Decentralized Federated Learning for Real-Time Traffic Prediction in Smart Cities

***Abstract*—Decentralized Federated Learning (DFL) is similar to a Machine Learning (ML) approach, which works in a distributed manner. This approach enables collaborative learning prediction without the need to share any raw data among different entities. Existing traffic prediction approaches by ML or Deep Learning have an excellent success rate. However, this raises a privacy concern as the data sets contain a large amount of user personal data. That's why the DFL approach is ideal for such predictions, as it offers an efficient solution by utilizing the capabilities of multiple learning agents while solving the concern by preserving data privacy. By applying Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) in a federated learning framework, the algorithm can learn from the vast amounts of data that are gathered from numerous sources, including sensors, modern vehicles, infrastructures, etc., of smart cities to improve traffic predictions and facilitate improved transportation systems. With this method, real-time traffic prediction is made possible in smart cities without sacrificing data security or depending on centralized data storage.**

***Index Terms*—Decentralized Federated Learning, Machine Learning, Recurrent Neural Network, Long Short Term Memory, collaborative learning**

I. Introduction

Machine learning (ML) and deep learning have substantially improved the potential for predictive analysis in the current era of data-driven decision-making. These advancements play a significant role in Traffic forecasting by designing effective transportation networks and reducing urban congestion. However, the privacy of sensitive user data is frequently compromised by existing approaches to traffic prediction, which trade off prediction accuracy. In order to address the difficulties of traffic prediction in smart cities, this paper explores the field of Decentralized Federated Learning (DFL). DFL is a distributed machine learning model that encourages collective learning without requiring different entities to share raw data. Large amounts of data are produced by the widespread use of contemporary transportation systems in smart cities from a variety of sources, including sensors, connected vehicles, and infrastructure elements. While traditional ML or Deep Learning algorithms have demonstrated impressive success rates in predicting traffic patterns, they frequently give rise to privacy-related problems. The DFL approach respects the crucial need to protect data privacy while leveraging the collective intelligence of various learning agents, in contrast to conventional models that aggregate data in a centralized manner. This study provides a thorough examination of the advantages of combining DFL and RNN algorithms to improve traffic prediction in smart cities. By leveraging the inherent capabilities of RNNs in processing sequential data and by dispersing the learning process through DFL, the model can give real-time insights without sacrificing on data security. LSTM is also a type of RNN that has a more complex architecture with specialized memory cells and gates that can capture and retain long-term dependencies in the data. It is particularly effective in learning patterns and dependencies in time-series data. This paper aims to contribute to the developing landscape of smart city infrastructure by pursuing more precise, effective, and privacy-aware traffic predictions. We anticipate a future in which urban transportation systems seamlessly integrate real-time insights with data privacy, ultimately resulting in improved mobility experiences for city people. To achieve this, we combine the strengths of Decentralized Federated Learning with RNN-based algorithms.

II. Literature Survey

In this section, we discuss a few related works that have already been done on network traffic prediction, the use of federated learning, prediction, and privacy schemes, traffic forecasting, and the use of short-term traffic prediction methods (parametric, non-parametric, and artificial intelligence) in various networks and scenarios.

The study by Sepasgozar et al. introduces "Fed-NTP," a novel approach for predicting network traffic flow in Vehicular Ad-Hoc Networks (VANETs) while safeguarding data privacy [1]. It combines federated learning (FL) and the Long Short-Term Memory (LSTM) algorithm to make accurate predictions while keeping data local and private. VANETs are self-organized wireless networks among vehicles on roads, enabling vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. Only model updates are shared for aggregation, ensuring privacy, lower bandwidth usage, and improved energy efficiency. The Fed-NTP model combines the VANET environment, LSTM for local training, and FL to achieve accurate traffic flow predictions while maintaining data privacy in a distributed manner. Combining the strengths of LSTM and FL algorithms, the proposed model achieved accurate predictions while preserving data privacy. Furthermore, this study opens up opportunities for further exploration in implementing AI algorithms for network traffic prediction in advanced wireless networks which can further improve the accuracy and efficiency of network traffic prediction.

