IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 June 2024, accepted 9 August 2024, date of publication 14 August 2024, date of current version 28 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443298

RESEARCH ARTICLE

Redefining Urban Traffic Dynamics With TCN-FL Driven Traffic Prediction and Control Strategies

K. M. KARTHICK RAGHUNATH[1], (Member, IEEE), C. ROHITH BHAT[2], VENKATESAN VINOTH KUMAR[3], (Member, IEEE), VELMURUGAN ATHIYOOR KANNAN[4], T. R. MAHESH[5], K. MANIKANDAN[6], AND N. KRISHNAMOORTHY[3]

[1]Department of Computer Science, JAIN (Deemed-to-be University), Global Campus, Bengaluru, Karnataka 560034, India
[2]Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai 602105, India
[3]School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, Tamil Nadu 632014, India
[4]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur 522302, India
[5]School of Computer Science and Engineering, JAIN (Deemed-to-be University), Bengaluru, Karnataka 560034, India
[6]School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Vellore, Tamil Nadu 632014, India

Corresponding author: N. Krishnamoorthy (krishnamoorthy.n@vit.ac.in)

ABSTRACT The smart city traffic management domain is perpetually a crucial sector that requires innovative strategies due to expanding urbanization and vehicle use. In this study, we have introduced a traffic prediction and handling system that utilizes Temporal Convolutional Networks (TCNs) combined with Federated Learning (FL) to deal with urban traffic effectively. This approach leverages the sophisticated functionalities of TCNs to evaluate and estimate traffic trends, such as congested phases, traffic flow, and ideal mobility routes. The system guarantees data privacy and utilizes decentralized information analysis using Federated Learning. In this approach, every point in the intelligent city network, including traffic sensors and cameras, serves to collectively comprehend traffic patterns without disclosing raw data. Using this cooperative method not only improves the model's ability to forecast outcomes accurately but also enables efficient real-time traffic management, with the ability to adapt to changing conditions. The method has shown significant efficacy in enhancing traffic flow and mitigating congestion. The critical criteria are a 20% drop in average commuting times, a 25% drop in traffic congestion, and a 15% enhancement in emergency response times. These statistics highlight the system's efficiency in improving urban traffic control by integrating modern technologies.

INDEX TERMS Decentralized learning, mobile patterns, transport, traffic prediction, accuracy, normalization, temporal convolution.

I. INTRODUCTION

Urban traffic patterns of today's world are a constantly changing facet of urban life that is inseparably linked with the speedy process of urbanization and technological advances. In modern urban traffic, we are facing a situation in which there is a high vehicle concentration, many different transport types, and growing market requirements for safe mobility [1]. Such challenges as traffic jams, longer driving times, and adverse environmental effects that include increased emissions of pollutants are just a result of the number of vehicles in use. Moreover, the complexity of traffic composition, comprising private and public transport vehicles and bicycles, is an intricate system demanding complicated management policies.

New technologies such as smart transportation systems, real-time data analytics, and AI-driven predictive models are also changing the current dynamics of urban traffic [2], [3]. These innovations seek to sharpen traffic flow, promote road safety, and improve urban transit's ecological footprint. Integration of IoT devices such as traffic sensors and cameras results in real-time data that makes it a lot easier to understand/predict patterns relating to how movement continues. To conclude, contemporary urban traffic dynamics combine

The associate editor coordinating the review of this manuscript and approving it for publication was Yanli Xu.

the traditional issues amplified by the growth of metropolitan populations with many solutions presented through technological innovations [4], [27]. It is now a critical field of attention in urban planning and development.

Redefining urban traffic dynamics captures a significant and ever-growing component of the field. With an increasing number of people moving to urban centres and a corresponding rise in motor vehicle usage, cities across the globe are confronted with urgent problems with traffic routing, congestion mitigation, and transit effectiveness. Conventional traffic control systems are usually much less efficient in managing the complex and diverse nature of the urban flow, creating a need for creative alternatives that change with time [5].

This need has created a lot of enthusiasm towards using new technologies, especially AI and data analysis [6]. Thus, integrating AI-driven approaches like Temporal Convolutional Networks (TCNs) [7], [23] and Federated Learning (FL) [8] is a viable solution. Time-series data intrinsic to the traffic patterns can be easily analyzed with the use of TCNs that provide, in turn, reliable predictions for flow level and congestion phases as well as optimal mobility routes. At the same time, FL can be viewed as a revolutionary approach to data processing focused on distributed learning directly at information sources such as traffic sensors and cameras. This improves scalability and efficiency and fits modern data security issues.

By adopting such advanced methodologies, urban traffic dynamics will be redefined in the cities, resulting in more intelligent, more responsive, and sustainable cities [9]. Research in this field is critical for designing intelligent traffic systems capable of responding to dynamic conditions, balancing resource allocation, and increasing the quality of urban life.

This research focuses on an enhanced traffic management system integrated with smart city infrastructures. It focuses on integrating TCNs with FL to enhance traffic prediction accuracy and efficiency. This system aims to address the challenges of urban traffic congestion and mobility, leveraging cutting-edge AI technologies for real-time, data-driven traffic management and decision-making.

The prominent work of the research is primarily focused on attaining the following objectives.

1) To utilize TCNs to analyze time-series data from various urban traffic sources, aiming to accurately predict traffic patterns, including congestion phases and flow rates.
2) To implement FL to process data from multiple points in the city network, like traffic sensors and cameras, to enable more efficient traffic control and reduce average commuting times.

To leverage the integrated TCN-FL system's predictive capabilities to enhance emergency services' responsiveness, aiming for a substantial reduction in emergency response times.

II. RELATED WORK

This section delivers a rundown of the current approaches, emphasizing how well they handle prevalent issues in urban traffic dynamics and how efficient they are. The critical importance of research work in current traffic management systems is highlighted in this study, which aims to provide a full grasp of the advances and problems in this expanding subject.

Ajayi and Kumkal [10] investigated Multi-Agent Adaptive Routing Optimization (MAARO) to alleviate city traffic jams. It prioritizes decentralization, edge computing, and coordination at the cloud level to enhance traffic flow. The primary objectives of the research are to ascertain the efficacy of AI-driven solutions in easing traffic congestion, the viability of AI integration into transportation networks, and the impact of Intelligent Route Planning and Optimization (IRPO) on traffic management. Additionally, it explores pertinent literature on ITS, smart transportation, and AI transportation technology. The model for implementation, cybersecurity priorities, and future suggestions for researchers and policymakers are also covered. The research offers a thorough strategy for reducing traffic jams in metropolitan areas using decentralized routing optimization and AI-driven solutions. Cuong and Aziz [11] investigated using AI-powered vehicle recognition for traffic management. It uses deep learning integrated with computer vision to recognize and categorize cars in real-time, improving incident management, monitoring congestion, and analysis of traffic flow. The report delves into ethical and privacy concerns, highlighting the need for responsible implementation and adherence to industry norms and laws. Methods include training models using transfer learning, deploying them in real-time, and evaluating their performance using measures like accuracy and precision. The implementation of these systems is what the methodology is all about. The article talks about how AI-powered vehicle identification might revolutionize city transportation systems, highlighting how accurate, efficient, and flexible it is while also mentioning its ethical and legal hurdles. Li et al. [12] introduced a thorough method known as Dynamic Graph Convolutional Recurrent Network (DGCRN) for traffic prediction. The authors proposed a model that successfully captures traffic data's geographical and temporal interdependence. They use a generic training strategy for RNN-based models, perform considerable experiments on open-source datasets, and provide a novel dataset about real-world congestion. Additionally, the paper contrasts the suggested model with fifteen baselines, showing how it significantly reduces errors and achieves state-of-the-art prediction accuracy. This technique uses various methods, including recurrent neural network (RNN) models, graphical neural network (GNN) models, attention mechanisms, and external characteristics such as time and weather data, to improve performance and capture spatial and temporal correlations. In their study, Djenouri et al. [13] focused on creating a novel convolutional graph-based neural network for predicting urban traffic flows in an edge IoT setting. This research integrates graph convolutional neural networks for training, outlier identification for data cleaning, and hyperparameter branch-and-bound optimization. Using outlier identification, the pre-processing step eliminates unnecessary nodes from the road network. A novel optimization method based on branch and bound is used to find the optimal values for the framework's hyperparameters, and the graph convolutional neural network is enhanced to make traffic flow predictions. With features including hyperparameter optimization, graph convolutional neural networks, and outlier detection, the suggested PROMOTION framework is a smart hybrid that can predict traffic flow. Extensive trials with existing traffic flow prediction data revealed that it outperforms baseline techniques in terms of both accuracy and runtime. He et al. [14] introduced ST-3DGMR, an end-to-end framework tailored for region-based traffic analysis, as a revolutionary approach to the challenge of urban flow forecasting. In practice, a distinctive blend of closeness and periodic 3DCNN branches is the essence of ST-3DGMR. The ST-3DGMR provides a novel solution for urban planners and systems that monitor urban mobility; it is built on grouped multi-scale approaches incorporated into the ResNet framework and can remember more complicated "urban" traffic patterns. This strategy outperforms current options regarding traffic prediction, as seen by the 2.6% and other more substantial reductions in RMSE for state-of-the-art systems. The model's superior capacity to handle and analyze complicated spatiotemporal data drives this improvement over previous models. Optimal vehicle routing and reduced traffic congestion are the primary goals of Dikshit et al. [15] investigated the revolutionary possibilities of AI in urban traffic management. To improve transportation efficiency and positively impact urban environments, it explores how AI can analyze real-time traffic data, create smart routing methods, and adjust to traffic situations on the fly. The study combines comprehensive simulations with case studies to assess the efficacy of AI-driven traffic management systems. This research will be useful for transportation authorities and lawmakers looking for new ways to promote sustainable urban growth. Methods like AI-powered vehicle route optimization and machine learning-based traffic management are part of the used approach. Intelligent assignment models for urban traffic planning using optimum route optimization algorithms are the subject of Lu et al. [16]. It tackles complicated traffic problems and the disparity between the expansion of automobiles and transportation infrastructure. The research stresses using contemporary technology to make traffic flow smoother, lessen congestion, increase road safety, and lessen the adverse effects of air pollution. The article delves into urban road traffic simulation, specifically how a dynamic route optimization algorithm may be developed utilizing graph theory and real-time traffic data. Aligning with users' travel demands, the algorithm strives to deliver the quickest real trip time. Using data on urban road networks [28] and traffic conditions, the research emphasizes the significance of route optimization in dynamic path guidance systems and their function in real-time vehicle guiding. At the end of the study, the authors address the difficulties of traffic growth and congestion in big and medium-sized cities throughout the globe, as well as the effects of IT on urban motorization [17]. Despite progress, current research in urban traffic control needs to adequately combine privacy-aware, scalable learning models with comprehensive, real-time traffic data analysis. To bridge this gap and improve traffic prediction precision and adaptation in dynamic urban environments, it is necessary to combine TCNs with FL. This will allow us to use TCNs' expertise in time-series assessment with FL's decentralized data processing.

