Utilizing Federated Learning for Enhanced Real-Time Traffic Prediction in Smart Urban Environments

Mamta Kumari[1], Zoirov Ulmas[2], Suseendra R[3],
Janjhyam Venkata Naga Ramesh[4], Prof. Ts. Dr. Yousef A. Baker El-Ebiary[5]

Assistant Professor in CSE (AI-ML) Dept. Panipat Institute of Engineering and Technology (PIET), Samalakha, India[1]
Assistant Teacher at the Artificial Intelligence Department at the Tashkent State University of Economics[2]
Assistant Professor, Department of IT, Panimalar Engineering College, Chennai[3]
Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Dist., Andhra Pradesh - 522302, India[4]
Faculty of Informatics and Computing, UniSZA University, Malaysia[5]

Abstract—**Federated Learning (FL), a crucial advancement in smart city technology, combines real-time traffic predictions with the potential to enhance urban mobility. This paper suggests a novel approach to real-time traffic prediction in smart cities: a hybrid Convolutional Neural Network-Recurrent Neural Network (CNN-RNN) architecture. The investigation started with the systematic collection and preprocessing of a low-resolution dataset (1.6 GB) derived from real-time Closed Circuit Television (CCTV) traffic camera images at significant intersections in Guntur and Vijayawada. The dataset has been cleaned up utilizing min-max normalization to facilitate use. The primary contribution of this study is the hybrid architecture that it develops by fusing RNN to detect temporal dynamics with CNN for geographic extraction of characteristics. While the RNN's recurrent interactions preserve hidden states for sequential processing, the CNN efficiently retrieves high-level spatial information from static traffic images. Weight adjustments and backpropagation are used in the training of the proposed hybrid model in order to enhance real-time predictions that aid in traffic management. Notably, the implementation is done with Python software. The model reaches a testing accuracy of 99.8% by the 100th epoch, demonstrating excellent performance in the results and discussion section. The Mean Absolute Error (MAE) results, which show a 4.5% improvement over existing methods like Long Short Term Memory (LSTM), Support Vector Machine (SVM), Sparse Auto Encoder (SAE), and Gated Recurrent Unit (GRU), illustrate the efficacy of the model. This demonstrates how well complex patterns may be represented by the model, yielding precise real-time traffic predictions in crowded metropolitan settings. A new era of more precise and effective real-time traffic forecasts is about to begin, thanks to the hybrid CNN-RNN architecture, which is validated by the combined strengths of FL, CNN, and RNN as well as the overall outcomes.**

Keywords—Federated Learning; smart city; convolutional neural network; recurrent neural network; traffic prediction

I. INTRODUCTION

Federated Learning which tackles privacy issues and decentralizes model training, is a paradigm shift in machine learning. Due to the transmission and storage of sensitive data, traditional machine learning models are frequently trained on centralized servers using aggregated datasets, which raises privacy concerns [1]. Conversely, Federated Learning enables cooperative model training across decentralized devices, such as servers, edge devices, and smartphones, without requiring raw data exchange [2]. This novel method preserves the privacy of local data while enabling individual devices to make improvements to the model. Federated Learning is a tempting option for applications in healthcare, finance, and, most importantly, the setting of smart cities. It is especially applicable in situations where data privacy is a top priority [3].

The integration of technology to improve urban living is a hallmark of smart cities, and federated learning is essential to this shift. It is clear that efficient and private-preserving machine learning solutions are needed in the context of smart cities, where enormous volumes of data are produced on a regular basis from many sources [4]. Smart city applications that make use of Federated Learning may train models collaboratively with data from disparate sensors, devices, and systems dispersed across the city. This method is especially applicable to situations where precise decision-making requires the aggregation of real-time data, such as energy management, pollution monitoring, and traffic prediction [5].

Because Federated Learning is decentralized and keeps data localized, it naturally addresses privacy issues. To protect contributor privacy, only model changes are shared rather than raw data that is sent to a central server. This is particularly important in smart cities, where there is a need to manage data from several sources carefully, such as public databases, IoT devices, and security cameras [6]. In order to guarantee the safe and private transmission of model changes, Federated Learning also uses cryptographic methods and secures aggregation protocols. Federated Learning is a desirable alternative for implementing machine learning solutions in the delicate and changing contexts of smart cities because of these privacy and security constraints.

Federated Learning presents problems in addition to its many appealing benefits. Non-trivial problems include coordinating model updates from a variety of devices, handling heterogeneous data sources, and controlling communication overhead. Through breakthroughs like federated optimization algorithms, model compression approaches, and effective communication protocols, researchers and practitioners are actively attempting to overcome these difficulties [7]. Federated Learning will be more effective, scalable, and suitable for a larger range of smart city use cases thanks to these developments. Federated Learning is becoming more and more popular in smart cities, as seen by practical uses. Decentralized machine learning is helping smart cities anticipate traffic patterns, optimize waste management, and improve public safety. Success tales emphasize decreased latency, increased prediction accuracy, and above all the protection of citizen privacy. Federated Learning is positioned to be a key player in creating data-driven and privacy-preserving urban ecosystems as smart city programmes continue to grow [8].

Federated Learning has a bright future in smart city applications. Federated Learning is anticipated to become an essential component in the creation and implementation of intelligent systems as technology develops and the need for privacy-preserving solutions increases. The responsible and ethical deployment of Federated Learning in smart city contexts will depend heavily on the cooperation of academics, business, and government in tackling the remaining obstacles. Federated Learning stands out as a critical enabler as cities work to become more interconnected, effective, and sustainable [9]. It provides a means of using the combined intelligence of dispersed data sources while preserving people's security and privacy in urban areas. Smart cities are metropolitan regions that use data and technology to improve the effectiveness and standard of urban living. Cities must manage intricate systems like transit as the globe grows more urbanized in order to provide smooth movement for its citizens. Improving overall urban mobility, streamlining transport networks, and lowering congestion all depend heavily on real-time traffic forecast. The dynamic nature of urban areas presents challenges for traditional approaches of traffic prediction. Using cutting-edge technology like Federated Learning has become a viable strategy to solve these issues.

For efficient urban planning and administration, traffic prediction is essential. It makes it possible for local government officials to improve public transport services, optimize traffic signal timings, and proactively handle traffic congestion. Individual commuters can also benefit from real-time traffic data, which can assist them in making well-informed decisions regarding their travel schedules and routes. Predictive models allow smart cities to be proactive in addressing traffic problems, lessening their negative effects on the environment, and improving the general quality of life for their citizens [10]. By using a decentralized machine learning technique called federated learning, models may be trained on many servers or devices without requiring the exchange of raw data. Federated Learning is an innovative approach for privacy and data security in the context of smart cities. Without jeopardizing personal privacy, traffic data from several sources including sensors, GPS units, and cameras can be used to train models. Through collaboration, heterogeneous statistics from various sections of the city are leveraged to create strong and reliable traffic forecast models.

Federated Learning has many benefits, but there are drawbacks as well. It takes significant planning to coordinate and aggregate models from many places while upholding data security and privacy. Federated Learning implementation in real-time traffic prediction also requires eliminating potential biases and guaranteeing the interoperability of heterogeneous datasets. But there are also a lot of potential since this strategy enables cities to use the combined wisdom of disparate data sources to create traffic forecast models that are more precise and flexible [11]. Several smart cities worldwide have already started exploring the potential of Federated Learning for real-time traffic prediction. Case studies illustrate the successful implementation of this technology, showcasing improvements in traffic flow, reduced congestion, and enhanced transportation services. These examples serve as inspiration for other urban centers seeking innovative solutions to address their unique traffic management challenges. As smart cities continue to evolve, the integration of Federated Learning for real-time traffic prediction holds great promise [12]. The future implications extend beyond traffic management to contribute to a broader vision of sustainable and intelligent urban living. Cities can address existing transportation difficulties and provide the groundwork for a more connected, efficient, and resilient urban future by embracing cutting-edge technology like Federated Learning. The success of smart cities is largely determined by the cooperation of technology, data science, and urban planning; a crucial element of this revolutionary process is real-time traffic forecast.

The research questions for utilizing federated learning for enhanced real-time traffic prediction in smart urban environments may include:

- How can federated learning algorithms be adapted or optimized to effectively leverage distributed data sources for real-time traffic prediction in dynamic urban environments?
- What strategies can be employed to address privacy concerns while aggregating and learning from decentralized traffic data in federated learning settings?
- How can federated learning models be integrated with existing traffic prediction systems to enhance accuracy and reliability in smart urban environments?