The results showed that the Fed-NTP model outperformed the other algorithms in terms of prediction accuracy. It achieved the lowest values for MAE, MSE, RMSE, and MAPE, indicating smaller errors between the predicted values and the actual values. Additionally, the Fed-NTP model had

the highest R^2-Score, indicating a better fit to the actual data indicating the effectiveness of the Fed-NTP model in accurately predicting network traffic flow in the VANET environment while preserving data privacy.

We find in the paper by Zheng et al. that FL holds promise for privacy-preserving machine learning in smart cities, yet unresolved issues remain for successful implementation [2]. The authors reference several researchers' works and propose that FL has the potential to enhance urban infrastructure intelligence. The work of Liu and Zhang demonstrates privacy protection methods. Handling various data distributions, encouraging data quality, and controlling data fluctuation are challenges. Medical problems include medical data heterogeneity, dirty data detection, and model accuracy, while communication challenges include security, local learner selection, and algorithm efficiency. Strengthening security, optimizing algorithm performance for low-power devices, and evaluating FL's applications in healthcare and communication inside smart cities are some future directions suggested by the authors.

The article by Akallouch et al. introduces a novel method called 'Prediction and Privacy Scheme for Traffic Flow Estimation on the Highway Road Network' [3]. To solve the problem of anticipating traffic flow while preserving data privacy, this system uses local differential privacy (LDP) and federated learning. The report discusses relevant research on federated learning, safe traffic flow prediction, and privacy-preserving machine learning. It emphasizes the use of local differential privacy combined with federated learning in the FL-LDP system, as well as the use of differential privacy to secure data in machine learning. This method employs LDP to modify gradients, allowing customers to participate in model training without disclosing raw data. The results of the experiments show that FL-LDP offers precise forecasts while greatly lowering privacy risks.

The paper discusses the predictions of urban traffic in smart cities. The authors suggest a cutting-edge strategy utilizing Edge Intelligence, Federated Learning, and Continual Learning to address the problems caused by expanding urbanization [4]. The authors suggest a cutting-edge strategy utilizing Edge Intelligence, Federated Learning, and Continual Learning to address the problems caused by expanding urbanization. Their approach eliminates delays and privacy concerns by running Machine Learning directly on edge devices, as opposed to standard approaches that rely on cloud processing. They unveil the Federated peer-to-peer Continual Learning (FpC) algorithm, which uses networked sensor data to cooperatively train a global traffic prediction model. Compared to centralized approaches, this strategy increases accuracy and energy economy. They also provide the FpC with an early stopping (FpCes) approach, which uses less energy to achieve an equivalent level of precision. According to the study's findings, collaborative edge-based training offers the potential for precise traffic prediction in smart cities, as well as the possibility of energy savings and increased privacy.

The authors propose 'BFRT: Blockchained Federated Learning for Real-Time Traffic Flow Prediction,' a unique architecture that combines blockchain with federated learning (FL) to forecast real-time traffic flow while preserving decentralization and data privacy] [5]. They outline the shortcomings of traditional central systems before introducing the two key elements of BFRT: federated learning and a permission blockchain network. Without exchanging raw data, FL enables edge devices to work together to build a global traffic flow prediction model, and the blockchain assures secure, transparent FL transactions.

III. Proposed Methodology

As it has been already proven that gathering real-time traffic data with these features is extremely hard and time-consuming, we are working with randomly generated data. So, some preprocessing with these datasets as these data are not ready yet to be run by our models is inevitable.

So now we will discuss how we can enable us to give traffic predictions. There are so many algorithms that work nicely and efficiently with federated learning but when it comes to real-time traffic prediction the number of algorithms is much less which are eligible to work collaboratively with federated learning. In the primary section, we considered many machine learning algorithms and deep learning algorithms such as vanilla RNN, CNN, LSTM, random forest, ARIMA, linear regression, and many more. Here we are going to discuss some algorithms that we are predicting that go well with federated learning.