III. METHODOLOGY

The architecture of the proposed model depicted in Figure 1 exhibits the integration of TCNs, and the FL has been formulated to improve the functioning of urban traffic control through an all-around feed cycle processing strategy. This process starts by gathering the traffic information from the General Modeling Network Specification (GNMS) [18] and Urban Traffic Dataset (UTD19) [19] datasets to train. Pre-processing of the collected data includes steps like normalization and timestamp encoding. In the FL framework, individual nodes, such as the traffic sensors, train local models using their decentralized capability primarily to increase scalability and efficiency in dealing with massive data. This local training is part of a central aggregation process, wherein the server collects model updates from all nodes, emphasizing parameter aggregation as an improvement for accuracy and efficiency. The global TCN model is then updated with these aggregated parameters, leveraging its time-series feature analysis abilities to determine traffic conditions correctly. Real-time traffic management, including phasing and timing of signal control, as well as route selection, is guided by these predictions. Finally, a feedback loop is created where real-time traffic information is continually incorporated into the system for dynamic adjustments and ongoing enhancement of traffic control measures. While keeping the data secure, this architecture focuses mainly on the scalability and efficiency that are essential in managing the dynamic complexities of traffic at a town level.

Several computational processes comprise the proposed strategy for enhancing urban traffic management with the concourse of Federated Learning (FL) and Temporal Convolutional Networks (TCNs).

A. DATA COLLECTION

For study purposes with optimal data diversity, we utilized two different traffic datasets: GMNS and UTD19 datasets. The GNMS dataset appears to contain multiple CSV files, each likely representing different components of the dynamic multi-modal transportation network model of Melbourne, Australia. The dataset adheres to the GMNS, ensuring software-agnostic human and machine readability.

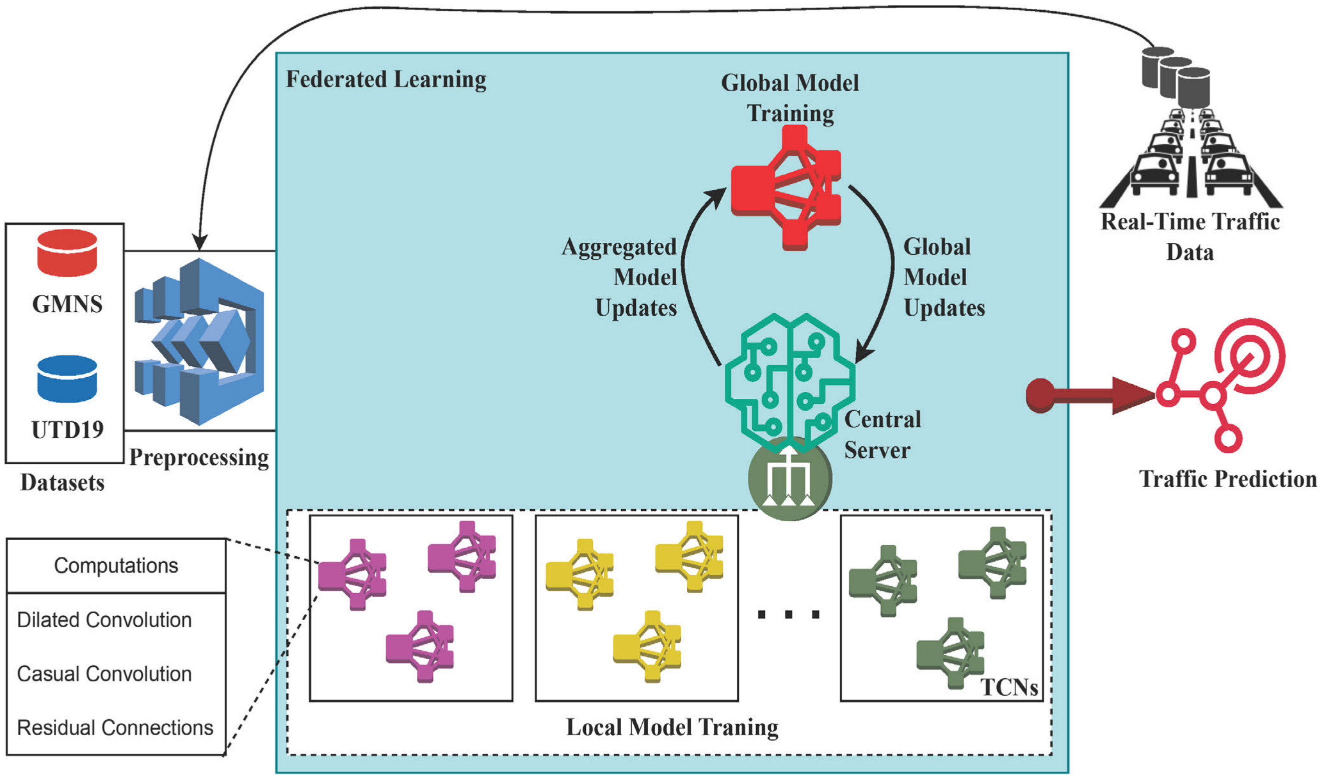

FIGURE 1. Architecture of the proposed model.

This makes the dataset suitable for various transportation operations and planning models. It likely includes data for different modes of transportation and their interactions over time, represented through various timestamped files (for different time intervals). Table 1 summarizes the key attributes of the files in the dataset, providing a glimpse into the structure and type of data available for the Melbourne transportation network under the GMNS.

UTD19 data is valuable for understanding city traffic capacities and the points where urban networks get congested. Key highlights include,

1) Macroscopic Fundamental Diagram (MFD) Analysis: The dataset applies the MFD approach, linking vehicle accumulation to travel production in urban networks.
2) Critical Point Analysis: It determines the vehicle's critical number and traffic capacity, a foundation for understanding free-flow congested state boundaries in urban transport.
3) Data Source and Methodology: The data is compiled from billions of vehicle observations across over 40 cities and network information sourced from OpenStreetMap.
4) Statistical Modeling and Traffic Analysis: The dataset requires sophisticated statistical modeling such as two-stage least squares regression analysis to identify the factors contributing towards critical points.

This dataset is instrumental in urban planning and traffic management, allowing for the prediction of traffic capacities and the understanding of critical congestion points in urban networks. Table 2 highlights the vital components of the dataset for precise understanding.

TABLE 1. Key attributes of GMNS dataset.