The research objectives could be:

- To investigate and analyze existing federated learning algorithms and assess their suitability for real-time traffic prediction tasks.
- To develop novel techniques for privacy-preserving federated learning in the context of traffic prediction, ensuring compliance with regulatory standards and user privacy preferences.
- To design and implement a federated learning framework that integrates seamlessly with urban traffic monitoring systems, enabling real-time data exchange and model training across distributed nodes.

The research significance lies in its potential to revolutionize traffic prediction systems in smart urban environments by leveraging federated learning techniques. By addressing privacy concerns and enabling decentralized model training, this research could pave the way for more accurate, efficient, and scalable traffic prediction systems that can adapt to the dynamic nature of urban environments. Furthermore, the outcomes of this research could have broader implications for the development of federated learning applications in other domains, contributing to advancements in decentralized machine learning and data privacy preservation.

The key contributions of the paper are given as follows:

- The study presents unique hybrid architecture for real-time traffic prediction in smart cities that combines CNN and RNN. This combination of techniques incorporates the best aspects of temporal analysis from RNN, which identifies sequential patterns and relationships in traffic data with spatial evaluation from CNN, which identifies high-level characteristics from static traffic images.
- In contrast to traditional models, the suggested method combines geographical and temporal components for traffic prediction. While the CNN module retrieves static geographical information to provide an extensive understanding of traffic patterns, the RNN component methodically examines sequential data, taking into account unpredictable shifts in traffic circumstances over time.
- The study proposes a paradigm that improves privacy and efficiency by utilizing the concepts of federated learning. The federated learning strategy guarantees decentralized learning, protecting the privacy of individual sources of information while jointly enhancing the global model, by permitting local training on customer devices with locally produced data.
- To maximize the model's performance, the hybrid model uses backpropagation-based training and weight modification techniques. In the end, this increases the accuracy of real-time traffic forecasts by allowing the network to adjust to complicated temporal correlations and geographical patterns in traffic information.
- The suggested architecture is mostly used in the field of smart city traffic management. The model enhances road safety in urban contexts, reduces congestion, and improves traffic flow efficiency by improving real-time traffic forecast skills. The suggested model stands out as a complete response to traffic issues in smart cities due to its holistic approach, which takes into account both static and dynamic elements.

The arrangement of the remaining content is as follows. The traffic prediction literature is illustrated in Section II. The Problem Statement is provided in Section III. The suggested method for predicting traffic in real-time in smart cities is discussed in Section IV. In Section V, the method's performance is compared to earlier approaches, and the performance measurements are illustrated along with a summary of the findings. The conclusion and future works is summarized in Section VI.

II. Related Works

The widespread use of Internet of Things (IoT) sensors and devices, in conjunction with artificial intelligence, has resulted in the creation of "smart environments [13]." However, these solutions have high latency conditions and more information transmission from a network standpoint. Accordingly, this paper suggests a Federated Learning structure for Real-Time Traffic Calculation, which is supported by Roadside Units for simulation aggregation. The solution envisions learning being done on clients with locally generated information, and fully dispersed on the Edge, with outstanding learning rates, low latency, and less bandwidth consumption. To that end, this paper addresses tools and necessities for FL implementation towards asynchronous traffic estimation, as well as how such an approach could be assessed employing VANET and network simulations. The study first provides a preliminary assessment of a learning model on a group of automobiles that exemplify a distributed learning technique as a practical step. The study intends to employ a distributed technique like to this one in our proposed design. It is necessary to talk about the suggested solution's suitability in situations when vehicular ad hoc networks aren't present. The study should examine the approach's flexibility given that not all places may have a substantial VANET infrastructure.

Since federated learning is decentralized and protects data sources' privacy, it is commonly used in traffic forecasting employment requiring large-scale IoT-enabled sensor data. The current FL frameworks face significant overhead in communication when transmitting changes to parameter values for state-of-the-art deep learning-derived traffic indicators in FL systems, as the models' extensive and deep modelling necessitates the incorporation of a large number of parameters. To address this issue, we provide in this paper a workable FL scheme: Clustering-based modular and Two-step-optimized FL [14]. The suggested plan uses a divide et impera technique to categorize the clients according to how comparable the parameters of their local models are. Researchers include the particle swarm optimization technique and develop a two-phase method for optimizing local models. This technique lowers the communication cost of the model update transmission in FL by allowing just one representative local model upgrade from each cluster to be published to the central server. In order for the gradient compression or sparsification-based techniques to coordinate and minimize communication cost, CTFed is perpendicular to them. Comprehensive case studies utilizing three real-world sets of information and three cutting-edge models show how the suggested approach excels in terms of training effectiveness, precise forecasting performance, and resilience to unstable network settings. The suggested scheme's scalability, however, could have drawbacks. In large-scale IoT-enabled networks of sensors, in particular, the number of clients can lead to computationally demanding clustering and optimization stages that compromise scalability.

Recent developments in cloud computing, which offer near real-time processing along with storage scalability, have accelerated the development of data-intensive smart city applications. Millions of people rely on centralized, effective route planning systems like Google Maps as a result of this. Algorithms for route planning have advanced along with the cloud settings in which they operate [15]. As current state-of-the-art solutions are predicated on a shared memory paradigm, their deployment is constrained to data center multiprocessing scenarios. As a result of centralizing these functions, latency is becoming the limiting factor for emerging technology like driverless vehicles. These services also need connectivity to external networks, which raises questions about availability in the event of a disaster. As a result, this study offers a decentralized method for commercial fog network route planning. The study uses a hypothetical case study from a mid-sized American city to explore our method of cooperatively learning shared prediction models online, using recent breakthroughs in federated learning. However, a number of variables, such as network congestion, device malfunctions, and environmental circumstances, might affect the dependability and accessibility of private fog networks. The article ought to go into how the suggested strategy handles these kinds of obstacles.

Intelligent transportation systems, particularly in urban settings, are undergoing a transformation as a result of the Internet of Things' exponential expansion [16]. Transport network intelligence and efficiency are improved by an Intelligent Transportation System by utilizing data analytics and communication technology innovations. These IoT-enabled ITSs also produce a large amount of complicated data that is categorized as Big Data. Due to the enormous volume, velocity, diversity, and serious data privacy problems associated with Big Data, traditional information analysis frameworks require assistance in order to analyze it effectively. Federated Learning, which is well-known for protecting privacy, is a promising technique that may be used in ITSs to handle Big Data created by IoT devices. However, the variety of data, the varying nature of devices, and the dynamic environment in which ITS functions provide difficulties for the system. The concrete selection of an averaged technique during the server's aggregation phase and the practical training of dynamic clients are the main areas of recent endeavour to address these difficulties. The research that is now available, notwithstanding these efforts, still depends on customized FL with customized averages and customer education. In this study, a tailored architecture utilizing FL for effective and real-time large-scale data analysis in IoT-enabled ITSs is presented, along with an efficient Federated Averaging technique. The conventional averaging process is improved by applying a variety of customizing techniques. Weighted averaging and local fine-tuning adapt the global algorithm to the unique client data. Further performance improvement is achieved by using custom learning rates. To keep the model's efficacy intact, regular assessments are recommended. The suggested architecture provides a complete solution for contemporary urban transportation systems utilizing Big Data, addressing important issues including actual existence federated environment applications integrating information, and substantial data protection. The study implements the suggested methods for vehicle detection on the Udacity Self-Driving Car Database to show our model's effectiveness. The empirical findings confirm the architecture's superiority in terms of data privacy protection, instantaneous decision-making capabilities, and scalability.

Since sharing confidential information puts people's lives in danger, privacy concerns are seen as one of the biggest obstacles in smart cities. Federated learning has shown to be a successful method for both protecting privacy and optimizing data usage [17]. However, the amount of identifiable information acquired in smart cities is limited, while the amount of unlabelled data produced is abundant; this makes the use of semi-supervised learning necessary. We suggest FedSem, a semi-supervised collaborative learning technique that makes use of unlabelled data. The technique is split into two stages, the first of which uses the labelled data to train a global model. To enhance the model in the second stage, the study employs semi-supervised learning depending on the pseudo labelling approach. Utilizing the traffic sign dataset, the study ran a number of tests to demonstrate how FedSem may use unlabelled information to enhance accuracy throughout the procedure of learning by a maximum of eight percent.