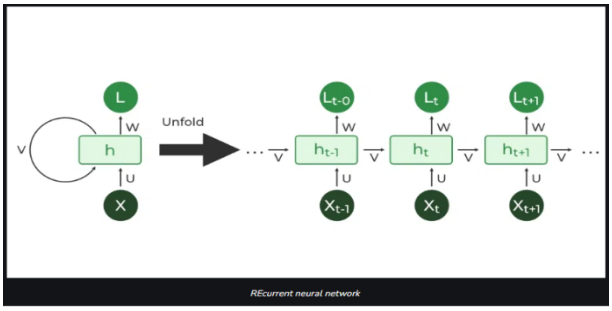

Fig. 1. RNN

A. Machine Learning Models

1) RNN: Recurrent Neural Networks (RNN) are mainly designed to work with sequential data or capture data sequences by maintaining the internal hidden states that help it to capture temporal dependencies. It remembers data from earlier steps while processing inputs one step at a time. In our research, we can utilize RNN by implementing its algorithm into every local branch. Therefore, each branch will have its own RNN model which will work on local traffic data. Then these forecasts may then be formed into several groups by taking into account the contributions from all branches, once these local branches have been aggregated by sharing their parameters or gradients. The decentralized system may use this method to preserve data confidentiality and privacy while collectively learning and utilizing the temporal trends in local traffic data for real-time prediction.

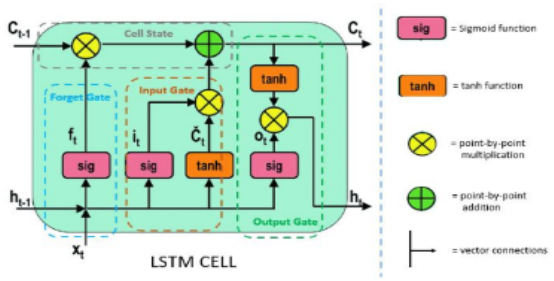

Fig. 2. LSTM CELL

2) LSTM: LSTM is a kind of recurrent neural network (RNN) that is different from other feed-forward neural networks as it has the ability to store context information that traditional RNNs can struggle with due to the vanishing gradient problem. In the very beginning, LSTM showed promising results as a neural network model that is capable of learning order dependence in sequence prediction problems. But now LSTM models are evolving and already shown massive success in various types of research fields. In general, an LSTM model uses 3 types of gates to control what information it should forget and what it should remember and update that information in its memory state which allows it to capture patterns and long-term dependencies in data. The three different types of gates are -

Input Gate - This input gate generally receives two arguments known as current state X(t) and the previous hidden state h(t-1) as input. The general values are in the range of 0-1 labeling 0 as 'important' and 1 as 'not-important'. But these values and range can be modified as per requirement and we will modify these values as per our need. The tan(h) function will then receive identical data from the hidden state and current state.

Output Gate - Which earlier stages' pertinent data is required is decided by the forget gate. The output gates complete the next concealed state, while the input gate determines what pertinent information can be supplied from the current stage.

Forget Gate - The forget gate determines what information must be remembered and what can be forgotten.

With the help of the LSTM model and decentralized federated learning, we can get great benefits in real-time traffic prediction. The decentralized technique is complemented by LSTM's ability to capture temporal connections in sequential data, allowing numerous nodes to cooperatively train a model without exchanging raw data. This technique ensures data security as local nodes provide a number of traffic patterns [6].

B. Parallel, Distributed, and High-Performance Computing Tools and Other Frameworks

1) Apache Spark: Apache Spark is a powerful framework or engine for running large data engineering, data science, and machine learning tasks on a node or clustered servers. It uses in-memory caching to speed up query execution when processing large data clusters. Furthermore, for data processing needs, Spark provides a variety of libraries and tools such as batch processing, machine learning, and so on. Apache Spark can assist in distributing and parallelizing computations across several data sources or locations while protecting data privacy in the context of federated learning for real-time traffic prediction [7].

2) TensorFlow: A complete open-source machine learning platform is called TensorFlow. The lesson concentrates on utilizing a specific TensorFlow API to create and train machine learning models, despite the fact that TensorFlow is a robust framework for managing all parts of a machine learning system. A straight route to production has always been offered by TensorFlow. No matter what language or platform you choose, TensorFlow makes it simple to train and deploy your model on servers, edge devices, or the web [8].