File Name	Key Attributes	Values/Ranges
node.csv	- node_id	60399, 60400, ...
	- x_coord, y_coord	(315981.0612, 5810552.063), ...
	- node_type	-
	- ctrl_type	-
link.csv	- link_id	2967, 2970, ...
	- from_node_id	60399, 60404, ...
	- to_node_id	60400, 60405, ...
	- length	-
	- capacity	9000, ...
	- free_speed	100 km/h, 50 km/h, ...
	- lanes	5, 1, ...
movement.csv	- From Link - To Link - Type	includes authorized movement kinds, from and to link identifiers
signal_timing_plan.csv	- timing_plan_id	1, 2, ...
	- controller_id	1, ...
	- cycle_length	180 seconds, 179 seconds, ...
OD_matrix_615_630.csv	- Origin ID	317260, 317263, ...
	- Destination ID	317263, 317265, ...
	- Demand	0, 14, 274, ...
observed_traffic_volume.csv	- link_ID	8220, ...
	- time	6:15, 6:30, ...
	- observed_volume	47, 70, 88, ...

TABLE 2. Vital components of the UTD19 dataset.

Attribute	Description	Computed Units
Vehicle Accumulation (n)	Number of vehicles in an urban network	Vehicles/km²
Travel Production (P(n))	Measure of traffic flow efficiency	Vehicle-km/h
Critical Accumulation (a*)	Point of maximum traffic capacity	Vehicles/km²
Critical Speed (v*)	Speed at which congestion starts	km/h
Road Network Density (R)	Road space per unit area	Lane-km/km²
Intersection Spacing (I)	Average distance between signalized intersections	LSA network-km⁻¹
Bus Production Density (B)	Density of the bus network per unit area	Bus-km/h/km²
Network Capacity (P*)	Maximum traffic capacity at the critical accumulation	Vehicle-km/h/km²
Statistical Measures	Various statistical metrics for understanding traffic and network data	Elasticities, mean, standard deviation etc.

The GMNS dataset contains a broad and well-defined geographical representation of Melbourne's transportation system with its specially marked Nodes, Links, movements, Signal timings and traffic volumes. As such, it is used to train TCNs for learning temporal as well as /or spatial dependencies in the traffic data. The feeding of the timestamped data into the TCNs assists the model in the learning of the information given to predict the traffic flow, phases of congestion as well as the best routes, depending on the phase numbers learned by the model. On the other hand, the UTD19 dataset provides a bird's eye view as to the overall traffic capacities for roads and where the issues are. It includes a number of vehicles on the road, traffic generation, importance accumulation, importance velocity and the network supply. These attributes are desirable for FL that enables distributed data analysis of multiple sensors and cameras in smart cities network. FL makes sure that the raw data is not disclosed to the outside world for the user's information privacy, but it allows the aggregation of the different models simultaneously.

Thus, the GMNS dataset is used in training TCNs to some specific traffic characteristics and behaviors while the UTD19 gives only general overview of traffic capability with congestion zones. These datasets are integrated into the TCN-FL method and enables the system to learn, and manage the traffic in real-time based on the proposed predictions. This dataset ensures a wide range of scenarios and traffic conditions are covered.

B. PRE-PROCESSING

Data analysis and modeling rely heavily on pre-processing, especially when dealing with complicated datasets such as GMNS and UTD19. Normalization and timestamp encoding calculations are performed to apply pre-processing to the GMNS and UTD19 datasets. Normalization is used to modify their scale to ensure that all data characteristics are treated similarly in the analysis. Using a standard Min-Max scaling procedure [20], all numerically attributed files are selected from the datasets and computed, which can be expressed as,

$$\textit{\textbf{normalized}}\ (\boldsymbol{i}) = \frac{\boldsymbol{i} - \textit{\textbf{min}}\ (\boldsymbol{i})}{\textit{\textbf{max}}\ (\boldsymbol{i}) - \textit{\textbf{min}}\ (\boldsymbol{i})} \tag{1}$$

From (1), i indicates the original value, $min(i)$ and $max(i)$ represents the minimum and maximum values of the each attribute, respectively. In addition, converting timestamps into a format suitable for TCNs, like one-hot encoding for time of the day, day of the week, and other OD (origin-destination) metrics. Given a categorical variable T representing the hour of the day (where $T$is an integer representing varying timestamps), the one-hot encoding process can be represented as follows:

Let $\hat{V}$ be a vector corresponding to the timestamps in a day. The one-hot encoding function for each element V_i of vector $\hat{V}$, where i ranges as per the time units, and is defined as:

$$\boldsymbol{V_i} = \begin{cases} \mathbf{1}, & \textit{if}\ i = T \\ \mathbf{0}, & \textit{otherwise} \end{cases} \tag{2}$$

C. TEMPORAL CONVOLUTIONAL NETWORKS

TCNs are a type of neural network designed for time-series data. The key components in TCNs include convolutions and connections, namely, dilated and casual convolutions and residual connections. To account for interdependencies across time scales, the model uses dilated convolutions. For example, it analyzes transient and persistent traffic patterns over more extended periods. Causal convolutions guarantee that forecasts are grounded only upon historical and real-time data. This is crucial for traffic forecasting in real-time. To avoid training problems and ensure the model can successfully understand complicated traffic patterns, residual connections are used to construct deeper TCN models. The end result is a forecast of traffic conditions, such as congestion levels and best routes, for the future. User can utilize the predictions to make real-time adjustments to traffic signals or get routing information. The system can then utilize the results of these actions to refine the model continuously. This section describes the computational aspects of essential components. It explains in detail how TCNs are computed concerning the proposed traffic prediction model.

Dilated Convolutions: With dilated convolutions, the network can significantly increase its receptive field without a commensurate parameter count rise. Initially, an input sequence $S(i)$ is considered, where i ranges from 0 to $N-1$ and N is represent the size of the input. The dilated convolution output $DC(s)$ at time step s is computed as follows:

$$\boldsymbol{DC}\ (\boldsymbol{s}) = \sum_{\boldsymbol{i=0}}^{\boldsymbol{N-1}} \boldsymbol{S(i)} \cdot [\boldsymbol{F}\ (\boldsymbol{s} - \boldsymbol{d} \cdot \boldsymbol{i})] \tag{3}$$

From (3), F denotes the filter, d specifies the dilution factor, and d increases (as moving deeper into the network) in an exponential way (example: $d = 2^n$). The $S(i)$ represents

TABLE 3. **Algorithm 1: computational process of TCN.**

Inputs:
- $S(i)$: Input traffic data sequence.
- L: Number of layers in the TCN.
- $\{F_1, F_2, \cdots, F_L\}$: Set of filters for dilated convolutions (DC), one for each layer.
- $\{d_1, d_2, \cdots, d_L\}$: Dilation factors for each layer.

Output:
Y: Output sequence after processing through TCN layers.

1: function TCN_Layer_Processing$(S(i), L, \{F_l\}, \{d_l\})$
2: $\quad Y \leftarrow S(i)$
3: $\quad$ for $l = 1$ *to* L do
4: $\qquad Y \leftarrow R(Y, F_l, d_l)$
5: $\quad$ end for
6: $\quad$ return Y
7: end function
8: function R$(S(i), F, d)$
9: $\quad DC_{output} \leftarrow$ DC$(S(i), F, d)$
10: $\quad$ if length$(S(i)) \neq$ length$\left(DC_{output}\right)$ then
11: $\qquad S(i) \leftarrow adjust_{dimensions}\left(S(i), length\left(DC_{output}\right)\right)$
12: $\quad$ end if
13: $\quad R \leftarrow DC_{output} + S(i)$
14: $\quad$ return R
15: end function
16: function DC$(S(i), \{F_l\}, \{d_l\})$
17: $\quad N \leftarrow$ length$(S(i))$
18: $\quad M \leftarrow$ length(F_l)
19: $\quad f \leftarrow$ Initialize empty sequence of length N
20: $\quad$ for $s = 0$ to $N - 1$ do
21: $\qquad$ sum $\leftarrow 0$
22: $\qquad$ for $i = 0$ to $M - 1$ do
23: $\qquad\quad$ if $(s - d * i) \geq 0$ then
24: $\qquad\qquad sum \leftarrow sum + S(i)[s - d * i] * F[i]$
25: $\qquad\quad$ end if
26: $\qquad$ end for
27: $\qquad f[s] \leftarrow sum$
28: $\quad$ end for
29: $\quad$ return f
30: end function

the input traffic data sequence which is derived from the GMNS and UTD19 datasets.

Causal Convolution: Ensures that the model does not violate the temporal order, meaning it doesn't use future information to predict the present or past. In causal convolution, the output at time t depends only on inputs from time t and earlier; thus, it is computed as follows (for g with size k).

$$\boldsymbol{DC}\,(\boldsymbol{t}) = \sum\nolimits_{\boldsymbol{i=0}}^{\boldsymbol{k-1}} \boldsymbol{S}(\boldsymbol{t}-\boldsymbol{i}) \cdot \boldsymbol{g}(\boldsymbol{i}) \tag{4}$$

Thus, (4) ensures that for the computation of $DC\,(t)$, only values up to $S\,(t)$ are utilized.