Over the next several decades, Artificial Intelligence will revolutionize many aspects of our lives and careers, from face recognition to autonomous driving. Current AI methods for urban computing face several obstacles, such as managing the processing and synchronization of massive amounts of data created by edge devices and protecting user privacy and security, including biometrics, geolocation, and itinerary data [18]. Conventional centralized-based methods need uploading all organizational data to a single database, which may be against the law according to data protection laws like the CCPA and GDPR. Federated Learning, a novel training paradigm, is suggested as a way to separate model training from the requirement to keep the data on the cloud. With FL, the danger of privacy leakage may be greatly reduced as several devices can work together to jointly build a common framework while retaining the training data locally on each device. However, data in urban computing situations are frequently asynchronous, high-frequent, and communication-heavy, which presents additional difficulties for the implementation of FL. The study suggests StarFL, a novel hybrid federated learning architecture, as a solution to these problems. Secure key distribution, encryption, and decryption are made possible by StarFL in conjunction. Additionally, StarFL offers a verification method for every participant to guarantee the confidentiality of the local data. Furthermore, StarFL can offer precise timestamp matching to make it easier for several clients to synchronize. With all of these enhancements, StarFL is now more suitable for use in security-sensitive circumstances in the upcoming urban computing age.

Through decentralized training initiatives, federated learning has already been utilized for a variety of activities in automated transportation systems to preserve data privacy [19]. When it comes to learning spatial information, most of the most sophisticated approaches in automated transportation systems depend on graph neural networks. The present architectures for federated learning in ITS activities utilizing GNN-based models are limited to safeguarding data privacy, and they fail to consider the topological data related to transportation systems. To address this issue, the study presents a unique architecture for federated learning in this study. To safeguard the topological information, the study specifically presents an adjacency matrix preservation method that employs differential privacy. Additionally, the study suggests using an adjacency matrix aggregation technique to provide local GNN-based models access to the global network for improved training outcomes. Additionally, the study suggests the attention-based spatial-temporal graph neural networks model for traffic speed forecasting, which is based on GNNs. For traffic speed forecasting, we combine ASTGNN as FASTGNN with the suggested federated learning system. Numerous case studies using an actual dataset show that FASTGNN is capable of producing precise forecasts while adhering to the privacy preservation requirement.

The fields of real-time traffic estimates and smart city applications have already investigated a number of approaches, such as graph neural networks, semi-supervised learning, and federated learning. Federated learning has proven successful in maintaining privacy, whereas FedSem and other semi-supervised learning techniques have tackled the difficulties associated with using unlabelled data. Transportation systems have used graph neural networks to extract topological information. But these methods frequently run into problems, such scalability problems in large-scale IoT networks, communication overhead, and dependence on specialized federated learning methods. Furthermore, difficulties with data synchronization, privacy issues, and processing needs have been identified. In order to overcome these drawbacks, the research suggests a hybrid CNN-RNN approach to real-time traffic prediction. This model seeks to improve precision as well as effectiveness in smart city traffic management by utilizing the advantages of both spatial as well as temporal analysis.

The proposed utilization of FL for real-time traffic prediction in smart urban environments holds significant promise, addressing issues of privacy preservation and scalability inherent in centralized models. However, several limitations and challenges need to be considered. Firstly, the scalability of FL frameworks, particularly in large-scale IoT-enabled networks, may be compromised due to computational demands during clustering and optimization stages. Additionally, while FL offers privacy protection, the decentralized nature of urban computing environments presents asynchronous, high-frequency, and communication-heavy data, posing challenges for FL implementation. Moreover, existing FL architectures may not adequately address topological data related to transportation systems, necessitating innovative approaches for preserving such information while ensuring privacy. Furthermore, the effectiveness of FL techniques may be impacted by variables such as network congestion, device malfunctions, and environmental circumstances, which could affect the reliability and accessibility of FL-based traffic prediction systems. Thus, future research efforts should focus on addressing these limitations to realize the full potential of FL in enhancing real-time traffic prediction in smart urban environments. Table I shows the advantages and disadvantages of existing methods.

TABLE I. ADVANTAGES AND LIMITATIONS OF EXISTING APPROACHES

Authors	Methods	Advantages	Limitations
M. V. S. da Silva, L. F. Bittencourt, and A. R. Rivera,	Utilizes FL for real-time traffic prediction, supported by Roadside Units for simulation aggregation. Learning done on clients with locally generated data, dispersed on the Edge, with low latency and less bandwidth consumption.	Decentralized model training enhances privacy protection. Scalability due to distributed nodes.	Computational demands in large-scale networks may compromise scalability. Challenges with data synchronization and communication overhead.
C. Zhang, L. Cui, S. Yu, and J. J. Q. Yu,	Uses divide et impera technique to categorize clients based on local model parameters. Incorporates particle swarm optimization and two-phase optimization for local models, reducing communication costs.	Reduction in communication overhead. Effective training and precise forecasting performance. Resilience to unstable network settings.	Computational demands during clustering and optimization may affect scalability.
M. Wilbur, C. Samal, J. P. Talusan, K. Yasumoto, and A. Dubey	Hybrid federated learning architecture with secure key distribution, encryption, and decryption. Offers verification method for data confidentiality and precise timestamp matching for synchronization.	Enhanced security and privacy protection. Precise timestamp matching improves data synchronization.	Complexity in implementation and management. Potential performance overhead due to encryption and decryption processes.
S. Kaleem, A. Sohail, M. U. Tariq, and M. Asim,	Tailored architecture for real-time data analysis in IoT-enabled ITSs. Improves conventional averaging process with customizing techniques like weighted averaging, local fine-tuning, and custom learning rates.	Improved data privacy protection and model effectiveness. Scalability with efficient customization techniques.	Complexity in customization and management. Potential performance overhead due to customization processes.
A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha,	Utilizes both spatial and temporal analysis for real-time traffic prediction. Combines advantages of CNNs and RNNs.	Improved precision and effectiveness in traffic management. Enhanced capability for spatial and temporal analysis.	Potential complexity in model architecture and training process. Dependency on accurate data synchronization and integration of spatial-temporal features.

III. Problem Statement

Prior research in the field of smart city applications has examined a number of strategies, including semi-supervised learning, and graph neural networks, with a focus on real-time traffic estimates. Although federated learning has shown promise in protecting privacy and maximizing data use, large-scale IoT-enabled networks may provide difficulties for current models due to high communication cost and scalability constraints. Furthermore, other research has addressed data security, synchronization, and topological data preservation difficulties. These initiatives, however, frequently have their own set of drawbacks, including high processing requirements, dependence on tailored federated learning strategies, and challenges managing asynchronous and highly communicative urban computing settings [20]. This paper suggests unique hybrid CNN-RNN architecture for real-time traffic forecasting in smart cities in light of these factors. By combining the advantages of recurrent neural networks for temporal evaluation and convolutional neural networks for spatial extraction of characteristics, the suggested model seeks to address the noted limitations and offer an efficient method for traffic prediction that takes into account both the static and dynamic aspects of the data.

IV. Federated Learning for Real-Time Smart City Traffic Forecasting

In order to develop hybrid CNN-RNN architecture for real-time traffic prediction, two steps are involved in the proposed approach: first, data collecting and pre-processing. A systematically low-resolution dataset of 1.6 GB was created by methodically gathering real-time CCTV traffic camera images from important crossroads in the cities of Guntur and Vijayawada under a variety of situations. The dataset was pre-processed, using min-max normalization to normalize pixel values, to improve usability. Next, the suggested hybrid model combines RNN to capture temporal elements of traffic data with CNN for extracting geographical features. The RNN uses recurrent interactions to preserve hidden states for sequential information processing. The CNN uses static traffic images to extract high-level spatial information. By utilizing weight adjustment and backpropagation-based training, this hybrid CNN-RNN architecture intends to enhance traffic management in smart cities by improving real-time traffic prediction while taking into account both static and dynamic features. Fig. 1 shows the overall architecture of the proposed approach.

