more manageable in size and computational complexity. We utilize advanced techniques like

Principal Component Analysis (PCA), which reduce the data’s dimensionality and retain the most critical features and relationships within the dataset. This is particularly important in Big Data analytics, where computational resources could become a bottleneck.

3. The Data Filtration sub-module is designed to enhance the data quality by using statistical methods to identify and remove noise and outliers.
4. Data encoding is a critical preprocessing technique in Big Data analytics and machine learning. It involves converting raw data into a format easily ingested and analyzed by data algorithms. The primary aim is to transform the data into a form that reduces complexity and size while retaining the essential features and relationships within the data. In Big Data, which often involves massive and heterogeneous datasets, encoding is vital for reducing storage space, improving computational efficiency, and enabling faster data processing.

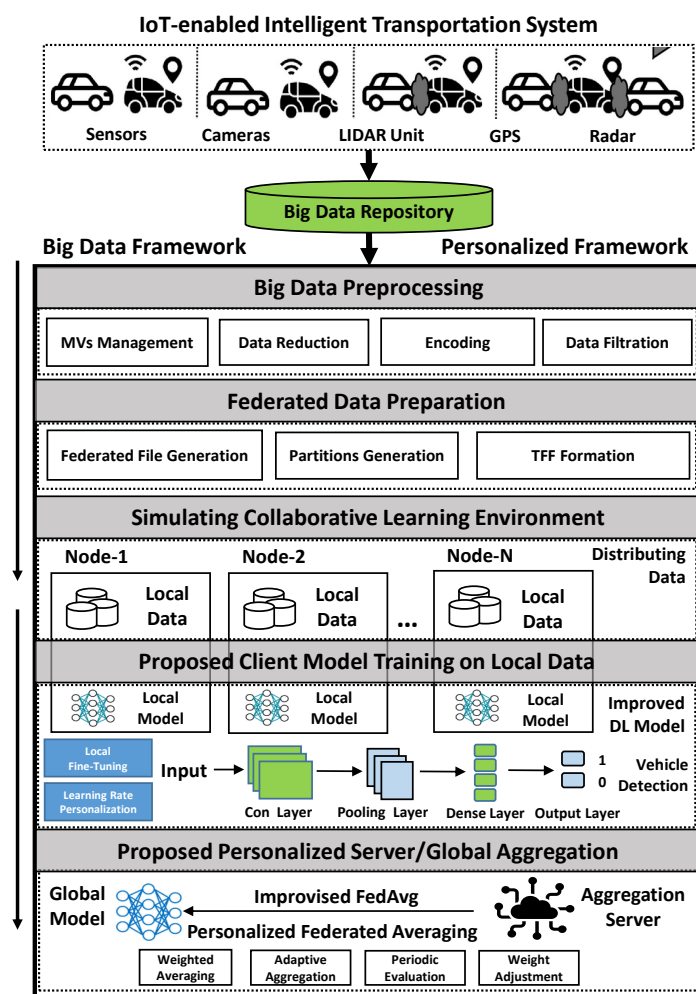


Figure 1. Proposed Framework.

3.2. Federated Data Preparation

Federated Learning requires data to be in a specific format. This layer transforms the pre-processed data into a format suitable for FL. The transformation includes data tokenization, batching, and serialization. In our methodological framework, we have emphasized the creation of meta-files, the arrangement of directories, and the strategic segmentation of data, among other vital activities. Initially, we produced metadata to assist with efficient data mapping processes. Subsequently, our image directories were systematically organized for effortless data retrieval. With a focus on effective data handling, we divided

our dataset into discrete ‘shards’ to facilitate experimental operations. We established ordered dictionaries to guarantee consistent data access. In addition, our dataset was custom formatted to support Federated Averaging, which is a vital element in FL. Federated Averaging allows for the aggregation of local model updates from multiple devices in a decentralized manner, ensuring efficient global model training without compromising data privacy. Given its significance in FL, our dataset was specifically custom-formatted to align with the requirements of Federated Averaging, ensuring seamless integration and optimization for our FL-based analysis. Beyond that, we constructed a comprehensive directory hierarchy for image categorization, which aids in more evenly disseminating data across client devices. Such a structured approach optimizes the dataset and enhances the training milieu, allowing machine learning algorithms to discern class-specific features better. This degree of meticulous structuring, whereby images are sorted into separate class-specific directories, contributes to a decrease in classification errors, ultimately elevating the overall accuracy and effectiveness of the model. The federated data preparation, including client map generation, is described as Algorithm 1.

Algorithm 1 Generate Clients Map for Federated Learning

```

1: procedure GEN_CLIENTS_MAP(train_ds, size, alias)
2:   Initialize train_iter as iterator from train_ds
3:   Initialize empty ordered dictionary clients_map
4:   Compute batch_from_ds = length(train_ds) / size
5:   for index_ in 0 to size - 1 do
6:     Initialize empty lists data_list, label_list
7:     for each batch in batch_from_ds do
8:       Retrieve image, label from train_iter.next()
9:       Convert label, image to numpy arrays
10:      Append image to data_list
11:      Append label to label_list
12:    end for
13:    Create ordered dictionary data with keys ‘pixels’ and ‘label’
14:    Add data to clients_map with key client_alias_index_
15:    Clear Keras session
16:  end for
17:  return clients_map
18: end procedure

```

3.3. Simulating Collaborative Learning Environment

To mimic real-world applications, a collaborative learning environment is simulated. This involves:

- The creation of various clients: Various nodes or clients are created to simulate a distributed environment. A dynamic algorithm is proposed to vary the number of clients for multiple experiments and verification.
- Data distribution: Data that have been pre-processed and formatted are allocated across these clients to mimic real-world conditions. A dynamic algorithm is also introduced to distribute the data among various nodes, allowing for varying data sizes and samples to be held by different nodes.