Residual Connections: It makes the process easier to train deeper networks by preventing the explosion or vanishing of gradients as they pass through layers. A residual block in TCNs aims to learn the input-output difference or residual. This process facilitates deep network training by reducing the impact of the vanishing gradient issue. After processing through these layers, the output $DC(i)$ represents the transformed version of the input, which is applied as an input to perform the process of residual blocks. The $DC(i)$ represent the output from the convolutional layers within the residual block when applied to i. This function includes operations like dilated convolutions, activation functions, etc. thus, the computation of the blocks are represented as,

$$\boldsymbol{R} = \boldsymbol{DC}\,(\boldsymbol{i}) + \boldsymbol{S}(\boldsymbol{i}) \tag{5}$$

In (5), $DC\,(i)$ is the transformation applied by the convolutional layers, and $S(i)$ is the original input to the block. The addition of $S(i)$ to $DC\,(i)$ is what constitutes the residual connection. This step is crucial as it allows the network to learn residuals. If the identity mapping is optimal (i.e., the input is already close to the desired output), the network can easily achieve this by pushing $DC\,(i)$ towards zero, making R approximately equal to $S(i)$.

Table 3 represents the TCN process in an algorithmic to exhibit the computational process of a TCN for processing time-series data, such as traffic patterns. The process begins by initializing the output sequence Y with the input sequence $S(i)$, ensuring that the initial state of Y reflects the raw input data (Line 2). The algorithm then iterates over each layer of the TCN (Lines 3-5), where a crucial transformation occurs at every layer by applying a residual block (Line 4).

Each residual block, defined in Lines 8-15, first applies a dilated convolution to the input sequence. This convolution uses a specific filter and dilation factor unique to that layer, allowing the network to capture temporal dependencies at varying scales. The dilated convolution function (Lines 16-30) systematically alters the input data by applying the convolution operation across the sequence, considering the dilation factor, which effectively expands the receptive field without increasing the parameters.

After convolution, the residual block returns the convolved output to the original input sequence. This step may include an adjustment of dimensions if the sizes of the convolved output and the original input differ. This addition is a crucial aspect of residual learning, enabling the network to learn modifications to the identity mapping of the input rather than learning from scratch, thus easing the training of deeper network architectures.

Finally, the algorithm concludes each iteration by updating the output sequence Y with the result from the current residual block, iteratively refining Y as it progresses through each layer of the TCN. This iterative refinement captures complex patterns and dependencies in the data, making TCNs particularly effective for time-series analysis tasks.

D. FL PROCESS IN TRAFFIC PREDICTION

The combination of FL with TCNs in the context of traffic prediction is a complex technique [24]. Such hybridization enables distributed training and aggregation, improving the prediction of traffic patterns, congestion cycles and optimal routes. The FL approach comprises five processes: local model training (LMT), central server aggregation (CSA), global model update (GMU), traffic prediction and real-time adaptation with a feedback loop for continuous optimization.

LMT: Without disclosing this information to other parties, it trains local TCN models using data acquired at each node (such as a traffic sensor or camera) and wherein the updated model parameter, ϑ_i^{t+1} can be computationally expressed as,

$$\boldsymbol{\vartheta_i^{t+1}} = \boldsymbol{\vartheta_i^{t}} - \boldsymbol{\alpha\nabla\gamma}\left(\boldsymbol{\vartheta_i^{t}}, \boldsymbol{D_i}\right) \tag{6}$$

From (6), $\vartheta_{\mathrm{i}}^{\mathrm{t}}$ depicts the LMT parameter at iteration τ for the i^{th} node, α signifies the learning rate (an iterative hyperparameter that sets the size of the step as the function approaches its minimum), $\nabla\gamma\left(\vartheta_i^t, D_i\right)$ at node i denotes the local data $D_{i,}$ which is used to assess the gradient of the loss function γ concerning the model parameters, ϑ_i^t. If the loss is minimized, this gradient will show the way to change the model parameters.

CSA: At this stage, all of the nodes' parameter changes are aggregated and the global model is updated, which is computationally expressed as,

$$\vartheta_{\boldsymbol{i}}^{\boldsymbol{t+1}} = \vartheta^{\boldsymbol{t}} + \frac{\mathbf{1}}{\boldsymbol{n}} \sum\nolimits_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\Delta\vartheta_i^{t+1}} \tag{7}$$

From (7), n represents the total of all the nodes, and $\Delta\vartheta_i^{t+1}$ denotes the update from theith node.

GMU:Use the aggregated transforms to update the global TCN model; this will be utilized for traffic estimation. The aggregated updates from all nodes are used to revise the global model variables, ϑ_i^{t+1}, ensuring that the global model learns from the combined knowledge of all local models.

Traffic Predictions: The global TCN model is revised by adding parameters ϑ_i^{t+1} to forecast upcoming traffic conditions. Patterns and temporal relationships in the aggregated data across nodes are included in the model. The prediction can be represented as:

$$\boldsymbol{\vartheta^{global}} \leftarrow \boldsymbol{TrafficConditions} = \boldsymbol{TCN}\left(\boldsymbol{\vartheta_i^{t+1}}, \boldsymbol{\beta}\right) \tag{8}$$

From (8), β denotes the current data. Table 4 shows the algorithm that successfully integrates FL's distributed learning and TCN's ability to recognize temporal patterns.

TABLE 4. Algorithm 2: training and aggregating local TCN models in a global setting.

Input:
Number of nodes: n
Number of local iterations: τ
Learning rate: α
*Output:*Updated global TCN model parameters: ϑ^{global}

Initialize global TCN model parameters, ϑ^{global}.
For each global iteration **do**:
  Initialize aggregated updates $\Delta\vartheta_i^{t+1}$ to zero.
    For each node i from 1 to n in parallel **do**:
    Set local model parameters ϑ_i equal to ϑ^{global}
//Copy global parameters to local model)
    For =1 to τ (local iterations) **do**:
    Local model training using TCN:
        Compute gradient $g_i$as $\nabla\gamma(\vartheta_i^t, D_i)$
        Update local model parameters: $\vartheta_i = \vartheta_i - \alpha \cdot g_i$
    End for
        Compute local updates $\Delta\vartheta_i$ as $\vartheta_i - \vartheta^{global}$
        Send $\Delta\vartheta_i$ to the central server.
    End for
    Aggregate updates: $\vartheta_i^{t+1} = \vartheta^t + \frac{1}{n}\sum_{i=1}^{n} \Delta\vartheta_i^{t+1}$
    Update global model: $\vartheta^{global} = \vartheta^{global} + \vartheta_i^{t+1}$
End for

E. REAL-TIME ADAPTATION AND FEEDBACK LOOP

Real-time adjustment and feedback loop in FL combined with TCNs for traffic prediction is an adaptive, continuous process that strives to enhance model accuracy while adapting according to the changing nature of traffic. This process is cyclical and consists of several key steps:

Implement Decisions: The traffic predictions generated by the global TCN model are used to make real-time traffic management decisions. Such decisions involve adjusting traffic signal timings, rerouting traffic, or implementing congestion control measures.

Collect New Data: Following the implementation of those decisions, new data is collected from each node (such as traffic sensors or cameras). This data reflects the current state of traffic post-decision implementation, providing insight into the effectiveness of the decisions and current traffic conditions.

Update Local Models: Each node then uses this newly collected data to update its local model. This is done by retraining the local TCN model on the updated dataset, which includes the most recent traffic information.

Aggregate Updates: Each local model's parameter updates $\Delta\vartheta_i^{t+1}$ are sent to a central server. These updates represent the changes in the local models due to the newly incorporated data.

Update Global Model Again: The central server aggregates these new updates from all nodes and uses them to update the global TCN model. This step ensures that the global model benefits from all local models' collective learning and experiences, reflecting the latest traffic trends and effects of the implemented decisions.

This system's real-time adaptability is based on this feedback loop of prediction, decision execution, data collecting, and model update. As a result, the traffic management system can improve its real-time control and prediction capabilities via continual development and adaptation to new trafficking trends. The system ensures it stays sensitive to the ever-changing nature of urban traffic circumstances, which improves management efficiency and effectiveness.

Within the scope of our research, the network based on TCN can perform the function of a real-time traffic management system by adjusting traffic lights' duration and route suggestions. For instance, when a surge in traffic inflow is expected at a particular intersection during rush hours, which could be the case, the traffic control system can timely extend the green light duration to reduce congestion. On the one hand, during peak hours or when traffic congestion is expected, the system is designed to have a longer green phase time to improve traffic flow and minimize excessive waiting. In contrast, during off-peak times or hours when fewer vehicles are expected on the road, the system adjusts the green phases to optimize traffic flow and reduce delay times [26]. In addition, the predictions are applied to navigation systems to give drivers timely route directions at every present. By monitoring anticipated traffic situations, these systems provide recommended alternative routes, distributing the traffic equally throughout the system network.