A. Data Collection

In order to create the dataset for our study, "Leveraging Federated Learning for Real-Time Traffic Prediction in Smart Cities," real-time CCTV traffic camera images were systematically gathered. Our study was aimed at capturing the dynamics of traffic flow in the well-known towns of Guntur and Vijayawada. Particularly, information was gathered at important crossroads in Guntur City, such as Brundhavan Gardens and Guntur Market, as well as in Vijayawada at Benz Circle, Seetharamapuram, Guru Nanak Colony, and Ramavarappadu Junction. The study measured the time it took for a vehicle to travel from one intersection to the other endpoint for each junction, taking into account four different signal points. The network of eighty-three cameras that have been strategically positioned around the cities to provide an accurate representation of various traffic situations is included in the dataset. The study collected 1.6 GB of information in total for our trials. The images in the collection are intentionally low-resolution, having been taken in a variety of lighting situations, perspectives, and locations. Every image has a fixed dimension of 800 pixels for width and 600 pixels for height. The goal of this dataset is to improve the effectiveness of traffic management in smart cities by facilitating the study of real-time traffic forecasting algorithms within the context of federated learning [21].

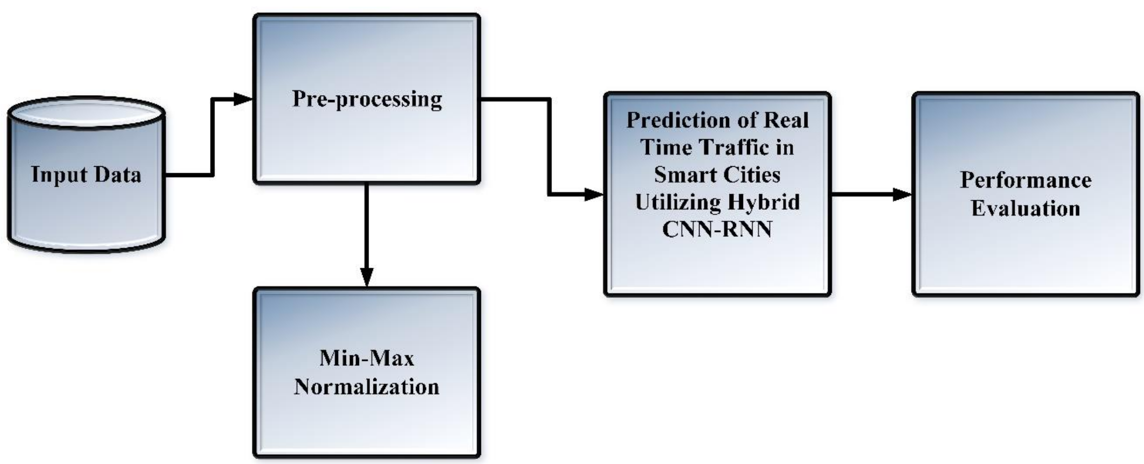

Fig. 1. Overall architecture of the proposed approach.

B. Pre-processing Employing Min-Max Normalization

In advance of employing the CCTV image collection for any analytic or deep learning tasks, pre-processing is required to make it more usable. The collection, which was gathered by traffic cameras, consists of 1.6 GB of indistinct images that have been taken under different lighting conditions. The width and height of every image are 800 and 600 pixels, respectively. Given the heterogeneity of the images in terms of angles, light levels, and climatic conditions, pre-processing is necessary to preserve consistency and compatibility for the operations that follow. The pixel values must be scaled, normalized, and min-maxed in order to bring them into a common range for effective assessment.

Feature scaling, also known as the min-max normalization process, is a crucial step in the pre-processing of images. This technique involves rescaling the image's pixel values such that they fall into a certain range, usually [0, 1]. This uniformness the intensity levels across all images, lessening the effect of variations in illumination and pixel dispersion of values. The two main steps in the min-max normalization procedure are determining the maximum and lowest values for each pixel in the dataset and then using a linear transformation to change each pixel's original value to one that matches inside the specified range. This process helps to produce more accurate and consistent results in machine learning and subsequent assessments. It also improves image consistency. Before the dataset is subjected to min-max normalization, the minimum and highest pixel values for each picture in the dataset are first established. The least significance in the dataset denotes the lowest pixel, while the highest value indicates the most lighted pixel. After these values have been determined, each pixel's initial intensity value is linearly altered to a different value that falls inside the [0, 1] range. Eq. (1) gives the formula that was applied to this conversion.

$$P_{out} = (P_{in} - Min)\frac{newMax - newMin}{Max - Min} + newMin \qquad (1)$$

Following min-max normalization, the resulting image is calledP_{out}, and the new minimum and maximal intensities are called $NewMin$ and $newMax$. P_{in} represents the initial real-time traffic image; Min and Max, respectively, represent the lowest and highest intensity standards, which extend from 0 to 255. This alteration was applied to every image in the dataset, ensuring that the pixel values were consistent and suitable for additional processing. By eliminating biases resulting from variances in pixel values and illumination, this normalization approach helps to improve the dataset's suitability for accurate and consistent evaluation.

C. Prediction of Real Time Traffic in Smart Cities Utilizing Hybrid CNN-RNN

Feature extraction, a crucial stage in computer vision applications, uses convolutional neural networks to capitalize on the unique qualities of the network, such as weight sharing and local connection. CNNs comprise of layers of convolution that adjust to express unique qualities within input images and pooling layers that integrate shift consistency. When interpreting an input image's characteristic, the CNN demonstrates exceptional characteristics including weight sharing and local connectivity to the neurons. The layers of CNN are the pooling layer, which ensures shift invariance, and the convolutional layer, which grows to reflect the unique qualities of the input image.

The nearest group of neurons in the layer before the resultant layer will supply input to the convolutional layer's neurons. The different unique representations were produced by combining many kernels from the preceding layer. Eq. (2) is used to build the convolution layer.

$$v_d^j = \sigma\left(\sum_{i=1}^{d_{j-1}} v_l^{j-1}, M1_{ld}^j + b1_d^j\right), d\,\epsilon[1, d_1] \qquad (2)$$

The $(j-1)^{th}$ layer's l^{th} activation map is denoted byv_l^{j-1}, the j^{th} convolution layer'sd^{th} activation mapping is suggested byv_d^j, and the weight connecting the d^{th} layer's lth activation map at position can be determined by $M1_{ld}^j$ and $b1_d^j$. The different filters in the d^{th}layer may be described by both l_1 and the elementwise exponential activation function.

Although the pooling procedures possess the required information, they might reduce the activation map's spatial dimension. Eq. (3) yields $v_f^j(x, y)$ when the output of the previous layer is handled bitwise nonlinear activation and curled with the dimensions (p, q) in the convolution filter. The positions of the kernel are a1 and b1.

$$v_d^j(x, y) = \sigma\left(\sum_{l=1}^{d_{j=1}} \sum_{a1=0}^{q=1} \sum_{b1=0}^{q=1}\left(M1_{ld}^j(a1, b1) \otimes v_l^{j-1}(x + a1, k + b1) + b1_d^j\right)\right), d\,\epsilon[1, d_1] \qquad (3)$$

The convolution layer was supervened by the location of the d^{th}activation map of the $j+1^{th}$ pooling layer by obtaining the results of the previous layer with a filter of size (2, 2). This resulted in the production of$v_l^{j+1}(x, y)$, which was then used to perform bitwise nonlinear activation using Eq. (4).

$$v_d^{j+1}(x, y) = \sigma\left(\sum_{l=1}^{d_{j=1}} \sum_{a1=0}^{q=1} \sum_{b1=0}^{q=1}\left(M1_{ld}^{j+1}(a1, b1) \otimes v_l^j(2x + a1, k + b1) + b1_d^{j+1}\right)\right), d\,\epsilon\left[1, d_{j+1}\right] \qquad (4)$$

Enhancing efficacy and safety in real-time traffic estimation has been demonstrated through the application of federated learning approaches. Following the crucial step of using convolutional neural networks to extract characteristics, RNNs are utilized to assess the temporal aspects of traffic data. RNNs are an effective solution since they perform well in tasks involving sequential data when prediction and modelling traffic patterns over time. The CNN's output, which typically consists of high-level features and spatial representations taken from static traffic images, serves as the RNN's input. This input includes crucial information about the current traffic flow, including vehicle locations, concentrations, and movement patterns. But traffic conditions are dynamic and ever-changing by nature. To effectively manage traffic and make choices, it is important to consider the temporal linkages and sequential nature of traffic information.