Within our FL procedure, we load directories pre-configured with positive, normal, and overall image data into memory for either the training or testing phase. These directories serve multiple roles, such as facilitating label assignment, determining batch size, resizing images, shuffling data, and managing color channels. This ensures optimized resource use while mitigating the risk of memory overload. By organizing images into batches for training, we align with FL’s standard practices for data segmentation and enable targeted performance evaluations, which are vital when dealing with classes of varying or imbalanced attributes. For the node-mapping aspect of FL, the dataset of images is

transformed into an ordered dictionary, which allows TensorFlow objects to facilitate the partitioning of nodes, each receiving a distinct subset of the entire dataset. This function iterates through the dataset, dividing it into smaller batches that are subsequently allocated to simulated client nodes. This mimics a dispersed transportation data setting where each vehicle can access only a fraction of the collective dataset. By disseminating data among multiple nodes, our methodology replicates authentic FL conditions and supports decentralized model training. Each node conducts training on its specific data subset; afterward, improvements to the global model are synthesized from all the updates received from the nodes. This enhances the FL process as a whole. Furthermore, our proposed algorithm has unique dynamic node creation and data allocation capabilities. The number of nodes can be dynamically generated through the algorithm, offering flexibility in creating a scalable and adaptable collaborative learning environment. The process of creating the collaborators is depicted in Algorithm 2.

Algorithm 2 Creating Collaborators for Federated Learning Round

```

1: procedure CREATING_COLLABORATORS(dataset, client_data)
2:   Initialize Sample Size:
3:   Calculate sample_size as half of the total number of client IDs in client_data
4:   Sample Clients:
5:   Randomly select sample_size number of client IDs without replacement
6:   Store these in sampled_clients_ids
7:   Generate Sampled Client Datasets:
8:   for each client_id in sampled_clients_ids do
9:     Generate a TF dataset
10:  end for
11:  Preprocess Datasets:
12:  Preprocess each dataset using preprocess() function
13:  Store the preprocessed datasets in sampled_clients_data
14:  return sampled_clients_data
15: end procedure

```

3.4. Client Model Training

Machine learning models, such as MLP, CNN, and VGG16, are deployed for training within each client. The system's architecture incorporates two key strategies. First, it utilizes Transfer Learning by fine-tuning pre-existing models like VGG16 to better suit the specialized task. Second, it employs Ensemble Learning by integrating the outputs from multiple models, thereby enhancing the overall prediction accuracy. We employed Multi-Layer Perceptrons (MLPs) as a foundational algorithm to rigorously validate our hypotheses. Often referred to as a class of artificial neural networks, an MLP consists of at least three layers of nodes or neurons: an input layer, one or more hidden layers, and an output layer. As a supervised learning model, MLPs are trained using labeled data for tasks like prediction and classification. Each neuron within a layer is interconnected with every neuron in the following layer via weighted connections. To optimize global accuracy, we explored a variety of algorithms for different client nodes. We strategically deployed MLPs in specific client scenarios where we assessed they would yield favorable outcomes. In parallel, we also implemented other models, like CNN and VGG16, to enrich our proposed FL framework. Each client model is trained on a local dataset during the FL collaborative environment simulation, specifically allocated to that client node. By generating a diverse set of clients, we could closely emulate real-world scenarios. It explains how these different algorithms perform in a Federated Learning context. Each client's learning rates are personalized based on their local loss landscapes. Some nodes benefit from a faster learning rate, while others might need a slower one for better convergence.

3.5. Proposed Personalized Server/Global Aggregation

One well-known constraint in FL is the network bandwidth that limits the rate at which local updates from different organizations can be combined in the cloud. To mitigate this, FedAvg uses local data for gradient descent optimization before conducting a weighted average aggregation of the models uploaded by each node. The algorithm proceeds iteratively, updating the global model in each training round based on the contributions from participating organizations. Traditional centralized learning approaches merge data from different organizations into a single database. This results in considerable communication costs and risks to data privacy. To tackle these challenges, we introduce a privacy-preserving module equipped with a prediction algorithm for vehicle detection. Our solution starts by leveraging the FedAvg algorithm for parameter aggregation, collecting gradient data from various nodes. We then introduce an enhanced version of FedAvg to minimize communication overhead and perform efficient aggregation. This is particularly beneficial for large-scale and distributed prediction tasks.

The server layer aggregates the trained models from all clients to create a comprehensive global model. This is carried out using:

- Individual contributions from clients are weighted based on their data volume and model performance.
- An improvised Federated Proximal algorithm with Federated Averaging is used for robust aggregation, accounting for system heterogeneity and stragglers.
- Instead of simple averaging, we used weighted averaging, where the weights are determined based on each client's data distribution, quality, or performance metrics. This will give more influence to clients with more relevant or high-quality data.
- In place of straightforward averaging, we deploy advanced aggregation techniques that factor in the statistical attributes of client updates, such as variance or confidence intervals, for a better-informed global update.

The aim of using weighted averaging is to consider data's uneven distribution and quality across clients. By doing so, we prevent clients with minimal or low-quality data from dominating the global model update. Instead of uniformly averaging the model updates from each client, we assigned weights to each client's update. The weights were calculated based on the client's data distribution, quality, and training performance.

Weight Calculation: For client i , let d_i be its data size, q_i represent the quality score (based on internal metrics), and p_i represent its training performance. The weight w_i for the client can be formulated as:

$$w_i = \lambda \times \frac{d_i}{\sum_{j=1}^N d_j} + \mu \times q_i + (1 - \lambda - \mu) \times p_i \quad (1)$$

$$w_i = \lambda \times \frac{\sum_{j=1}^N d_j}{d_i} + \mu \times q_i + (1 - \lambda - \mu) \times p_i \quad (2)$$

where λ and μ are hyperparameters determining the significance of data size and data quality, respectively.

The idea behind adaptive aggregation is to consider the variations in model updates from different clients. By accounting for these attributes, we ensure a robust global model update. Instead of naive averaging, we integrated the statistical attributes of client updates to formulate the global update. This method ensures that outliers or divergent updates do not adversely impact the global model.

Aggregation Formula: Let u_i be the model update from client i , and v_i be its variance. The aggregated update U is then computed as:

$$U = \frac{\sum_{i=1}^N u_i}{1 + \gamma \times v_i} \quad (3)$$

$$U = \frac{\sum_{i=1}^N u_i}{1 + \gamma \times v_i} \quad (4)$$

where γ is a hyperparameter determining the influence of variance on the aggregation.