IV. PERFORMANCE EVALUATION AND ANALYSIS

A. EXPERIMENTAL CONFIGURATIONS

Table 5 represents the essential empirical parameters for the evaluation of the proposed model. Beside such parameters, a collection of specialized software platforms and frameworks constitutes the empirical setting for deploying a traffic management system that uses TCNs and FL. Python v3.8 is the language of choice for implementation and data processing. Keras, which is compatible with TensorFlow v2.0 and provides a high-level API for model development and training, is considered for developing TCN and FL models. Pandas and NumPy provide potent tools for numerical computation, data manipulation, and analysis in their most recent iterations. Data preparation and metrics computation are handled using Scikit-learn. The development and testing of the AI-driven traffic control solution are built upon this software ecosystem. Besides the software requirements, a few vital hyperparameters are listed in Table 6 for training purposes.

TABLE 5. Fundamental empirical parameters.

Parameter	Value/Setting
Simulation Software	SUMO (Simulation of Urban MObility)
Traffic Dataset	GMNS, UTD19
Simulation Area	Downtown area of a major city
Total Simulation Time	100 hours
Time Step	1 second
Vehicle Entry Rate	50-120 vehicles/hour
Traffic Signal Timing	60 seconds green, 20 seconds red
Data Refresh Rate	every 5 minutes
Model Update Frequency	every 6 hours
Learning Rate (for FL)	0.01

TABLE 6. List of hyperparameters.

Parameters	Specifications	
Number of Layers (L)	10	TCN
Filters (F)	128 per layer	
Dilation Factor (d)	Exponential growth	
Learning Rate (α)	0.001	
Batch Size	32	
Number of Nodes (n)	100	FL
Local Iterations (τ)	10	
Global Iterations	50	

To conduct a thorough and definitive comparison of the proposed traffic management system that uses TCNs and FL, it is crucial to consider benchmarking against many state-of-the-art approaches. A few examples include ST-3DGMR, MAARO, and DGCRN. Our comparison study with the FL methodology helps put our TCN's performance and efficiency in perspective with the present landscape of sophisticated traffic management systems by including these methodologies. We aim to demonstrate the suggested system's validity as a novel strategy for urban traffic management and prediction by comparing it to these existing approaches and highlighting its advantages, weaknesses, and potential for development.

Many of the performance metrics would be critical to evaluate empirically how effective the proposed models are. The following measures reflect the system's ability to predict traffic trends, manage real-time congestion, and reduce overall observed levels.

Accuracy of Traffic Prediction: The Mean Absolute Error (MAE) [21] serves the central role in determining the accuracy of a traffic prediction model, which features within the smart urban traffic management system to predict different measures such as congestion, flow rates and mobility routes. With respect to this task, MAE proves the most appropriate since it represents a rather direct measure of the average absolute values of errors between model-generated predictions and real observed traffic flow conditions. This feature of MAE provides it with a property that can be considered an effective and understandable metric for the assessment of predictive accuracy in various traffic situations. At the same time, a parallel set of real observed traffic data was being collected. This real-world data served as an empirical reference, reflecting the actual traffic status and against which model predictions could be calibrated. The data at the times and locations corresponding to model predictions were used for a direct comparison of predicted traffic scenarios with the observed ones.

With these two sets of data available, the analysis began to measure the accuracy of the model through the calculation of MAE. The MAE was computed as the average of absolute differences between the predicted and observed values across all observations. Mathematically, the process was encapsulated in the formula [21]:

$$MAE = 1/N \sum_{i=1}^{N} \langle P_i - A_i \rangle \tag{9}$$

In (9), N was the total number of the data points considered. P_i, and A_i signifies the predicted and actual values, respectively. The MAE was computed by summing up these absolute differences between the predicted and observed data points, followed by averaging such sums over all observations, which provided a very objective numerical measure of accuracy with prediction from a model performance standpoint.

From the observations in Table 7, the TCN+FL methodology has the lowest MAE value of 1.5, which means it performs better than its competitors (ST-3DGMR, MAARO and DGCRN) at traffic pattern prediction. The MAE is a metric used to determine how close the forecasts or predictions are to the results. The lower the MAE is, the better its performance.

The superiority of TCN+FL is that both methods manage time-series data, and FL allows learning across various nodes in a distributed manner without sending all the data itself, thereby preserving privacy. This synthesis enables TCN+FL to learn from a much vaster set of data points and generalize

TABLE 7. Comparative analysis of MEA of different methodologies.

Methodology	MAE Value
ST-3DGMR	3.2
MAARO	2.8
DGCRN	2.5
TCN+FL	1.5

better across various traffic situations, as evidenced by the low MAE count.

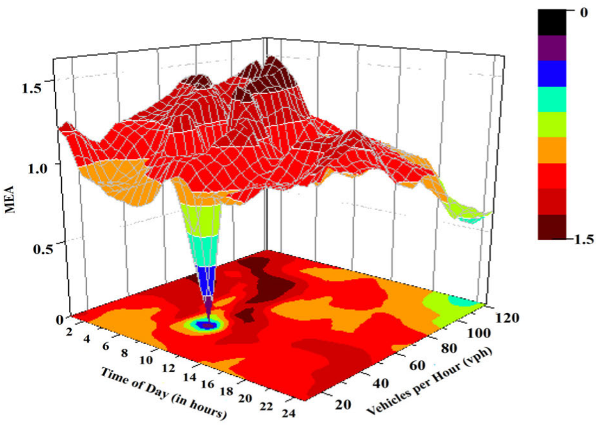

FIGURE 2. Analysis of MAE for the proposed TCN with FL.

Figure 2 visually represents the 3D contour MAE map, where the variation in MAE of the TCN+FL methodology across two dimensions, Time of the day and traffic flow (number of vehicles per hour) is shown. The X-axis represents the time of day describing a daily pattern in the traffic flow; The Y-axis shows the traffic volume to reflect intensity, and the Z-axis includes MAE values. The peaks and the valleys in the contour surface display variations of MAE concerning the time interval and traffic volume. A 'valley'—where the surface dips downward —implies a low MAE, which means better prediction accuracy is achieved, while a 'peak' implies otherwise.

The TCN+FL methodology keeps lower MAE values for most of the traffic flow irrespective of the time and value, although there are designated spots where the peak approximation happens. These peaks may refer to more chaotic traffic patterns or anomalies in the data that are difficult for this model to predict well. The global lower surface of TCN+FL shows its resistance to traffic prediction in different circumstances, improving local regulation's reliability and efficiency for better approaches.

Average Commuting Time (ACT) [25]: Evaluating the system's effectiveness in reducing commuter travel time.

$$ACT = \left[\frac{\mu_t^b - \mu_t^a}{\mu_t^b}\right] \times 100 \tag{10}$$

In (10), μ_t^b indicates the average time before the implementation of the proposed model and μ_t^a denotes the average time after the implementation of the proposed model. Assessing the performance of a transportation system in terms of reducing congestion can require complicated mathematical calculations, particularly with heterogeneous and large datasets for urban traffic relationships. Each traffic scenario is associated with a Traffic Congestion Index (TCI). The TCI is a summary measure incorporating other measures such as speed, density and count, which is computed as,

$$TCI = \omega_1 \cdot S_{factor} + \omega_2 \cdot \Psi_{factor} + \omega_3 \cdot \boxed{?}_{factor} \tag{11}$$

Form (11), ω_1, ω_2, ω_3 represents the weights of speed (S), density (Ψ), and count(€) factors, respectively. The inverse of mean speed (lower speeds = higher congestion) is computed to obtain the speed factor, which is expressed as,

$$S_{factor} = 1/average\_speed \tag{12}$$

Similarly, vehicle per kilometre (Q) is considered for measuring vehicle density (12), and the prevalence of vehicles in the network is considered for count factor (13).

$$\Psi_{factor} = Q/road\_segment\_length \tag{13}$$

Reduction in Traffic Congestion: The overall congestion measure for the area under study is calculated by aggregating TCI scores across all segments.This signifies the weighted average of each road segment depending on traffic volume.The aggregated congestion measure is calculated for the pre and post-implementation phases.

Figure 3 shows the relative efficacy of diverse traffic prediction techniques in reducing the ACT. The methodologies compared are ST-3DGMR, DGCRN, MAARO, and TCNs+FL, with the following observed reductions in ACT: ST-3DGMR: 13.7%, DGCRN: 12.8%, MAARO: 10%, and TCNs+FL: 20%.

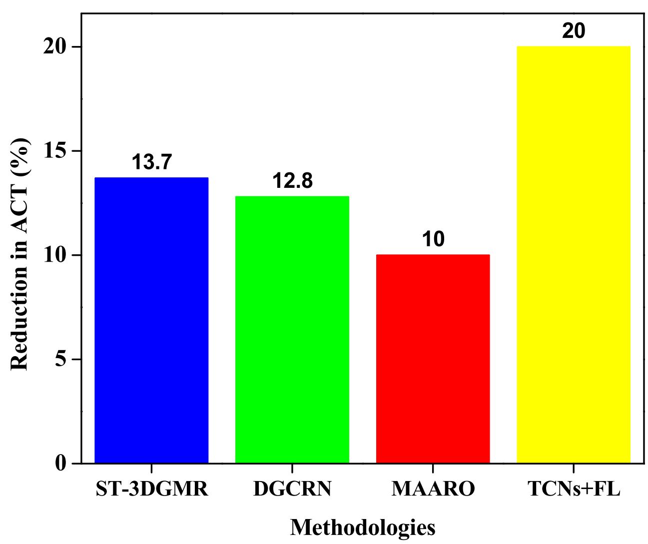

FIGURE 3. Comparing the efficacy of different traffic predictive methods in decreasing the ACT.