RNNs are efficient at processing the sequential data because of their recurrent interactions, which allow them to maintain a hidden state that accumulates data from earlier time phases. Because of this hidden state, which acts as a memory, the network may retain and utilize data from earlier traffic measurements. The RNN systematically analyses the incoming input at each time step, updating its hidden state and producing output predictions in the process. This method enables RNNs to identify complex temporal correlations and patterns in the traffic information, such as variations in traffic flow and congestion and recurring patterns at specific times of the day. The hidden state at time step t, denoted asrr_t, is found by applying Eq. (5) to the previously calculated hidden state, r_{t-1} and the current input,y_t.

$$r_t = d(M_{ir}y_t + M_{rr}r_{t-1} + b_r) \tag{5}$$

The weight matrices in this instance are M_{rr}andM_{ir}, the bias terma_r, and the activation function d, which is usually a reconditioning linear unit (ReLU) function or a hyperbolic tangential (tanh) function. The output at time step t, or asy_t, is given by Eq. (6) and is generated based on the hidden state that is in effect at that moment.

$$x_t = h(M_{r0}r_t + b_0) \tag{6}$$

where, M_{r0} is the weight matrix and b_0 is the bias factors for the resulting layer. RNNs employ recurrent links to store information from previous stages of time. The hidden state r_t is found using the current input y_t and the hidden states r_{t-1} which appeared before it. The network may be trained and its weights and biases adjusted by using backpropagation across time. This enables the network to identify and respond to temporal trends in the traffic data. In smart cities, traffic management solutions may leverage RNNs' capacity to anticipate traffic in real time, improving flow, reducing congestion, and enhancing road safety. In the context of smart city applications, effective traffic management is essential to maintaining vehicular flow and improving urban mobility as a whole. This research suggests a hybrid strategy integrating CNN and RNN for addressing the problems related to real-time traffic estimation. First, from static traffic images, features are extracted using CNNs, which are particularly good at extracting high-level characteristics and spatial representations. Convolutional and pooling layers make up the CNN layers, which use weight sharing and local connection to identify distinctive features in input pictures. The usage of RNNs is then extended to evaluate the temporal dimensions of traffic data, taking into consideration the dynamic and ever-changing characteristics of traffic situations. RNNs are efficient at processing sequential data, which enables them to model and forecast traffic patterns in the long run. The hybrid model can identify both temporal and geographic correlations in the traffic information because the CNN's output, which represents high-level spatial information, is introduced into the RNN. This combined CNN-RNN architecture shows how to estimate traffic in real time while taking into account both the static and dynamic aspects of the data. Recurrent interactions are used by the hybrid model's RNN component to preserve a hidden state that gathers data from previous time periods. Fig. 2 shows the architecture of the hybrid CNN-RNN.

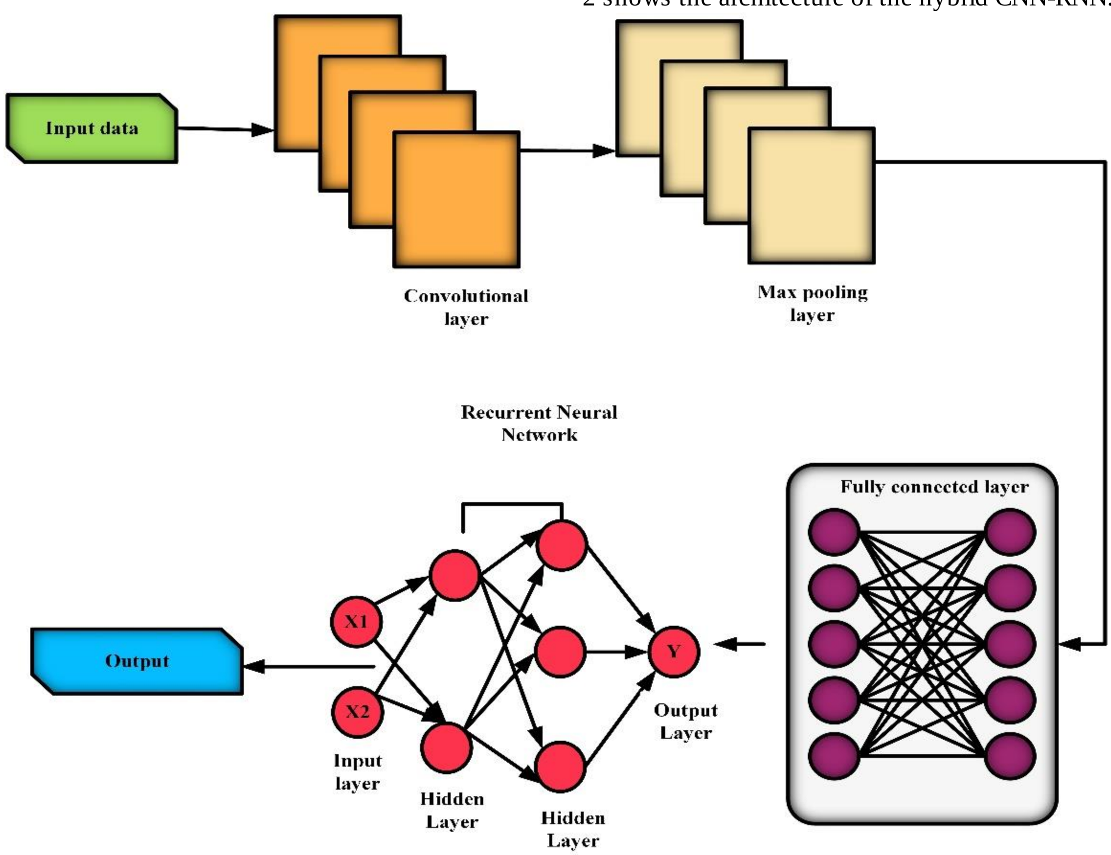

Fig. 2. Architecture of Hybrid CNN-RNN.

By serving as an instance of memory, this concealed state helps the network store and use information from earlier traffic measurements. The model is an effective tool for real-time traffic prediction because of its capacity to recognize intricate temporal correlations and patterns in traffic data, such as fluctuations in congestion and recurrent traffic behaviours at particular times of the day. The hybrid CNN-RNN architecture has tremendous potential for improving traffic management systems in smart cities, helping to improve traffic flow, reduce congestion, and increase road safety through training and correction of weights and biases utilizing backpropagation over time. The pseudocode for the proposed approach is given below.

Pseudocode: Proposed Federated Learning Approach

```
Input: Raw traffic camera images from Guntur and Vijayawada crossroads
// Data Collection and Pre-processing
raw_images = collect_traffic_images('Guntur', 'Vijayawada', 'crossroads')
normalized_dataset = preprocess_images(raw_images, resolution, normalization='min-max')
// Define Hybrid CNN-RNN Model
model = create_hybrid_model(image_shape=(resolution, resolution, channels), sequence_length, num_features)
Output: Trained hybrid CNN-RNN model for real-time traffic prediction
```

V. Results and Discussion

The findings and analysis of the suggested hybrid CNN-RNN architecture for real-time traffic prediction are covered in detail in this portion of the article. The performance of the hybrid model is assessed after the methodical collecting and preprocessing of a 1.6 GB low-resolution dataset from significant crossroads in Guntur and Vijayawada. To improve usability, min-max normalization was applied to the dataset. The architecture is then evaluated, fusing CNN's spatial extraction characteristics with RNN's temporal capture capabilities. While the CNN retrieves high-level spatial information from stationary traffic images, the RNN, which is outfitted with recurrent interactions for sequential processing of information, detects temporal subtleties in traffic patterns. The model is more adaptive to intricate spatial and temporal correlations in traffic data when weight adjustment and backpropagation-based training are used. The usefulness of the suggested model in enhancing real-time traffic forecast, taking into account both static and dynamic variables, is the main topic of the talks that follow. The investigation aims to assess the model's contributions to increased road safety, reduced congestion, and enhanced traffic flow. It covers the model's accuracy, effectiveness, and practical applications for smart city traffic management.

A. *Performance Metrics*

Performance metrics are essential for assessing the efficacy of models and algorithms because they offer quantitative measures to evaluate their predictive accuracy and reliability. For the purposes of this paper, the performance metrics that were selected are MAPE, MSE, MAE, and RMSE. By calculating the percentage disparity between predicted and actual values, MAPE is used to assess prediction accuracy and is particularly useful for evaluating forecast accuracy in real-time traffic predictions. MSE, on the other hand, measures the average squared difference between predicted and actual values, which highlights the model's error-minimizing capabilities. Finally, MAE determines the average absolute differences between predicted and actual values, providing a reliable indicator of prediction accuracy. With the extra advantage of declaring outcomes in the same units as the original data, RMSE, like MSE, emphasizes the model's success in minimizing mistakes. The purpose of the paper is to provide a thorough evaluation of the suggested federated learning-based traffic prediction model in the dynamic framework of smart city traffic management by employing this suite of performance metrics.