Our proposed algorithm for model aggregation combines the client updates to construct a unified global model. This guarantees a well-balanced and precise representation of data from all collaborating clients. During each communication round, each device calculates a local update, which is then transmitted to a central server for aggregation. This loop persists until the model converges or a predefined number of communication rounds is met. Metrics like training loss and accuracy are diligently tracked in every round. The FedAvg algorithm not only amalgamates these local updates but also refines the global model by considering the volume and quality of each client's data. The study follows a cyclical training protocol, where each round selectively chooses client data subsets for training and modifies the server's status accordingly. Performance indicators like accuracy are continuously logged, offering a dynamic snapshot of how the model fares over time. This strategy promotes decentralized and cooperative model training, achieving fairness and comprehensive data representation. Alongside the standard FedAvg, we also experimented with an optimized version of FedAvg to yield personalized and optimal outcomes. Additionally, we implemented an advanced version of FedProx to refine global accuracy further. This allows for a more nuanced and compelling Federated Learning process. The proposed algorithm is depicted in Algorithm 3.

Algorithm 3 Personalized Federated Averaging Algorithm

```

1: procedure PERSONALIZED_ALGORITHM(node_data, num_rounds)
2:   Initialize Metrics and Optimizers:
3:   Define Client and Server optimizer functions
4:   Specify Model Input:
5:   input_spec = get_input_spec(node_data)
6:   Build Federated Averaging Process:
7:   Build the federated averaging process using TFF
8:   Perform weighted averaging for global model customization
9:   Initialize Federated Averaging:
10:  Initialize state for federated averaging
11:  for round_num in range(num_rounds) do
12:    Create federated_train_data
13:    Update state and metrics
14:  end for
15:  Local Fine-Tuning:
16:  Fine-tune model locally, utilize custom learning rates
17:  Regularly evaluate model for efficacy
18:  Clear Session:
19:  Clear Keras session
20:  return losses, accuracy
21: end procedure

```

4. Results

4.1. Dataset and Experimental Setup Detail

We employ the re-labeled Udacity Self-Driving Car Dataset provided by Roboflow for the initial training phase of our models. This dataset comprises 15,000 images of 1280×1280 resolution and contains 97,942 annotations across 11 categories. The annotations are designed to be compatible with YOLO formatting and include details such as the object's class ID, the coordinates for the center of the object in both X and Y dimensions, and the dimensions of the bounding box. In our study, we chose the Udacity Self-Driving Car Dataset due to its comprehensive array of features pertinent to ITSs [43–45]. The dataset's

meticulous structure and substantial size make it exceptionally suited for exploring sophisticated FL techniques, including Federated Learning algorithms, specifically in vehicle recognition. We subjected our architectural framework to an exhaustive set of tests to assess its performance, scalability, and resilience under different conditions. For computational resources, our setup included an Intel(R) Core (TM) i9 processor operating at 3.20 GHz, supplemented by 64 GB of RAM and running on a Windows Operating System. Some computations were also done on Google Collab to corroborate our findings further. Our experiments were designed within the TensorFlow Federated (TFF) environment, enabling us to simulate real-world collaborative learning settings for scrutinizing FL algorithms. All coding tasks were performed in Python, utilizing Jupyter Notebook as our integrated development environment.

4.2. Experiments and Results

The results from Table 1 are compelling, demonstrating the superior performance of our proposed personalized approach compared to the standard FedAvg method. In a simulated real-world collaborative environment featuring 10 nodes, the model's accuracy using our proposed approach reached 93.27%, significantly outpacing the standard FedAvg method, which achieved 87.35% over 50 communication rounds. The results of communication rounds are depicted in Figure 2. Furthermore, the training loss is also depicted in Figure 3.

The results presented in Table 2 further validate the effectiveness of our proposed approach, this time in a more complex federated environment involving 20 nodes. Again, over 50 communication rounds, our tailored Federated Averaging algorithm significantly outperforms the standard FedAvg method. While FedAvg achieves an accuracy rate of 87.30%, our proposed approach raises the bar by attaining an accuracy of 92.89%. The advancement in accuracy is consistent with the objectives in the abstract and previous findings with 10 nodes. It accentuates the architecture's scalability, indicating that the proposed architecture remains robust, efficient, and highly accurate as we increase the number of nodes participating in the FL system. This scalability is particularly vital for IoT-enabled ITS, where the number of edge devices and the volume of data they generate can be highly variable and massive. The accuracy for all communication rounds and the training loss are depicted in Figure 4 and Figure 5, respectively.

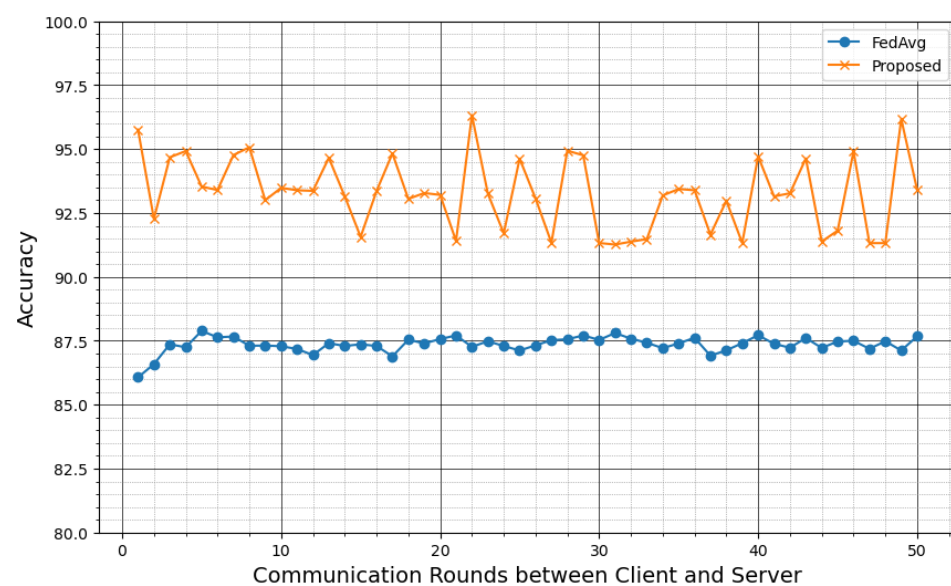


Figure 2. FedAvg vs. proposed model accuracy (10 nodes).