C. Data Collection and Prepossessing

In this section, we talk about how the dataset used for the research has been collected and which of them has been used here.

1) Data Types and Sources: The dataset that has been used in this research paper is randomly generated for the purpose of experimenting, as collecting a huge amount of data would require additional time and funding. Our fabricated dataset was created by utilizing a combination of Python libraries like NumPy and pandas along, with algorithms tailored to simulate real-life traffic conditions. We have taken the following data into consideration for the implementation of our model.

Traffic Flow Data:
Real-time traffic metrics can be gathered from road segments within the city including information about traffic flow, congestion levels, speed, and occupancy rates.

Spatial Data:
Geospatial information that describes the road network is taken into account. This includes details about roads, intersections, and their corresponding geographical coordinates. Additionally, characteristics like road type, number of lanes, and speed limits are also considered.

Temporal Data:
Timestamps are recorded to indicate the time when data is collected. This is particularly useful as roads may experience varying levels of traffic on days.

Population and Demographic Data:
Data depicting population density, distribution, and demographic characteristics within the city provide insights into traffic patterns across different areas. This information assists in gaining an understanding of populated regions.

2) Data Collection and Generation: Due to the challenges involved in collecting data for our research purposes, we plan to create a dataset that encompasses the characteristics of the aforementioned data types. This simulated dataset will be specifically designed to mimic real-world traffic scenarios in cities.

3) Data Preprocessing and Augmentation: In order to ensure the quality of our simulated dataset we will carry out steps to preprocess the data, such, as cleaning, removing outliers, and scaling features. Furthermore, if applicable we will also explore data augmentation techniques to enhance the diversity of our dataset. Some of the steps for the cleaning of the dataset are elaborated below. Normalization of Data:
We standardized features with scales to bring them to a common range typically between 0 and 1 using Min Max scaling. Encoding of Data: variables, such as road types and speed limits were transformed into formats using methods like one hot encoding. Time Alignment:
We standardized timestamps to ensure that all data points could be arranged in order making time series analysis more convenient. Enhancement of Features:
We derived features from existing ones to capture intricate relationships. For instance we calculated speeds, over time intervals or aggregated congestion levels based on geographical areas.

4) Ethical Consideration: We acknowledge the significance of upholding standards even though we are not working with real-world data. Our simulated dataset will be carefully developed to avoid introducing biases or causing any harm.

5) Summary Statistics: To provide an overview of the dataset's characteristics and scale we will present preliminary summary statistics. Although hypothetical, in nature these statistics will illustrate the distribution patterns and overall scope of our dataset.

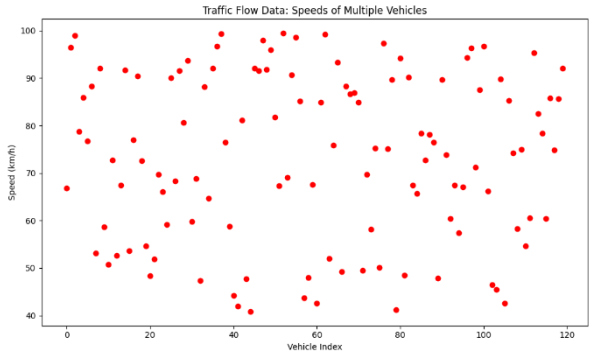

Fig. 3. Speed Data

D. Model Proposal

Federated learning enables collaborative model training across scattered data sources, which is especially helpful when working with private or real-time data. Also, using Machine Learning Algorithms is the best possible solution to learning and predicting traffic in smart cities. However, we cannot use