1) Mean Squared Error (MSE): Mean squared error, or MSE, is a common statistic used in machine learning to evaluate the performance of regression models. The method used to compute the MSE in the dataset is the average squared variance between the expected and actual values. Eq. (7) provides evidence for this.

$$MSE = \frac{1}{a}\sum_{i=1}^{a}\left(W_i - \widehat{W_i}\right)^2 \quad (7)$$

where, a is the total amount of information points, W_i is the original values and $\widehat{W_i}$ is the anticipated values.

2) Mean Absolute Error (MAE): The average amount of errors between anticipated and real outcomes is measured using a metric called mean absolute error in statistical and machine learning. It estimates the average of these absolute variations after measuring the percentage difference between every projected value and its matching real value. A predictive model's accuracy may be easily evaluated using MAE, with reduced MAE values representing greater predictive effectiveness. It is characterised by Eq. (8).

$$MAE = \frac{1}{a}\sum_{i=1}^{a}\left|W_i - \widehat{W_i}\right| \quad (8)$$

where, W_i denotes the actual values, $\widehat{W_i}$ denotes the predicted values, and a is the total number of information points.

3) Mean Absolute Percentage Error (MAPE): The average absolute percentage variance between the real and displayed values of a target variable is determined by the Mean Absolute Percentage Error, or MAPE, which is a commonly used statistic to evaluate the performance of regression algorithms in machine learning. Eq. (9) incorporates the MAPE equation.

$$MAPE = \frac{1}{a}\sum_{t=1}^{a}\left|\frac{W_t - \widehat{W_t}}{W_t}\right| \times 100\% \quad (9)$$

where, a denotes the sample size, W_t is the real value of the intended variable, and $\widehat{W_t}$ is the parameter's projected value.

4) Root Mean Square Error (RMSE): RMSE is a frequently used measure to assess how well regression techniques function. By considering the squared variances, it calculates the average variation between the expected and actual outcomes. Because it draws attention to greater differences, RMSE is especially helpful when errors are bigger and more significant. Its equation is given in Eq. (10) below.

$$RMSE = \sqrt{\sum_{j=1}^{A} \frac{\|W(j) - \widehat{W}(j)\|}{N}} \qquad (10)$$

The variable j is displayed here together with the actual observation time series $W(j)$, the anticipated observation time series $\widehat{W}(j)$, and the non-missing data points A.

The training and testing accuracy of a hybrid CNN-RNN model over several epochs is shown in Fig. 3. The training accuracy of the model gradually rises as it is trained over more epochs, demonstrating its capacity to pick up new skills and adjust to the dataset.

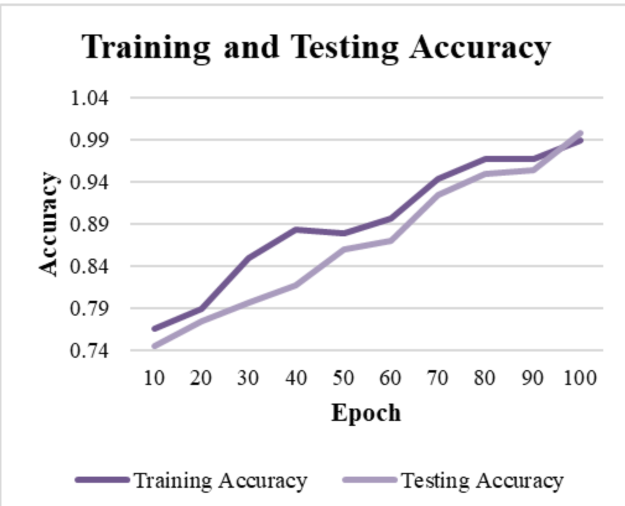

Fig. 3. Training and testing accuracy.

The training accuracy begins at 0.766 in the first epoch and increases steadily, hitting 0.99 by the 100th epoch, which shows a high level of skill in identifying the underlying patterns in the data. Concurrently, testing accuracy shows a similar rising trajectory when assessed on a different dataset to determine the model's capacity for generalization. The testing accuracy starts at 0.745 in the first epoch and steadily increases to an astounding 0.998 by the 100th epoch. The model's superior learning from training data and good generalization to unknown data is indicated by the convergence of training and testing accuracy towards later epochs, underscoring its usefulness in real-time traffic prediction. The general pattern indicates that the hybrid CNN-RNN model was successfully trained and validated, confirming its potential for use in improving traffic management systems in smart cities.

The training and testing loss values for the suggested hybrid CNN-RNN architecture for real-time traffic prediction are shown in Fig. 4 spanning several epochs. Both training and testing losses are somewhat substantial in the early epochs, which is an indication of the model's immaturity and its difficulty in correctly capturing the complex patterns present in the data. A pattern of declining loss values can be seen as the epochs go by, which emphasizes the model's ongoing development and learning from the training set. The training loss has dramatically decreased to 0.06 by the 100th epoch, demonstrating the model's effectiveness in reducing errors throughout the learning process. Additionally, the testing loss shows a significant decrease to 0.14, highlighting the model's capacity to generalize to previously untested data. This pattern demonstrates how well the suggested CNN-RNN architecture performs in terms of improving its predictive performance throughout subsequent epochs, providing a strong basis for precise real-time traffic forecasts in the context of smart cities. The model's improved capacity to capture temporal and spatial dynamics is indicated by the decreasing loss values, confirming its promise as a useful instrument for enhancing traffic management systems.

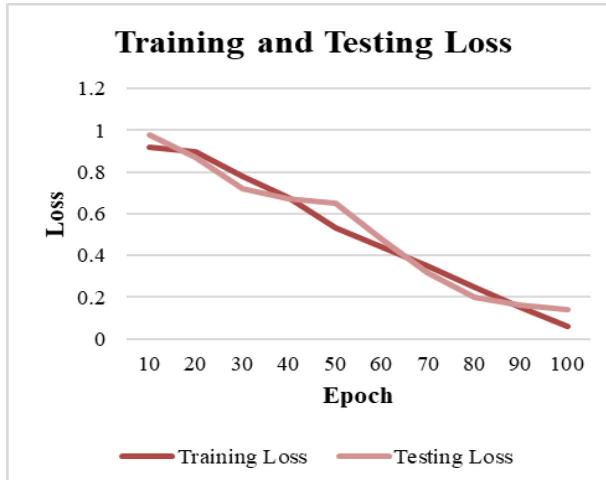

Fig. 4. Training and testing loss.

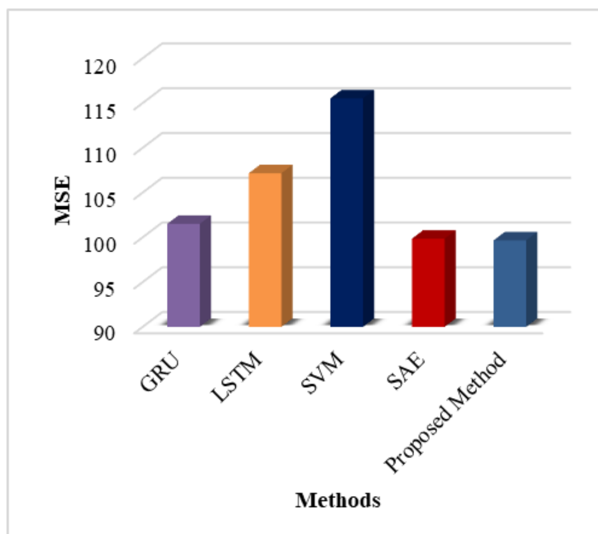

Fig. 5. Mean squared error.

The performance of several techniques for real-time traffic prediction using the Mean Squared Error (MSE) metric is shown in Fig. 5. The suggested approach is contrasted with four other approaches are LSTM (Long Short-Term Memory), SAE (Stacked Auto encoder), GRU (Gated Recurrent Unit), and SVM (Support Vector Machine). The average squared differences between the actual and anticipated traffic levels are represented by the MSE values, which are an essential measure of how well the models minimize prediction mistakes. MSE values that are lower are indicative of more accurate models. The suggested approach sticks out in this comparison thanks to its exceptionally low MSE of 99.66, which shows how well it minimizes squared prediction errors. With an MSE score of 99.85, SAE exhibits competitive performance as well. The MSE values of 101.5, 107.16, and 115.52 for GRU, LSTM, and SVM, respectively, are comparatively higher, indicating a lower efficacy of these algorithms in minimizing squared prediction errors. The suggested method's better performance in optimizing forecast accuracy is visually shown by the graph, which highlights its potential as an effective technique for real-time traffic prediction in the dynamic setting of smart cities.