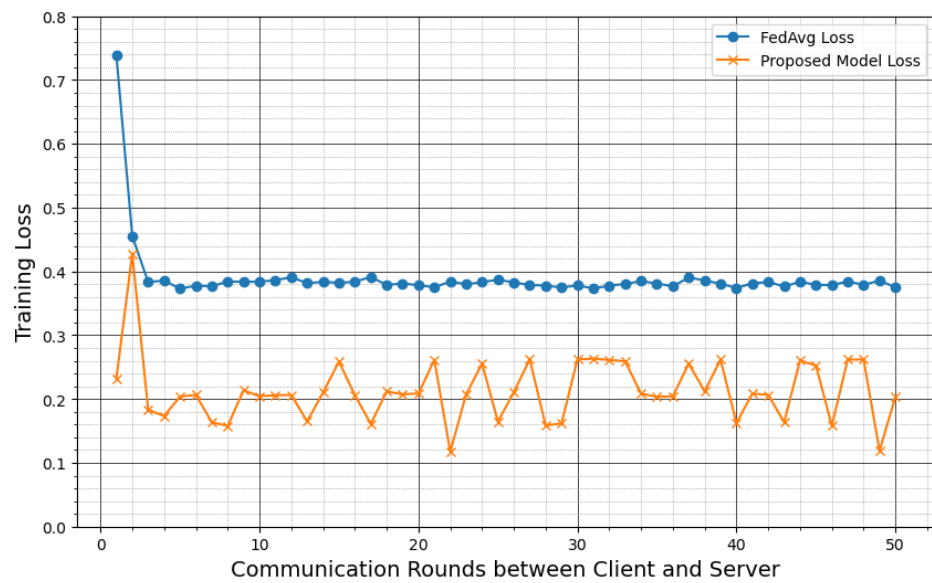


Figure 3. FedAvg vs. proposed model training loss (10 nodes).

Table 1. Accuracy using 10 Nodes.

| Nodes/Collaborators | Communication Rounds | Approach | Accuracy |
|---------------------|----------------------|----------|----------|
| 10 | 50 | FedAvg | 87.35% |
| | | Proposed | 93.27% |

Table 2. Accuracy using 20 nodes.

| Nodes/Collaborators | Communication Rounds | Approach | Accuracy |
|---------------------|----------------------|----------|----------|
| 20 | 50 | FedAvg | 87.30% |
| | | Proposed | 92.89% |

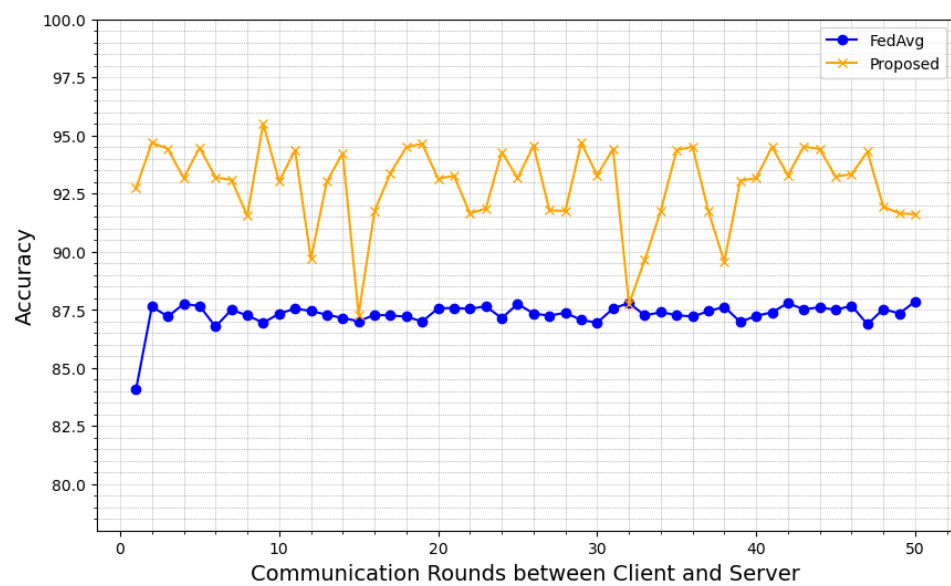


Figure 4. FedAvg vs. proposed model accuracy (20 nodes).

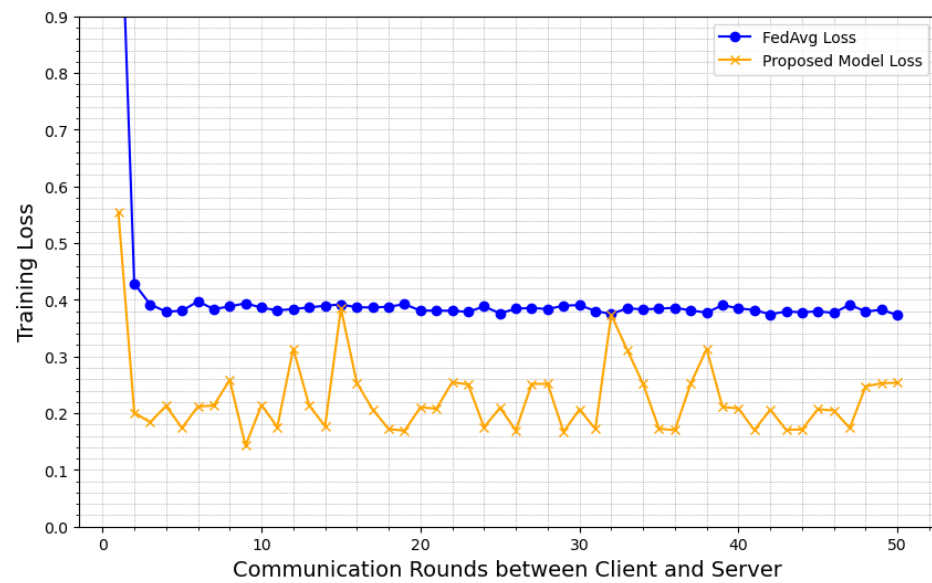


Figure 5. FedAvg vs. proposed model training loss (20 nodes).