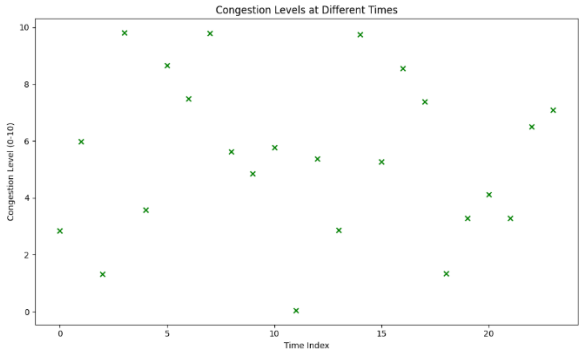

Fig. 4. Congestion Data

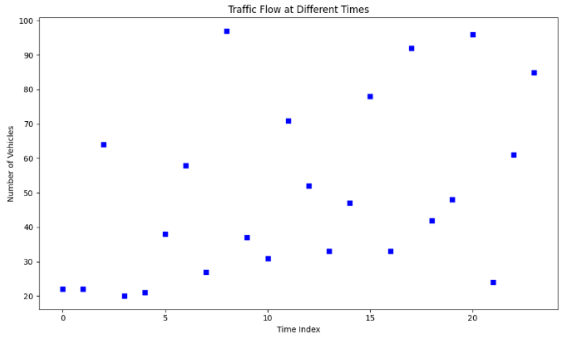

Fig. 5. Traffic Flow Data

ML in a decentralized manner and cannot handle large data. That's where Apache Spark comes into play. This enables us to handle the cluster of data and large data sets by scaling the machine-learning activities. With the help of the Spark MLlib library, we distribute data across nodes in the cluster and implement LSTM using TensorFlow. The data for the LSTM structure for a traffic flow prediction consists of traffic flow, traffic speed, traffic congestion, and traffic occupancy rate. The expected status of the present traffic flow, whether it is congested or not, is what the LSTM structure produces as its output. As stated above, we will train separate RNN models for each of the road segments that we are calling branches. These RNN models will work on the historical data of traffic flow and many other data points as per design including the activation functions, hidden layers, and sequence length for that particular road segment or branch. These models will take inputs from the data set and will give predicted traffic flow for the next time step for that particular branch. So, these RNN models will give results based on local traffic data which will be aggregated and used as input data for the LSTM model which will be placed at the zone level. We are considering the fact that a zone will consist of multiple branches. The goal of these LSTM models is to accumulate the predictions from each branch coming from RNN models. So, now the better RNNs or LSTM models will detect longer sequences in the RNN outputs and will result in better predicted answers.

In the figure "Fig. 6" we describe how to calculate the

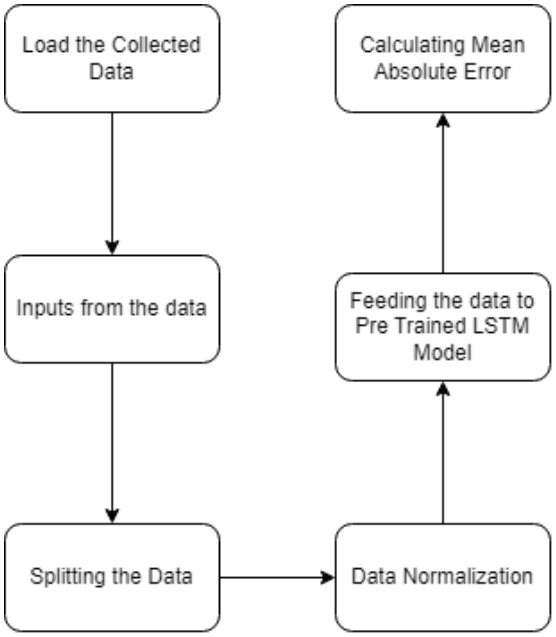

Fig. 6. Proposed Model

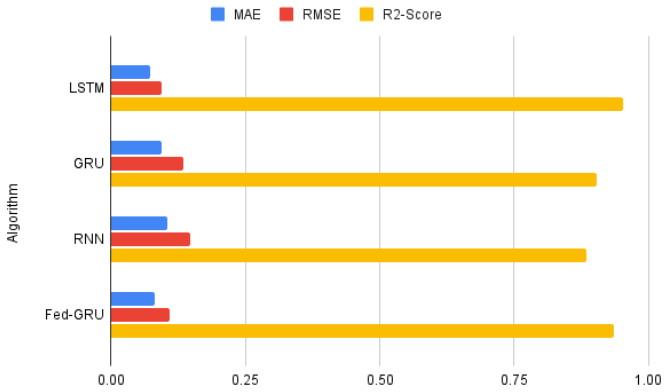

Fig. 7. Bar Chart

TABLE I
PERFORMANCE METRICS OF DIFFERENT ALGORITHMS

Algorithm	MAE	RMSE	R2-Score
LSTM	0.074	0.095	0.952
GRU	0.094	0.134	0.904
RNN	0.106	0.148	0.885
Fed-GRU	0.082	0.110	0.935