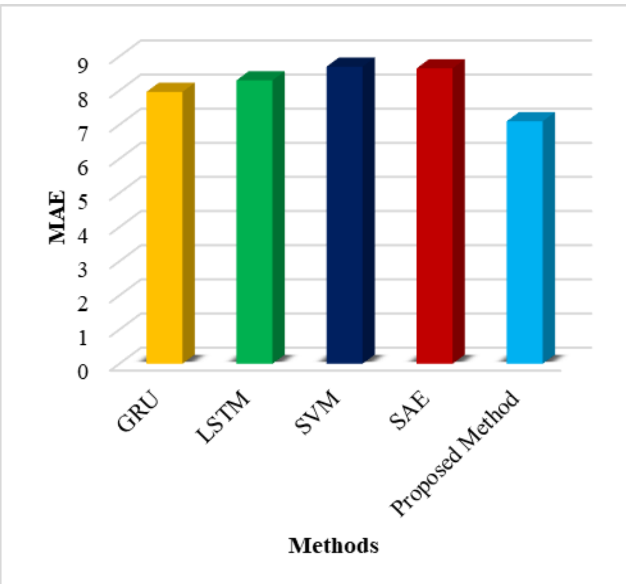

Fig. 6. Mean absolute error.

The MAE performance metrics for the several techniques used in real-time traffic forecast are shown in Fig. 6. Four other approaches are contrasted with the suggested approach are GRU, LSTM, SVM, and SAE. One important measure of prediction accuracy is the MAE between the actual and anticipated traffic levels. More accurate forecasts are suggested by lower MAE values. With a significantly lower MAE of 7.1 than the other models in this comparison, the proposed approach performs better, demonstrating its higher accuracy in real-time traffic estimates. GRU and SAE exhibit competitive performance as well, with respective MAE scores of 7.96 and 8.65. In contrast, the MAE values of 8.3 and 8.7 for LSTM and SVM are considerably higher, suggesting that their traffic projections are less accurate. The graph shows how well the suggested strategy performs in terms of decreasing prediction errors, and it shows that this approach has the potential to be a successful one for real-time traffic prediction in the setting of smart cities.

The MAPE for a variety of real-time traffic prediction techniques is shown in Fig. 7, which provides information on how well these models are in predicting traffic dynamics. Four other approaches are contrasted with the suggested approach are GRU, LSTM, SVM, and SAE. The average percentage difference between expected and actual traffic levels is a critical indicator for evaluating how well the models capture the degree of prediction mistakes. Models with higher accuracy are indicated by lower MAPE values. With a much lower MAPE of 17.23 than the other models in this comparison, the suggested strategy performs better, demonstrating its efficacy in reducing percentage mistakes in traffic forecasts. LSTM and SAE demonstrate competitive performance as well, with respective MAPE values of 20.32 and 19.72. In contrast, the MAPE values of GRU and SVM are considerably higher at 22.73 and 19.8, indicating a lesser level of accuracy in traffic dynamics prediction. The graph highlights the potential of the suggested method as a successful real-time traffic forecast tool in the context of smart cities by explicitly demonstrating its better performance in minimizing percentage mistakes.

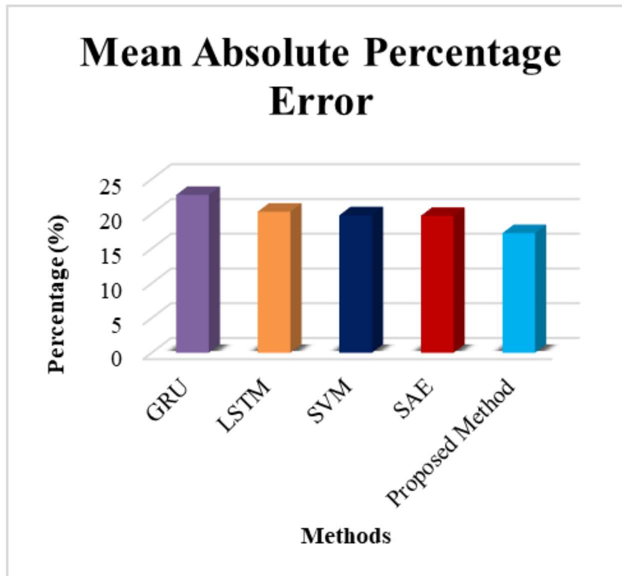

Fig. 7. Mean absolute percentage error.

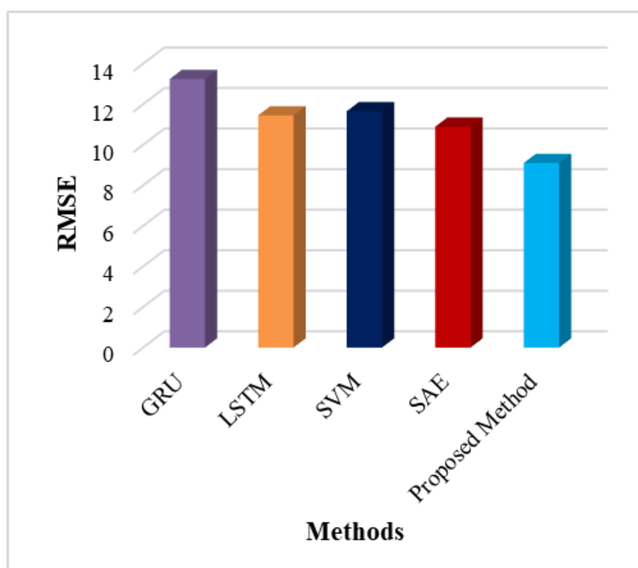

Fig. 8. Root mean square error.

The RMSE for many techniques used in real-time traffic prediction is shown in Fig. 8, which provides important information about how accurate these models are in predicting traffic dynamics. Four other approaches are contrasted with the suggested approach: GRU, LSTM, SVM, and SAE. The square root of the average squared discrepancies between the traffic values that were predicted, and the actual traffic values is represented by RMSE values, which provide a thorough assessment of how well the models minimize prediction mistakes. Models with higher accuracy are associated with lower RMSE values. With a relatively low RMSE of 9.1, the suggested technique stands out in this comparison and shows that it is successful in decreasing both squared and root-squared prediction errors. With an RMSE score of 10.89, SAE exhibits competitive performance as well. The comparatively higher RMSE values of 11.45, 11.65, and 13.24 for LSTM, SVM, and GRU, respectively, indicate that these techniques are less successful in lowering squared and root-squared prediction errors. The suggested method's improved performance in maximizing overall forecast accuracy is visually shown by the graph, which also highlights the method's potential as a useful strategy for real-time traffic prediction in the dynamic setting of smart cities.

The comparison of error metrics in Table II across different methods, including GRU, LSTM, SVM, SAE, and the proposed method, reveals that the proposed approach outperforms existing models in terms of mean absolute percentage error (MAPE), mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). Specifically, the proposed method achieves the lowest values across all error metrics, indicating its superior accuracy and effectiveness in real-time traffic prediction compared to the other models considered. The significant reduction in error metrics underscores the potential of the proposed method to provide more reliable and precise traffic forecasts, which is crucial for effective traffic management and urban planning in smart city environments.

TABLE II. COMPARISON OF ERROR METRICS

Methods	MAPE (%)	MAE (%)	MSE (%)	RMSE (%)
GRU	22.73	7.96	1.015	13.24
LSTM	20.32	8.3	1.0716	11.45
SVM	19.8	8.7	1.152	11.65
SAE	19.72	8.65	0.9985	10.89
Proposed Method	17.23	7.1	0.9966	9.1

B. Discussion

A thorough assessment of the suggested hybrid CNN-RNN architecture for real-time traffic prediction in smart cities can be found in the results and discussion section. After a 1.6 GB low-resolution dataset from important intersections in Guntur and Vijayawada was systematically collected and preprocessed, the hybrid model, which combines the spatial extraction skills of CNN with the temporal capture capabilities of RNN was assessed. The model's capacity to learn complex patterns and adapt effectively to new data is demonstrated by a gradual increase in testing and training accuracy over subsequent epochs. The performance metrics provide quantitative information on the correctness and dependability of the model. These measures include MSE, MAE, MAPE, and RMSE. Lower MSE, MAE, MAPE, and RMSE values highlight how well the recommended strategy performs in comparison to current approaches such as LSTM, SAE, GRU and SVM [21], highlighting its ability to reduce prediction errors and enhance real-time traffic predictions. The success of the hybrid model is ascribed to its ability to manage both static and dynamic elements, which improves traffic flow, lowers congestion, and increases road safety in smart city environments. The outcomes highlight the suggested architecture's potential as a useful instrument for improving traffic management systems and boosting the effectiveness of intelligent transportation networks.