This outcome is statistically significant and practically impactful, especially given the real-world complexity and privacy concerns embedded in ITS. The improved accuracy with more clients indicates that the system can adapt to increased complexity and heterogeneity in the data, typical conditions in expansive urban ITS networks. Our computational setup, which features robust hardware and advanced data analytics tools, is an ideal test bed for such multi-node, real-time applications.

The data shown in Table 3 takes our evaluation further by extending the FL environment to include 30 nodes. Like the previous configurations with 10 and 20 clients, this scenario also incorporates 50 communication rounds. The proposed Federated Averaging approach again outshines the standard model. FedAvg clocks an accuracy of 87.25%, whereas our approach leaps forward, achieving an accuracy of 92.96%. The accuracy for all communication rounds and the training loss using 30 nodes are depicted in Figure 6 and Figure 7, respectively.

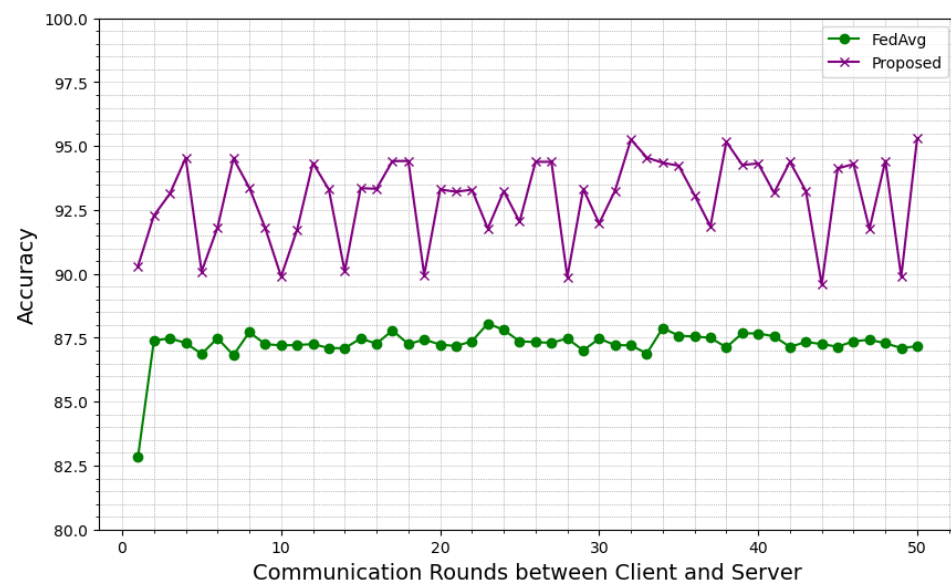


Figure 6. FedAvg vs. proposed model accuracy (30 nodes).

Table 3. Accuracy using 30 nodes.

| Nodes/Collaborators | Communication Rounds | Approach | Accuracy |
|---------------------|----------------------|----------|----------|
| 30 | 50 | FedAvg | 87.25% |
| | | Proposed | 92.96% |

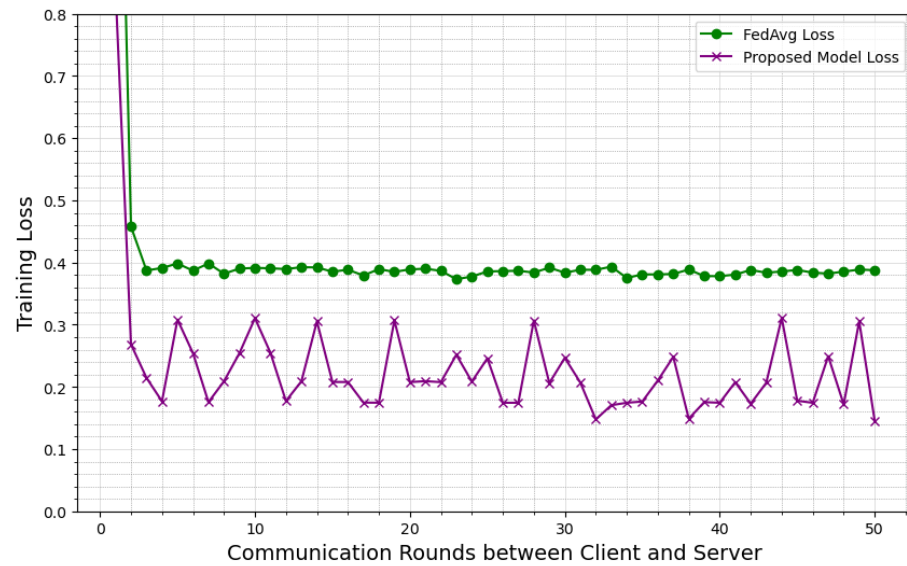


Figure 7. FedAvg vs. proposed model training Loss (30 nodes).

In a compelling advancement to our research, we did not just scale the number of nodes; we also upped the ante by extending the number of communication rounds to robustly substantiate the efficiency of our groundbreaking model across a more extensive range of interaction cycles. Table 4 illuminates an intriguing pattern: as we ramp up the communication rounds our avant-garde model’s accuracy elevates significantly. When we implemented 30 nodes and conducted 100 communication rounds, the FedAvg model achieved an accuracy of 87.27%. In contrast, our pioneering approach delivered an impressive accuracy of 93.09%. In a head-to-head comparison, our novel personalized averaging mechanism for FL outperforms traditional FL algorithms, reinforcing our claim of superior accuracy. Figures 8 and 9 comprehensively show the accuracy and training loss metrics across all communication rounds.