The TCNs with FL methodology demonstrate a 20% reduction in the ACT, which is significantly higher than other techniques. This excellent performance can be technically credited to the synergistic advantages of combining TCNs with FL. The TCNs are very good at modeling temporal correlations in sequential data, which is an essential requirement for accurately forecasting traffic patterns. The convolutional layers of TCNs can process very long sequences, which means that when making predictions, their model considers a wide range of past traffic data. For this, FL improves it by encouraging a distributed learning process over several nodes throughout the scenario. This implies that the TCN model is trained on a broad spectrum of data points, thereby leading to an effective and generalizable. FL further assists in adhering to privacy and minimizing centralized data aggregation, helping a logistically costly sensitive detail.

As a result of the heterogeneous data training, FL allows the TCN model to learn from multiple traffic states and trends that improve predictive accuracy. The capacity of TCNs to capture temporal patterns makes them very suitable for traffic data because they are naturally time-invariant. Additionally, continuous updates allow the model to make much better predictions over time as it captures new traffic patterns or revises its representativeness of urban infrastructure.

In the context of an FL framework, when evaluating the performance of a traffic prediction model, their accuracy varies on different numbers of nodes participating in said network, and this is by far its primary objective. The MAE estimates the model's performance for multiple combinations, altering 'n', which refers to a quantity in an FL network. In this sense, each node is regarded as a local source of traffic data that provides such information to the federated model through one or another sensor. The performance measure denoted by Performance(n) is the MAE of D_n, which is the aggregated data from these 'n' nodes. In other words, it is an average of the absolute errors between the traffic conditions predicted by the model and actual observed ones considering all contributing nodes. The performance of the model with 'n' nodes is computed as,

$$\boldsymbol{Performance\ (n) = MAE(D_n)} \tag{14}$$

From (14), performance(n) is evaluated with different values of 'n' to determine how adding or subtracting nodes makes a big difference in the accuracy score provided by the model. In FL, this analysis is critical since the model's efficiency might depend on quantitative and qualitative sourcing data from multiple nodes that are large in number and extent. Such an assessment enables the reduction of network size for efficient and accurate prediction of traffic flow optimization to ensure robustness, which is critical in dynamic urban environments characterized by growing complexity.

$$\boldsymbol{Efficiency\ (n) = \frac{Performance\ (n)}{|D_n|}} \tag{15}$$

In (15), the performance per unit of data is evaluated to determine how well the model performs relative to the amount of data it utilizes, which implies the amount of data gathered from 'n' nodes. This can be quantified in several metrics, including the number of data points, the volume of data (for example, megabytes), etc. An excellent efficiency rating implies the model can perform an adequate task with fewer details, which is beneficial for situations where collecting data is complicated or expensive.

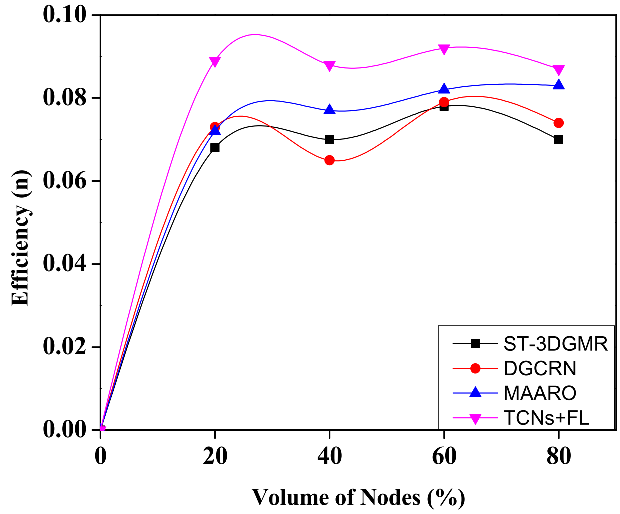

FIGURE 4. Comparative efficiency of traffic prediction methodologies by volume of nodes.

Figure 4, on the other hand, visualizes the overall efficiency result of all four methodologies as reflected in their predicted MAE per unit data from 'n' nodes. The X-axis represents the number of all participating nodes in a network.As the nodes increase from 0% to 80%, the efficiency varies for each methodology. Significantly, TCNs+FL demonstrate the best efficiency in some node volumes, reaching the highest value of 0.092 when volume is at 60%. This means that TCNs+FL is very good at staying with a lower MAE, even as the model's architecture.When the node volume reaches 20%, the TCNs+FL achieve an efficiency of 0. This implies that with only a portion of the network sharing data, TCNs+FL can use this as an input to make precise predictions. The efficiency of TCNs+FL drops a little with the increase in the volume of nodes to 40%, but it is still higher than the other models. This could suggest initial resistance to the growing size of the network with data quality or richness, which introduces overshadowing difficulties that a more extensive dataset entails. TCNs+FL achieve the highest efficiency of 60% node volume with a value: This represents the optimal efficiency of the model, and this can be due to effective aggregation with learning from the varied data sources in the FL network. However, at the same node volume of 80%, TCNs+FL show an efficiency drop to 0. This signifies diminishing returns since the model may become overwhelmed with redundant information or noise as more nodes contribute to data.

The new variability allows us to notice that the methodology of TCNs+FL retains a higher effectiveness for a wide range of node counts. This implies that a lower MAE (reflecting higher accuracy of traffic predictions) is achieved by

TCNs+ FL not only faster but also with much less data usage. It is especially apparent when the number of nodes grows, where TCNs+FL always performs better.

As such, when the TCN forecasts higher traffic volumes on routes used at peak hours, the traffic management system can immediately adjust the signal timings to extend the green light phase on these routes, thus balancing the traffic flows, minimizing congestion, and preventing delays. Correspondingly, the system could change the green phases to shorter periods or take optimal pedestrian crossing times in light traffic, ultimately leading to enhanced traffic efficiency and safety. With the support of machine learning, the signals of traffic facilities may be adaptive in forecasting traffic. As a result, cities can be equipped with responsive traffic management systems that help attain a smoother flow of vehicles, thus reducing congestion and potentially lowering emission levels due to minimized-idling.

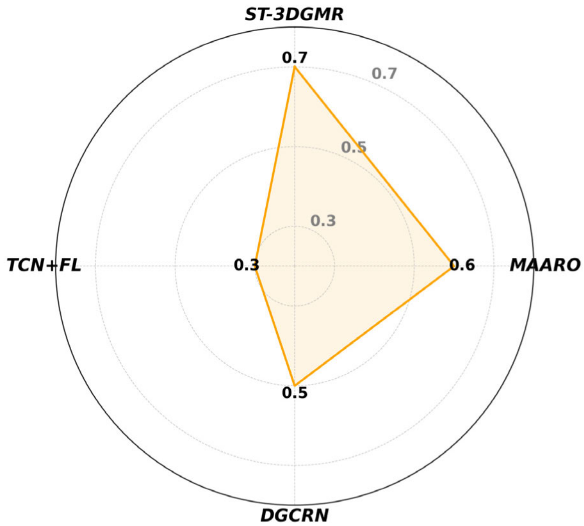

FIGURE 5. Analysis of data privacy via differential privacy.

To analyze the models' data privacy condition, we utilized two anonymization techniques:differential privacy and k-anonymity. Differential privacy is defined by the privacy loss parameter (ε), which is used to compute the added noise to assure privacy. In Figure 5, a spider chart displays the four differing models in terms of the measure of privacy loss (through an ε parameter). If the ε value is small, privacy is considered to be higher. This implies less chance of private data leaking from the model. In this respect, the TCN+FL approach depicted the maximum level of privacy with the lowest possible level of ε (0. 3), implying that it provides a minimum level of data leakage and ensures the maximum data protection of the system. This is indispensable in cases where the integrity of hidden information is of utmost importance.

Similarly, whereas ST-3DGMR presents a high risk of data leaking with an ε of 0.7, it might be good to have a moderate level of privacy. Both MAARO and DGCRN fall between these extremes, with ε values of 0.6 and 0.5, respectively, indicating progressively better privacy preservation. This gradation in ε values highlights the effectiveness of the privacy-preserving techniques applied in the TCN+FL model, underscoring its suitability for scenarios demanding stringent data privacy.