Utilizing FL for enhanced real-time traffic prediction in smart urban environments offers several advantages compared to other methods and models in similar fields. Firstly, FL enables decentralized model training, allowing for the utilization of locally generated data on clients without the need for data centralization, thereby addressing privacy concerns associated with centralized approaches. This decentralized nature also enhances scalability, as FL can accommodate a large number of distributed nodes without significantly increasing computational overhead. Additionally, FL can adapt to dynamic urban environments by continuously learning from diverse data sources without the need for centralized retraining, ensuring that traffic prediction models remain up-to-date and accurate. Furthermore, FL facilitates efficient model aggregation and communication among distributed nodes, resulting in lower latency and reduced bandwidth consumption compared to traditional centralized approaches. Overall, FL represents a promising approach for real-time traffic prediction in smart urban environments, offering improved privacy, scalability, adaptability, and efficiency compared to alternative methods and models.

VI. CONCLUSION AND FUTURE WORKS

The study concludes by introducing hybrid CNN-RNN architecture and demonstrating how well it can capture temporal and geographical data for real-time traffic prediction in smart cities. Reduced MSE, MAE, MAPE, and RMSE values show that the suggested model, trained on a methodically gathered and preprocessed dataset, performs better than current techniques. The findings suggest that it has the ability to improve traffic flow in dynamic urban contexts, reduce congestion, and improve road safety. The research presents a new contribution by highlighting the model's effective integration of federated learning and highlighting its privacy-preserving characteristics. Subsequent research endeavours may go into additional refinement of federated learning parameters, evaluate the model's adaptability to more extensive datasets and varied urban environments, and examine its practical implementation. To improve the model's prediction skills in intricate urban settings, the suggested architecture may also be expanded to meet multimodal data sources, such as input from IoT sensors. In order to meet the changing needs of smart metropolitan transportation networks, the study establishes the groundwork for sophisticated traffic management systems that make use of cutting edge technology.

Future work in utilizing federated learning for enhanced real-time traffic prediction in smart urban environments could focus on several key aspects to provide a clearer roadmap for potential developments and advancements. Firstly, there's a need for research into refining federated learning algorithms to effectively handle the complexities of real-time traffic data, including heterogeneous data sources and dynamic urban environments. Secondly, exploring innovative techniques to enhance model aggregation and communication efficiency among distributed nodes without compromising privacy and security is essential. Additionally, investigating strategies to integrate federated learning with other emerging technologies such as edge computing and blockchain for improved scalability, reliability, and transparency could further enhance the efficacy of traffic prediction systems. Furthermore, conducting extensive real-world deployment studies and collaborations with city planners and transportation authorities to validate the practical viability and societal impact of federated learning-based traffic prediction solutions is crucial. Finally, addressing ethical and regulatory considerations surrounding data privacy, bias mitigation, and algorithmic transparency will be paramount for the successful adoption and deployment of such systems in smart urban environments.

REFERENCES

[1] S. Hu et al., "A Federated Learning-Based Framework for Ride-sourcing Traffic Demand Prediction," IEEE Transactions on Vehicular Technology, 2023.

[2] G. Badu-Marfo, B. Farooq, D. O. Mensah, and R. Al Mallah, "An ensemble federated learning framework for privacy-by-design mobility behaviour inference in smart cities," Sustainable Cities and Society, p. 104703, 2023.

[3] W. Wang et al., "Data information processing of traffic digital twins in smart cities using edge intelligent federation learning," Information Processing & Management, vol. 60, no. 2, p. 103171, 2023.

[4] Y. Djenouri, T. P. Michalak, and J. C.-W. Lin, "Federated deep learning for smart city edge-based applications," Future Generation Computer Systems, vol. 147, pp. 350–359, 2023.

[5] Y. Pang, Z. Ni, and X. Zhong, "Federated Learning for Crowd Counting in Smart Surveillance Systems," IEEE Internet of Things Journal, 2023.

[6] R. Valente, C. Senna, P. Rito, and S. Sargento, "Federated Learning Framework to Decentralize Mobility Forecasting in Smart Cities," in NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2023, pp. 1–5.

[7] P. Agarwal, S. Sharma, and P. Matta, "Federated Learning in Intelligent Traffic Management System," in 2023 Winter Summit on Smart Computing and Networks (WiSSCoN), IEEE, 2023, pp. 1–6.

[8] A. Soleimany, Y. Farhang, and A. Babazadeh Sangar, "Hierarchical federated learning model for traffic light management in future smart," International Journal of Nonlinear Analysis and Applications, 2023.

[9] M. Arya et al., "Intruder Detection in VANET Data Streams Using Federated Learning for Smart City Environments," Electronics, vol. 12, no. 4, p. 894, 2023.

[10] L. Liu et al., "Multilevel Federated Learning based Intelligent Traffic Flow Forecasting for Transportation Network Management," IEEE Transactions on Network and Service Management, 2023.

[11] Y. Goto, T. Matsumoto, H. Rizk, N. Yanai, and H. Yamaguchi, "Privacy-preserving taxi-demand prediction using federated learning," in 2023 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2023, pp. 297–302.

[12] C. Lanza, E. Angelats, M. Miozzo, and P. Dini, "Urban traffic forecasting using federated and continual learning," in 2023 6th Conference on Cloud and Internet of Things (CIoT), IEEE, 2023, pp. 1–8.

[13] M. V. S. da Silva, L. F. Bittencourt, and A. R. Rivera, "Towards Federated Learning in Edge Computing for Real-Time Traffic Estimation in Smart Cities," in Anais do Workshop de Computação Urbana (CoUrb), SBC, Dec. 2020, pp. 166–177. doi: 10.5753/courb.2020.12361.

[14] C. Zhang, L. Cui, S. Yu, and J. J. Q. Yu, "A Communication-Efficient Federated Learning Scheme for IoT-Based Traffic Forecasting," IEEE Internet of Things Journal, vol. 9, no. 14, pp. 11918–11931, Jul. 2022, doi: 10.1109/JIOT.2021.3132363.

[15] M. Wilbur, C. Samal, J. P. Talusan, K. Yasumoto, and A. Dubey, "Time-dependent Decentralized Routing using Federated Learning," in 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC), May 2020, pp. 56–64. doi: 10.1109/ISORC49007.2020.00018.

[16] S. Kaleem, A. Sohail, M. U. Tariq, and M. Asim, "An Improved Big Data Analytics Architecture Using Federated Learning for IoT-Enabled Urban Intelligent Transportation Systems," Sustainability, vol. 15, no. 21, Art. no. 21, Jan. 2023, doi: 10.3390/su152115333.

[17] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting Unlabeled Data in Smart Cities using Federated Learning." arXiv, Mar. 04, 2020. doi: 10.48550/arXiv.2001.04030.

[18] A. Huang et al., "StarFL: Hybrid Federated Learning Architecture for Smart Urban Computing," ACM Trans. Intell. Syst. Technol., vol. 12, no. 4, p. 43:1-43:23, Aug. 2021, doi: 10.1145/3467956.

[19] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, "FASTGNN: A Topological Information Protected Federated Learning Approach for Traffic Speed Forecasting," IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8464–8474, Dec. 2021, doi: 10.1109/TII.2021.3055283.

[20] L. Li, Y. Zhao, J. Wang, and C. Zhang, "Wireless Traffic Prediction Based on a Gradient Similarity Federated Aggregation Algorithm," Applied Sciences, vol. 13, no. 6, p. 4036, 2023.

[21] N. Algiriyage, R. Prasanna, E. Doyle, K. Stock, and D. Johnston, "Traffic Flow Estimation based on Deep Learning for Emergency Traffic Management using CCTV Images," in Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management, Blacksburg, VA, USA, 2020, pp. 24–27.