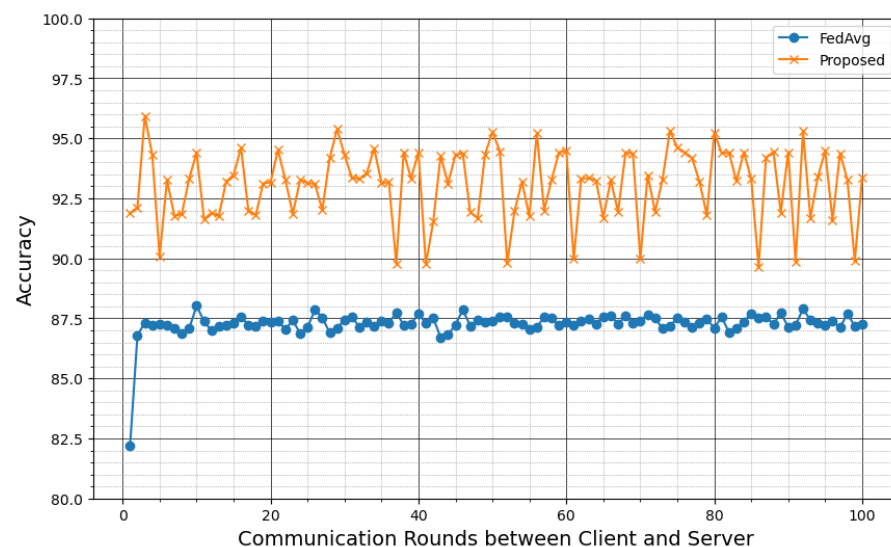
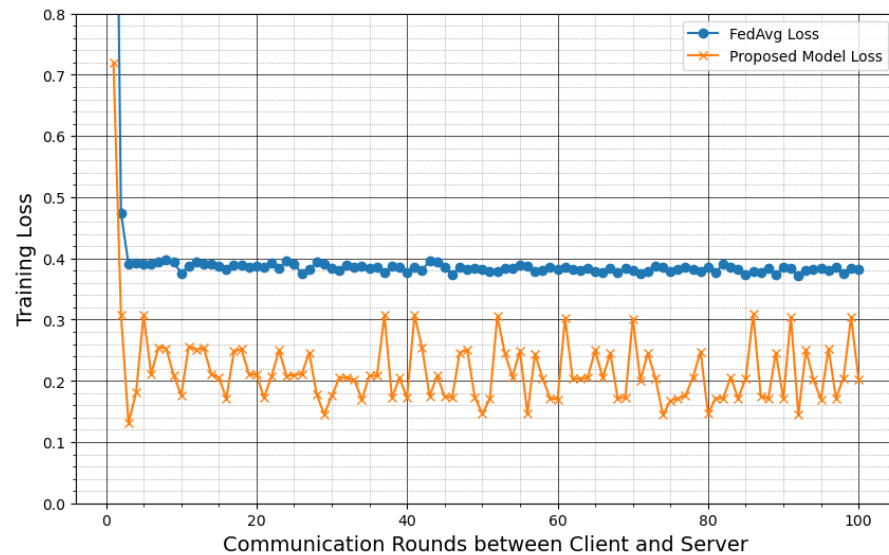


Figure 8. FedAvg vs. proposed model accuracy using 100 rounds (30 nodes).

Table 4. Accuracy using 30 nodes with 100 communication rounds.

| Nodes/Collaborators | Communication Rounds | Approach | Accuracy |
|---------------------|----------------------|----------|----------|
| 30 | 100 | FedAvg | 87.27% |
| | | Proposed | 93.09% |

**Figure 9.** FedAvg vs. proposed model training loss using 100 rounds (30 nodes).

This result adds credibility to the architecture's performance and scalability, demonstrating that the system maintains high accuracy even as the network complexity grows. Notably, the accuracy rate remains consistently high across varying numbers of clients: 10, 20, and 30. This demonstrates that the algorithm's performance remains consistent with increased network size, a commonly seen pitfall in FL implementations.

5. Discussion

This research aimed to address the challenges arising from the heterogeneity of devices, the dynamic conditions of ITSs, and data privacy concerns in the Big Data landscape. Our proposed architecture leverages an optimized Federated Averaging strategy to address these issues effectively, offering a robust solution in terms of scalability, real-time decision making, and data privacy preservation. Our empirical findings are aligned with these objectives. The accuracy of 93.27% underscores the model's proficiency in real-time Big Data analytics and highlights its capability in a real-life federated environment. This result further substantiates our claim that personalized approaches to Federated Averaging are effective and practical for modern ITS utilizing big data. In our study, the primary focus has been on the accuracy and robustness of the proposed architecture. Our personalized Federated Averaging method consistently achieved higher accuracy levels compared to the standard FedAvg, with the top performance being 93.27% accuracy using 30 nodes. This accentuates the model's proficiency in real-time Big Data analytics within a federated environment. In terms of efficiency and convergence speed, our method incorporates personalization techniques, which, while enhancing accuracy, can occasionally introduce slightly extended convergence times compared to the standard FedAvg. Nevertheless, the benefits of the improved accuracy outweigh the marginal increase in training time, especially when considering the critical nature of decision making in real-world ITS scenarios. The testing phase for our proposed approach remained comparable in speed to the standard FedAvg, ensuring timely decision making. Our model balances accuracy and efficiency, making it a promising solution for modern ITSs utilizing Big Data. Concerning stability, it is essential to distinguish between brief variations and sustained stability. Although our

technique shows increased short-term variations post-convergence, its overarching trend suggests sustained performance dominance over extended durations. The tailored strategy guarantees that the overarching model stays resilient, even when there is diversity in data distributions across individual nodes. Notably, the fluctuations we observed are within a range of 4–5%, without any significant deviations.

The proposed model shows more pronounced fluctuations in loss and accuracy after convergence compared to the standard FedAvg. This is because of the following factors:

- **Personalized learning approach:** Our approach diverges from standard Federated Averaging by incorporating personalized techniques. While this results in better-tailored models for individual nodes, it can also introduce variability in the global model, especially when individual client models differ significantly.
- **Local fine-tuning and weighted averaging:** We employ local fine-tuning and weighted averaging mechanisms that result in diverse local updates, contributing to oscillations during global model updates.
- **Custom learning rates:** As mentioned in the manuscript, we leverage custom learning rates, which can sometimes lead to more pronounced fluctuations, especially when the learning rate is not optimally set for some training rounds.