The k-Anonymity stands out among the widely used privacy-preserving data anonymization methods and helps protect individual identities within the data. K-anonymity is considered the main rule, and it is based on the fact that each record cannot be different from at least k-1 other records regarding specific attributes that identify individuals, quasi-terms or identifiers. Quasi-identifiers are variables that are indistinguishable in their characteristics unless they are pooled/matched with other identifiers. Figure 6 depicts that the k-anonymization level varies in different modes. ST-3DGMR, with a k-anonymity of 5, is an instrument that presents the lowest level of private autonomy. It means that each document is complicated to distinguish from the four other documents in a group. With the provision of k-anonymity 10, DGCRN doubles its in-distinguishability. It makes individual records indistinguishable from up to nine other records, which means higher privacy measures and a lower possibility of identifying unique individuals. MAARO builds on that by achieving 15-level protection of k-anonymity, meaning that records are shuffled amongst fourteen other ones, preserving the system's security against revealing identities. TCN+FL provides the most excellent protection level; a k-anonymity of 20 is attained, which makes each entry hidden in nineteen others' data. At the most advanced level, "unlinkability" means that the data is so protected that it is considerably more challenging to decipher individual records, and this mode is by far the most suitable for applications where anonymity and confidentiality should be absolute. The privacy-preserving approach to the data models reflects this growing level. Higher levels of anonymity are more robust and provide better data protection options, while the utility of data is kept at a sufficiently high level.

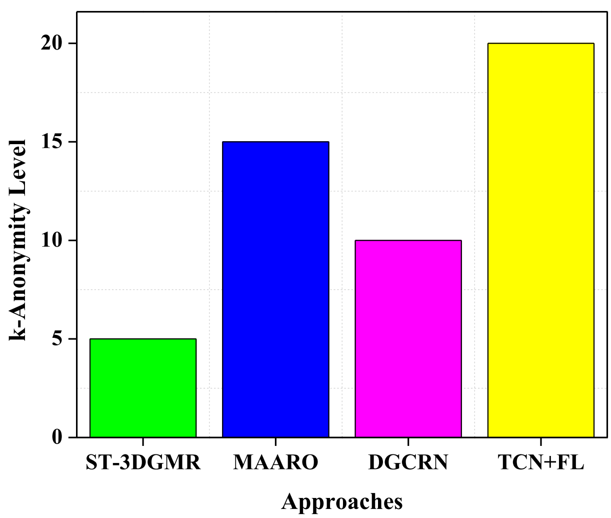

FIGURE 6. Evlauation of data privacy via k-anonymity.

Moreover, Table 8 clearly illustrates the relative effectiveness of the four traffic prediction methodologies as emergency response time (ERT) [22] reducers. The decrease in ERT is an essential indicator of traffic control because it directly relates to the performance of emergency services.

The findings imply that the TCNs+FL approach outperforms other approaches by more than 15% in reducing the ERT. Such an outcome can be attributed to the powerful temporal pattern recognition abilities of the TCNs, which benefit from the distributed nature of FL.

TABLE 8. Analysis of ERT reduction.

Model	ERT Reduction (%)	Error Margin
ST-3DGMR	25%	±5%
MAARO	30%	±4%
DGCRN	35%	±3%
TCNs+FL	42%	±2%

TCNs excel at processing sequential data and can also extract detailed temporal relationships. The model is supplemented with FL components that allow various nodes, such as traffic cameras and sensors, to participate in the learning process without unifying the data sources. These results in an overall model that can better forecast the traffic flows and, as a result, provide more appropriate route recommendations for emergency vehicles.

The fact that the error bars for TCNs+FL are smaller indicates a greater certainty in the methodology's performance, paving the way to robustness and reliability of ERT improvement. This reliability is vital in critical situations when every second matters a loT.

V. POTENTIAL IMPLICATIONS

The results of our study, which demonstrate overwhelming improvement in traffic flow and congestion management in the case of TCNs combined with FL, have been of enormous value for future urban traffic management strategies. Our study elucidates the system's capability to balance privacy intrusion with efficiency in traffic management that could be taken as a basis for scaling smart city projects, which in turn might help create policies to enhance urban living conditions. Moreover, the advancement of such technology in the future may include the development of real-time adaptive learning algorithms and sensor networks that are efficient for smooth road traffic and dynamic response to urban conditions. In this context, the foresight validates the effectiveness of the current approach and points out the path for further development of robust intelligent traffic management systems.

Moreover, the traffic control and management decisions obtained from the TCN-FL predictions can be integrated into the subsequent domains. If it is able to determine those traffic rates, congestion phases, and mobility routes, then it ought to continually adjust the traffic signals' timings in a bid to minimize traffic jams and, at the same time, enhance the overall traffic circulation. For instance, during a time when there is heavy traffic, the system can make sure that the green light stays on for a longer time on some important highways and also ensure that the sequences of sets are made in such a way that there is the creation of gaps. Therefore, the predictions enable the actual positioning of the vehicles within lesser imbricate zones to enhance the load sharing of the traffic within the systems. The system also has the possibility to give priority to emergency vehicles and guide them to the roads with the most minor traffic and the shortest possible time, though it actually enhances response times. Also, the FL framework mitigates the centralization issue because the model continuously improves every time each traffic sensor or camera in the entire smart city contributes to their learning and updates the combined model as per the existing traffic condition complexities.

VI. CONCLUSION AND FUTURE WORK

The integration of TCNs with FL is a unique and efficient method for the management of urban traffic. Through its distributed learning model, the combined methodology prioritizes data privacy and harnesses TCNs' predictive abilities to analyze and forecast traffic conditions. The comparative analysis reveals that TCNs+FL outperform the other approaches, such as ST-3DGMR, MAARO, and DGCRN, by a significant margin in key performance indicators. TCNs+FL significantly reduce the MAE and provide more accurate predictions, which are crucial for effective traffic management and emergency response. The methodology has been shown to cut the average commute times by a quarter, reflecting its ability to reduce congestion and improve urban traffic dynamics. In addition, its impressive achievement of a 15% reduction in the response time for emergencies represents a significant improvement in transport services. Together, these developments represent the capabilities of TCNs+FL towards transforming traffic management systems and paving the way for more adaptive smart city infrastructures.

Real-time dynamic data sources, like social media and IoT sensors, could be integrated as a future expansion of the traffic prediction model to enrich it further. The latency can be reduced by utilizing these progresses on edge computing, thereby allowing for real-time traffic adjustment.

REFERENCES

[1] S. Porru, F. E. Misso, F. E. Pani, and C. Repetto, "Smart mobility and public transport: Opportunities and challenges in rural and urban areas," *J. Traffic Transp. Eng., English Ed.*, vol. 7, no. 1, pp. 88–97, Feb. 2020.

[2] Z. Mahrez, E. Sabir, E. Badidi, W. Saad, and M. Sadik, "Smart urban mobility: When mobility systems meet smart data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6222–6239, Jul. 2022.

[3] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, Jan. 2019.

[4] J. Rosen and L. F. Alvarez León, "The digital growth machine: Urban change and the ideology of technology," *Ann. Amer. Assoc. Geographers*, vol. 112, no. 8, pp. 2248–2265, May 2022.

[5] A. A. Musa, S. I. Malami, F. Alanazi, W. Ounaies, M. Alshammari, and S. I. Haruna, "Sustainable traffic management for smart cities using Internet-of-Things-oriented intelligent transportation systems (ITS): Challenges and recommendations," *Sustainability*, vol. 15, no. 13, p. 9859, Jun. 2023.