prediction of a sample collected data using pre-trained LSTM models. At first, we load the data and identify the inputs of the data which are: traffic flow, traffic speed, traffic congestion, and traffic occupancy rate. Then we split the data set as 80% training data and 20% testing data. After that, we normalize all the values by scaling. Then we structure the LSTM model by defining the number of epochs and batch size. Now, we train the model and calculate predictions. After all of this, we measure the difference between actual values and predicted values which is the Mean Absolute Error(MAE). The lower the MAE, the better the accuracy of the model.

IV. Result and Analysis

After a lot of investigation and diligent examination of prior research papers, we have gathered some algorithm scores and their correct prediction rate. Among the algorithms, some notable ones are RNN [9], GRU [6], Fed-GRU [10], and LSTM [11]. From all the studies one significant study about the LSTM shows remarkable performance from the other models in terms of handling Traffic Prediction. The author explored the algorithm's potential in such a manner that we not only found the performance result but also understood the effectiveness of using this model in the application of real-time traffic prediction. From the papers, we have analyzed a bar diagram which is shown below "Fig. 7".

Here, the lower the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) scores are, the better the algorithm is. And the higher the R2- Score is, the better.

V. Conclusion

In this era of ML and AI, humankind still faces the problem with traffic prediction as the general structured ML and deep learning algorithms use centralized data for their prediction. Also, to some extent, the accuracy level of the predictions is not always high. So, this research endeavors a dynamic approach for traffic prediction that uses the collaboration of LSTM and RNN models with Apache Spark by following the convention of the federated learning approach. With the proper utilization of RNNs working with every branch, LSTM at zone level, and Apache spark, we are getting much better results than the other conventional approaches. Besides these, as this research follows a federated learning approach, users' data security is also maintained. In this era of digitalization and urbanization, the usage of our research is far-reaching. Traffic management agencies, transportation engineers, and Urban planners will have the authority to a powerful toolkit that will make them eligible to predict traffic dynamics with a level of precision that was a dream before. Our study offers a shining example of innovation as smart cities continue to evolve, easily integrating with the idea of data-driven urban planning and sustainable transportation systems. In a nutshell, this work has unveiled a cutting-edge policy regarding traffic prediction that merges hierarchical deep learning methods with Apache Spark in a beautiful manner. Our research unfolds the way for more effective, smarter, and seamlessly linked urban backgrounds as we traverse the undiscovered areas of urban mobility—a testament to the transformational power of multidisciplinary cooperation and cutting-edge technology.

References

[1] S. S. Sepasgozar and S. Pierre, "Fed-ntp: A federated learning algorithm for network traffic prediction in vanet," *IEEE Access*, vol. 10, pp. 119 607–119 616, 2022.

[2] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang, B. Liu, and K. Li, "Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges," *Connection Science*, vol. 34, no. 1, pp. 1–28, 2022. [Online]. Available: https://doi.org/10.1080/09540091.2021.1936455

[3] M. Akallouch, O. Akallouch, K. Fardousse, A. Bouhoute, and I. Berrada, "Prediction and privacy scheme for traffic flow estimation on the highway road network," *Information*, vol. 13, no. 8, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/8/381

[4] 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)*, 2023, pp. 1–8.

[5] C. Meese, H. Chen, S. A. Asif, W. Li, C.-C. Shen, and M. Nejad, "Bfrt: Blockchained federated learning for real-time traffic flow prediction," in *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2022, pp. 317–326.

[6] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.

[7] I. Pointer, "What is apache spark? the big data platform that crushed hadoop," Mar 2023. [Online]. Available: https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html

[8] "First steps with tensorflow toolkit," https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit, accessed on 26th July 2023.

[9] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[10] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.

[11] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X15000935