6. Conclusions

This paper tackles the burgeoning challenges posed by the intersection of Big Data analytics, the Internet of Things (IoT), and ITS. With data volume, variety, and velocity becoming increasingly formidable, traditional data analytics frameworks must be revised. Our research fills a significant gap by introducing a comprehensive Big Data analytics architecture tailored for an IoT-enabled ITS. Leveraging FL, we address pressing data integration issues, real-time analytics, and privacy concerns. Departing from conventional Federated Averaging methods, we champion a more personalized approach that refines global models to suit individual client data better. This personalization is achieved through innovative techniques, including local fine-tuning, weighted averaging, and custom learning rates. Transfer and ensemble learning approaches further amplify the model's accuracy and robustness. Empirical validations using the Udacity Self-Driving Car Dataset underline the efficacy of our architecture in terms of scalability, real-time decision making, and data privacy preservation. Overall, this work advances the state of the art in FL and ITS. It sets a new standard for how personalized, real-time Big Data analytics can be effectively conducted in complex, dynamic urban transportation environments. We attained accuracy levels of 93.27%, 92.89%, and 92.96% for our proposed model in a Federated Learning architecture with 10 nodes, 20 nodes, and 30 nodes, respectively. This is particularly noteworthy given the consistently high accuracy maintained across different client counts, be it 10, 20, or 30, showcasing the algorithm's resilience even as the network's complexity escalates. This constancy in performance, even with an expanding network size, signifies a remarkable deviation from typical pitfalls observed in FL systems. The architecture we present, fortified by an optimized Federated Averaging strategy, offers a potent solution for data privacy.

Author Contributions: S.K. conceived and planned the conceptualization and writing of the original draft preparation, methodology proposal, experiments performed, and visualization. A.S., M.U.T. and M.A. contributed to the investigation and validation. A.S., M.U.T. and M.A. also contributed to the analysis and interpretation of the results by reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Details of the data are provided in Section 4.1.

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-----|-----------------------------------|
| IoT | Internet of Things |
| ITS | Intelligent Transportation System |
| ML | Machine Learning |
| FL | Federated Learning |

References

- Muthuramalingam, S.; Bharathi, A.; Rakesh Kumar, S.; Gayathri, N.; Sathiyaraj, R.; Balamurugan, B. IoT based intelligent transportation system (IoT-ITS) for global perspective: A case study. In *Internet of Things and Big Data Analytics for Smart Generation*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 279–300.
- Ounoughi, C.; Yahia, S.B. Data fusion for ITS: A systematic literature review. *Inf. Fusion* **2023**, *89*, 267–291. [[CrossRef](#)]
- Babar, M.; Khattak, A.S.; Jan, M.A.; Tariq, M.U. Energy aware smart city management system using data analytics and Internet of Things. *Sustain. Energy Technol. Assess.* **2021**, *44*, 100992. [[CrossRef](#)]
- Ashfaq, T.; Khalid, R.; Yahaya, A.S.; Aslam, S.; Azar, A.T.; Alkhalifah, T.; Tounsi, M. An intelligent automated system for detecting malicious vehicles in intelligent transportation systems. *Sensors* **2022**, *22*, 6318. [[CrossRef](#)]
- Babar, M.; Arif, F.; Irfan, M. Internet of things-based smart city environments using big data analytics: A survey. In *Recent Trends and Advances in Wireless and IoT-Enabled Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 129–138.
- Hashmi, M.U.; Hussain, M.; Babar, M.; Qureshi, B. Single-Timestamp Skew Correction (STSC) in V2X Networks. *Electronics* **2023**, *12*, 1276. [[CrossRef](#)]
- Farman, H.; Khan, Z.; Jan, B.; Boulila, W.; Habib, S.; Koubaa, A. Smart transportation in developing countries: An Internet-of-Things-based conceptual framework for traffic control. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8219377. [[CrossRef](#)]
- Stergiou, C.L.; Bompoli, E.; Psannnis, K.E. Security and Privacy Issues in IoT-Based Big Data Cloud Systems in a Digital Twin Scenario. *Appl. Sci.* **2023**, *13*, 758. [[CrossRef](#)]
- Hijji, M.; Iqbal, R.; Pandey, A.K.; Doctor, F.; Karyotis, C.; Rajeh, W.; Alshehri, A.; Aradah, F. 6G connected vehicle framework to support intelligent road maintenance using deep learning data fusion. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7726–7735. [[CrossRef](#)]
- Talaoui, Y.; Kohtamäki, M.; Ranta, M.; Paroutis, S. Recovering the divide: A review of the big data analytics—strategy relationship. *Long Range Plan.* **2023**, *56*, 102290. [[CrossRef](#)]
- Qi, Q.; Xu, Z.; Rani, P. Big data analytics challenges to implementing the intelligent Industrial Internet of Things (IIoT) systems in sustainable manufacturing operations. *Technol. Forecast. Soc. Chang.* **2023**, *190*, 122401. [[CrossRef](#)]
- Vasa, J.; Thakkar, A. Deep learning: Differential privacy preservation in the era of big data. *J. Comput. Inf. Syst.* **2023**, *63*, 608–631. [[CrossRef](#)]
- Nair, A.K.; Sahoo, J.; Raj, E.D. Privacy preserving Federated Learning framework for IoMT based big data analysis using edge computing. *Comput. Stand. Interfaces* **2023**, *86*, 103720. [[CrossRef](#)]
- Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M.V.; Herrera, F.; Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Inf. Fusion* **2023**, *90*, 148–173. [[CrossRef](#)]
- Supriya, Y.; Gadekallu, T.R. Particle Swarm-Based Federated Learning Approach for Early Detection of Forest Fires. *Sustainability* **2023**, *15*, 964. [[CrossRef](#)]
- Mahmood, Z.; Jusas, V. Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy. *Electronics* **2022**, *11*, 1624. [[CrossRef](#)]
- Olukoya, O. Assessing frameworks for eliciting privacy & security requirements from laws and regulations. *Comput. Secur.* **2022**, *117*, 102697.
- Harth, N.; Anagnostopoulos, C.; Voegel, H.J.; Kolomvatsos, K. Local & Federated Learning at the network edge for efficient predictive analytics. *Future Gener. Comput. Syst.* **2022**, *134*, 107–122.
- Arafeh, M.; Ould-Slimane, H.; Otrok, H.; Mourad, A.; Talhi, C.; Damiani, E. Data independent warmup scheme for non-IID federated learning. *Inf. Sci.* **2023**, *623*, 342–360. [[CrossRef](#)]
- Wijesinghe, A.; Zhang, S.; Qi, S.; Ding, Z. UFed-GAN: A Secure Federated Learning Framework with Constrained Computation and Unlabeled Data. *arXiv* **2023**, arXiv:2308.05870.
- Pati, S.; Baid, U.; Edwards, B.; Sheller, M.; Wang, S.H.; Reina, G.A.; Foley, P.; Gruzdev, A.; Karkada, D.; Davatzikos, C.; et al. Federated learning enables big data for rare cancer boundary detection. *Nat. Commun.* **2022**, *13*, 7346. [[CrossRef](#)] [[PubMed](#)]
- Goto, Y.; Matsumoto, T.; Rizk, H.; Yanai, N.; Yamaguchi, H. Privacy-preserving taxi-demand prediction using federated learning. In Proceedings of the 2023 IEEE International Conference on Smart Computing (SMARTCOMP), Nashville, TN, USA, 26–30 June 2023; pp. 297–302.