[6] J. Bharadiya, "Artificial intelligence in transportation systems a critical review," *Amer. J. Comput. Eng.*, vol. 6, no. 1, pp. 34–45, Jun. 2023.
[7] A. K. Shaikh, A. Nazir, N. Khalique, A. S. Shah, and N. Adhikari, "A new approach to seasonal energy consumption forecasting using temporal convolutional networks," *Results Eng.*, vol. 19, Sep. 2023, Art. no. 101296.
[8] M. S. Farooq, R. Tehseen, J. N. Qureshi, U. Omer, R. Yaqoob, H. A. Tanweer, and Z. Atal, "FFM: Flood forecasting model using federated learning," *IEEE Access*, vol. 11, pp. 24472–24483, 2023, doi: 10.1109/ACCESS.2023.3252896.
[9] J. Rui and F. Othengrafen, "Examining the role of innovative streets in enhancing urban mobility and livability for sustainable urban transition: A review," *Sustainability*, vol. 15, no. 7, p. 5709, Mar. 2023, doi: 10.3390/su15075709.
[10] A. O. Ajayi,and H. Kumkal, "Optimising urban road transportation efficiency: AI-driven solutions for reducing traffic congestion in big cities," Tech. Rep, doi: 10.13140/RG.2.2.16130.45763. [Online]. Available: https://www.researchgate.net/publication/376517276
[11] N. V. Cuong and M. T. Aziz, "AI-driven vehicle recognition for enhanced traffic management: Implications and strategies," *AI, IoT Fourth Ind. Revolutions Rev.*, vol. 13, no. 7, pp. 27–35, Jul. 2023.
[12] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023, doi: 10.1145/3532611.
[13] Y. Djenouri, A. Belhadi, G. Srivastava, and J. C.-W. Lin, "Hybrid graph convolution neural network and branch-and-bound optimization for traffic flow forecasting," *Future Gener. Comput. Syst.*, vol. 139, pp. 100–108, Feb. 2023.
[14] R. He, Y. Xiao, X. Lu, S. Zhang, and Y. Liu, "ST-3DGMR: Spatio-temporal 3D grouped multiscale ResNet network for region-based urban traffic flow prediction," *Inf. Sci.*, vol. 624, pp. 68–93, May 2023.
[15] S. Dikshit, A. Atiq, M. Shahid, V. Dwivedi, and A. Thusu, "The use of artificial intelligence to optimize the routing of vehicles and reduce traffic congestion in urban areas," *EAI Endorsed Trans. Energy Web*, early access, 2023, doi: 10.4108/ew.4613.
[16] Y. Lu, S. Yao, and Y. Yao, "Research on the intelligent assignment model of urban traffic planning based on optimal path optimization algorithm," *Sci. Program.*, vol. 2022, pp. 1–7, Jan. 2022.
[17] M. Trouve, G. Lesteven, and F. Leurent, "Worldwide investigation of private motorization dynamics at the metropolitan scale," *Transp. Res. Proc.*, vol. 48, pp. 3413–3430, Jan. 2020, doi: 10.1016/j.trpro.2020.08.113.
[18] F. Nourmohammadi, M. Mansourianfar, S. Shafiei, Z. Gu, and M. Saberi, "An open GMNS dataset of a dynamic multi-modal transportation network model of Melbourne, Australia," *Data*, vol. 6, no. 2, p. 21, Feb. 2021.
[19] A. Loder, L. Ambühl, M. Menendez, and K. W. Axhausen, "Understanding traffic capacity of urban networks," *Sci. Rep.*, vol. 9, no. 1, Nov. 2019, Art. no. 16283.
[20] M. Mazziotta and A. Pareto, "Normalization methods for spatio-temporal analysis of environmental performance: Revisiting the min–max method," *Environmetrics*, vol. 33, no. 5, May 2022, Art. no. e2730.
[21] X. Chen, H. Chen, Y. Yang, H. Wu, W. Zhang, J. Zhao, and Y. Xiong, "Traffic flow prediction by an ensemble framework with data denoising and deep learning model," *Phys. A, Stat. Mech. Appl.*, vol. 565, Mar. 2021, Art. no. 125574, doi: 10.1016/j.physa.2020.125574.
[22] K. M. K. Raghunath and N. Rengarajan, "Response time optimization with enhanced fault-tolerant wireless sensor network design for on-board rapid transit applications," *Cluster Comput.*, vol. 22, no. S4, pp. 9737–9753, Dec. 2017.
[23] Y. Ren, D. Zhao, D. Luo, H. Ma, and P. Duan, "Global-local temporal convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1578–1584, Feb. 2022.
[24] H. Huang, Z. Hu, Y. Wang, Z. Lu, X. Wen, and B. Fu, "Train a central traffic prediction model using local data: A spatio-temporal network based on federated learning," *Eng. Appl. Artif. Intell.*, vol. 125, Oct. 2023, Art. no. 106612.
[25] S. Chavhan and P. Venkataram, "Commuters' traffic pattern and prediction analysis in a metropolitan area," *J. Vehicle Routing Algorithms*, vol. 1, no. 1, pp. 33–46, Mar. 2018.
[26] Y. Cui, Y. Yu, and D. Wang, "Impact of the link length on the delay in two-way signal coordination," *IEEE Access*, vol. 9, pp. 130823–130833, 2021.
[27] Y. Yu, Y. Cui, J. Zeng, C. He, and D. Wang, "Identifying traffic clusters in urban networks based on graph theory using license plate recognition data," *Phys. A, Stat. Mech. Appl.*, vol. 591, Apr. 2022, Art. no. 126750.
[28] Y. Cui, Y. Yu, Z. Cai, and D. Wang, "Optimizing road network density considering automobile traffic efficiency: Theoretical approach," *J. Urban Planning Develop.*, vol. 148, no. 1, Mar. 2022, Art. no. 04021062.

K. M. KARTHICK RAGHUNATH (Member, IEEE) received the B.Tech. degree in information technology from Anna University, Chennai, in 2008, the M.E. degree in pervasive computing technology from Anna University (BIT Campus), in 2011, and the Ph.D. degree from Anna University, in 2019. He is currently an Associate Professor with the Computer Science and Engineering Department, JAIN (Deemed-to-be University), Bengaluru, India. With more than one decade of experience in teaching, his areas of specialization include pervasive computing, artificial intelligence, the IoT, data science, and WSN. He has six patents in IPR and has published approximately more than ten articles in reputed international journals. He has authored "*Response Time Optimization in Wireless Sensor Networks*," published in LAP LAMBERT Academic Publishing, U.K. He has published numerous papers in national and international conferences. He is an Active Member of IEEE, ACM, I.E.(I), IACSIT, and IAENG. He has organized several national workshops and technical events. He is regularly invited to deliver lectures in various programs for imparting skills in research methodology to students and research scholars. He served as an Editor/Reviewer for Springer, Elsevier, Wiley, IGI Global, Emerald, and ACM.

C. ROHITH BHAT received the Engineering degree in CSE from the Dr. M.G.R. Engineering College, Chennai, and the master's and Ph.D. degrees from the Dr. M.G.R. Educational and Research Institute. His passion for academics and research has made him the right person to be in the Academic Sector. He has got immense interest in training the students for their placements and he is currently a Professor with the Saveetha School of Engineering (SIMATS), Chennai. He has got patents to his credit and had published journals in journals of repute. His interests research include process mining, the IoT, machine learning, deep learning, and to name a few. His academic interest has made him to pursue his research work on process mining and classification in general.

VENKATESAN VINOTH KUMAR (Member, IEEE) is currently an Associate Professor with the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, India. He is also an Adjunct Professor with the School of Computer Science, Taylor's University, Malaysia. He is having more than 12 years of teaching experience. He has organized various international conferences and delivered keynote addresses. He has published more than 100 research papers in various peer-reviewed journals and conferences. His current research interests include wireless networks, the Internet of Things, machine learning, and big data applications. He has filled nine patents in various fields of research. He is a Life Member of ISTE and a Professional Member of IEEE. He is an Associate Editor of *International Journal of e-Collaboration* (IJeC) and *International Journal of Pervasive Computing and Communications* (IJPCC) and an editorial member of various journals.

VELMURUGAN ATHIYOOR KANNAN received the degree in computer science and engineering from the Arunai Engineering College, University of Madras, Chennai, Tamil Nadu, India, the Master of Technology degree in computer science and engineering from the Dr. M.G.R. Educational and Research Institute (Deemed University), Chennai, and the Ph.D. degree in wireless sensor networks in computer science and engineering from the St. Peter's Institute of Higher Education and Research (Deemed University), Chennai. He is currently a Professor with the School of Computing, Koneru Lakshmaiah Education Foundation (Deemed to be University), Vijayawada, Andhra Pradesh, India. He has published a number of research papers in preferred journals, chapters in books, and conferences. His research interests include wireless sensor networks, the Internet of Things, artificial intelligence, and machine learning.

T. R. MAHESH is currently an Associate Professor and the Program Head of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bengaluru, India. He has to his credit more than 100 research articles in Scopus and SCIE indexed journals of high repute. He has been the editor of books on emerging and new age technologies with publishers, such as Springer, IGIGlobal, and Wiley. His research interests include image processing, machine learning, deep learning, artificial intelligence, the IoT, and data science. He served as a reviewer and a technical committee member for multiple conferences and journals of high reputation.

K. MANIKANDAN received the B.E. degree (Hons.) in CSE from Vellore Engineering College (presently Vellore Institute of Technology (VIT), affiliated to University of Madras), in 2004, the Master of Engineering (M.E.) degree (Hons.) in CSE from the Sona College of Technology, Salem, affiliated to Anna University, in 2006, the M.B.A. degree in systems from Alagappa University, in 2008, and the Ph.D. degree in wireless networks from VIT, in 2015. He is currently a Professor with the School of Computer Science and Engineering, VIT, Vellore. He has more than 18 years of teaching and research experience. He has presented a number of papers at the national and international conferences and has published more than 50 articles in reputed journals. His research interests include wireless networks, cloud computing, the Internet of Things (IoT), artificial intelligence, and data science. He is a Life Member of CSI and IAEng. He received the Best Techno Faculty Award from ICTACT, Government of Tamil Nadu, in 2014.

N. KRISHNAMOORTHY received the Ph.D. degree from Anna University, Chennai, in 2015. He is currently an Associate Professor with the Department of Software Systems and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India. He also got various funds from AICTE, DRDO, MoES, DeitY, DBT, NBHM, and ICSSR, for conducting various research projects, skill development programs, and also seminars and workshops. He is a Life Member of the Computer Society of India. He is involved in various aspects of life-style related mental disorders and engaged in finding preventive as well as therapeutic solutions from indigenous resources.