23. Elsagheer Mohamed, S.A.; AlShalfan, K.A. Intelligent traffic management system based on the internet of vehicles (IoV). *J. Adv. Transp.* **2021**, *2021*, 4037533. [[CrossRef](#)]
24. Tyagi, A.K.; Nair, M.M. Preserving Privacy using Distributed Ledger Technology in Intelligent Transportation System. In Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing, Noida, India, 4–6 August 2022; pp. 582–590.
25. Holt, C.; Calhoun, J.C. Stale Data Analysis in Intelligent Transportation Platooning Models. In Proceedings of the 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 26–29 October 2022; pp. 0313–0320.
26. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [[CrossRef](#)]
27. Moulahi, T.; Jabbar, R.; Alabdulatif, A.; Abbas, S.; El Khediri, S.; Zidi, S.; Rizwan, M. Privacy-preserving federated learning cyber-threat detection for intelligent transport systems with blockchain-based security. *Expert Syst.* **2023**, *40*, e13103. [[CrossRef](#)]
28. Billah, M.; Mehedi, S.T.; Anwar, A.; Rahman, Z.; Islam, R. A systematic literature review on blockchain enabled federated learning framework for internet of vehicles. *arXiv* **2022**, arXiv:2203.05192.
29. Xu, C.; Qu, Y.; Luan, T.H.; Eklund, P.W.; Xiang, Y.; Gao, L. An efficient and reliable asynchronous federated learning scheme for smart public transportation. *IEEE Trans. Veh. Technol.* **2022**, *72*, 6584–6598. [[CrossRef](#)]
30. Chellapandi, V.P.; Yuan, L.; Zak, S.H.; Wang, Z. A survey of federated learning for connected and automated vehicles. *arXiv* **2023**, arXiv:2303.10677.
31. Song, R.; Lyu, L.; Jiang, W.; Festag, A.; Knoll, A. V2X-Boosted Federated Learning for Cooperative Intelligent Transportation Systems with Contextual Client Selection. *arXiv* **2023**, arXiv:2305.11654.
32. Zeng, T.; Semiari, O.; Chen, M.; Saad, W.; Bennis, M. Federated learning for collaborative controller design of connected and autonomous vehicles. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 5033–5038.
33. Manias, D.M.; Shami, A. Making a case for federated learning in the internet of vehicles and intelligent transportation systems. *IEEE Netw.* **2021**, *35*, 88–94. [[CrossRef](#)]
34. Farnia, F.; Reiszadeh, A.; Pedarsani, R.; Jadbabaie, A. An optimal transport approach to personalized federated learning. *IEEE J. Sel. Areas Inf. Theory* **2022**, *3*, 162–171. [[CrossRef](#)]
35. Sangdeh, P.K.; Li, C.; Pirayesh, H.; Zhang, S.; Zeng, H.; Hou, Y.T. CF4FL: A Communication Framework for Federated Learning in Transportation Systems. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 3821–3836. [[CrossRef](#)]
36. Stergiou, K.D.; Psannis, K.E.; Vitsas, V.; Ishibashi, Y. A Federated Learning Approach for Enhancing Autonomous Vehicles Image Recognition. In Proceedings of the 2022 4th International Conference on Computer Communication and the Internet (ICCCI), Chiba, Japan, 1–3 July 2022; pp. 87–90.
37. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A selective federated reinforcement learning strategy for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1655–1668. [[CrossRef](#)]
38. Zhao, C.; Gao, Z.; Wang, Q.; Xiao, K.; Mo, Z.; Deen, M.J. FedSup: A communication-efficient federated learning fatigue driving behaviors supervision approach. *Future Gener. Comput. Syst.* **2023**, *138*, 52–60. [[CrossRef](#)]
39. Fantauzzo, L.; Fani, E.; Caldarola, D.; Tavera, A.; Cermelli, F.; Ciccone, M.; Caputo, B. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 11504–11511.
40. Yuan, L.; Su, L.; Wang, Z. Federated Transfer-Ordered-Personalized Learning for Driver Monitoring Application. *IEEE Internet Things J.* **2023**, *10*, 18292–18301. [[CrossRef](#)]
41. Elbir, A.M.; Coleri, S.; Papazafeiropoulos, A.K.; Kourtessis, P.; Chatzinotas, S. A hybrid architecture for federated and centralized learning. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1529–1542. [[CrossRef](#)]
42. Doshi, K.; Yilmaz, Y. Federated learning-based driver activity recognition for edge devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3338–3346.
43. Roboflow. Udacity Self-Driving Car Object Detection Dataset. 2020. Available online: <https://public.roboflow.com/object-detection/self-driving-car> (accessed on 15 July 2023).
44. Rajaji, P.; Rahul, S. Detection of Lane and Speed Breaker Warning System for Autonomous Vehicles using Machine Learning Algorithm. In Proceedings of the 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 11–12 August 2022; pp. 401–406.
45. Dubey, S.; Olimov, F.; Rafique, M.A.; Jeon, M. Improving small objects detection using transformer. *J. Vis. Commun. Image Represent.* **2022**, *89*, 103620. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.