

Research Article

Model Aggregation Federated Learning Approach for Vehicular Traffic ForecastingSavita Lonare^{1,*} and R. Bhramaramba²¹Pimpri Chinchwad College of Engineering, Pune²Department of Computer Science and Engineering, GITAM, Visakhapatnam, India

Received 8 October 2020; Accepted 1 July 2021

Abstract

To make the Intelligent Transportation System (ITS) more efficient and robust, researchers are working hard. Analysis of traffic data helps ITS to be more helpful. Mobile phones are the prime source of traffic data. The vast availability of data and increased processing speed of mobile phones is making ITS more robust. Presently for traffic prediction, the entire mobile user's data is accumulated at the central server. The information is then aggregated together to make predictions. In this approach, sensitive user data have the risk of privacy and security—massive user data uploading on the server results in latency. This paper proposes a decentralized approach for vehicular traffic prediction that allows 'selected' local mobiles/organizations (clients) to train the model and share the trained model securely to the server. The selection of organizations to participate in the training process is made by clustering algorithms. The server then aggregates the locally trained model and shares the aggregated model to all the clients again.

Keywords: Decentralized traffic prediction, Federated Learning, privacy-preserving, model averaging

1 Introduction

The cost of land in urban and metro cities is enormous. Rents of a residential apartment are also huge, so people prefer to stay in outskirts areas of the city. This is the reason that the outskirts of urban cities are growing at a tremendous rate. People need to travel to the city for their business purpose daily. Traffic in metropolitan cities is overgrowing with each passing day. A city like Pune, Maharashtra, is an education and industry hub. It is the second-largest city in Maharashtra and ranked ninth-largest town in India. The population in Pune is nearly 3.99 million, and it is one of the fastest-growing cities in Asia. Pune has shown decadal growth of 40% from the last 40 years, and if this trend continues, the population of Pune will hit 5.6 million by 2031. Pune is just an example. We have numerous fast-growing cities in India which have huge people.

The biggest problem faced by the population is the traffic. Travelling through nearly deadlocked traffic, the sound pollution due to vehicle horns is a pathetic condition and is getting worse day by day. Traffic jams also consume a person's time and productivity, which affects the concerned country's economy. The existing road network and public transportation are no longer capable of holding the requirements of a rapidly growing population. According to India Brand Equity Foundation (IBEF), commercial production for vehicles showed growth of 894,551 in 2017-2018 from 567,000 in 2009-2010, which is 5.87%. The Government of India has set a goal of completion of 300 highways by 2020. The estimated cost of this investment is nearly 1.58 trillion. While investing such a considerable amount, a thorough study of traffic patterns has to be done so that this new road infrastructure should accommodate the population's needs for at least 30 to 35 years. Machine

learning is always proven to be useful for such prediction problems. The Intelligent Transportation System (ITS) uses vast data collected by sensors, Road Side Unit (RSU), and surveillance cameras. The data is analyzed using different machine learning algorithms and is used for intelligent transportation decisions. We can use this analysis to ensure adequate public transportation facilities in rush hours, maintaining dynamic traffic signals that adjust signal timing according to traffic. The study of traffic data is also used in advanced traveler planning to avoid travelling in rush hours. Long-term traffic prediction can be useful for road construction planning.

2 Motivation

Machine learning provides good learning algorithms to find the solution for prediction problems. Many machine learning algorithms have proven to be highly successful for problem-solving. For example, the ARIMA [1] proposed in early 1980 is still found helpful in prediction. In years, researchers have modified machine algorithms to handle tremendous data that have increased significantly after the evolution of the Internet. For example, Neural Networks, Deep Learning algorithms give satisfactory results as the amount of data increases. These algorithms got famous because of their capability to provide precise results. Increased computation power is another reason behind the success of these machine learning algorithms. Traditionally machine learning algorithms are given input by collecting all the data from the client. For example, a cancer diagnosis can be made using various Lab tests, diagnostic imaging processing, endoscopic tests, genetic examinations, tumor biopsies. These reports may contain sensitive data, so the clinic or pathology labs are reluctant to share due to the patients' privacy.

Another example is what if the machine learning model needed input from multiple industries and industries are

*E-mail address: savi.lonare@gmail.com

ISSN: 1791-2377 © 2021 School of Science, IHU. All rights reserved.

doi:10.25103/jestr.143.13

unwilling to share their private data. Many times the data provided by industry is limited or of poor quality. Authentication is an additional issue. Increasing competition, privacy, and security of the data are of common worldwide concern. Recently Facebook has faced worldwide protest due to data leakage.

Due to this, Data Protection laws like General Data Protection Regulation (GDPR) [2] are introduced by European countries to protect and secure user's private data. It enforces businesses to make their data policies very clean and clear. User agreements allow users to withdraw their data anytime. Another problem with data collection is the amount of data generated by multiple sources has increased tremendously. So fusing such multiple source data and training the data is difficult. These all restrictions make the realization of Artificial Intelligence more difficult. Precisely the traditional method for data processing and analysis, the information is collected from the source and transferred to the third party where the data analysis is done. When data sources are multiple, this third party is responsible for data clean up and fusion of data. After this step finally the cleaned and fused data is used for model building. These final models are often sold as a service. This traditional method faces different challenges due to data protection laws discussed earlier. Usually, users are in a dilemma whether they will continue to use the model or not, so they are reluctant to share their data. Sometimes, company policies hinder transferring the data to third parties even if they need an AI model for business development. So AI practitioners are trying hard to find a solution so that the privacy and security of the user's data are not violated.

Federated Learning (FL) is one for this challenge. There is very little work that has been done using FL so far. This concept is explained in [3, 4].

3 Related Work

Traffic flow forecasting needs no introduction as it has been a crucial part of the intelligent transportation system. Many researchers have contributed to providing more sophisticated ways to handle the traffic and make travelers plan their travel. In the last few decades, various traffic flow forecasting models have been introduced and implemented successfully to facilitate ITS. Those models were efficient and used for guiding travelers for the best route, traffic conditions, etc. In Literature [5], the authors proposed the SVM-based model SRHTCP (SVM-based real-time highway traffic congestion prediction), which tackles big data collected from various sensors. For the prediction of traffic, the authors have used fuzzy logic. The paper authors claimed that their model improves the prediction by 25.6% accuracy over the traditional method, i.e., weighted exponential moving average. The article [6] proposed an LSTM- NN model for traffic speed prediction using nontraffic parameters, i.e., weather information. In paper [7] authors proposed two models for the prediction of short-term and long-term traffic. Another work [8] discussed traffic prediction in case of congested traffic patterns. The paper [9] authors presented a model for short-term traffic prediction for motorways in the UK. In this paper, they have used an artificial neural network (ANN). A hybrid approach for traffic prediction is discussed in the article [10]. In this paper, researchers presented a model that integrates Singular Spectrum Analysis (SSA) with Artificial Neural Networks (ANN) to predict traffic volume. In recent work [11], authors tried to predict traffic flow from

the data mined from Twitter. In [12], the authors proposed a hybrid convolutional LSTM for critical road sections. In another work [13], the authors used deep autoencoder neural networks for traffic congestion forecasting. In this work, the authors used deep learning on traffic images. Colossal research has been done on making ITS more powerful. Researchers are working hard to make ITS more robust. But most of those techniques were used for centralized training for machine learning. The data is collected from the source/sensor / participants (mobile device) before training in this method. In this kind of approach, the information is collected either from RSU's (Road Side Unit) installed on route or from individual mobile phones, and then the model is trained centrally. The primary concern in this kind of learning is the security and privacy of the data.

So a novel approach called 'FL is proposed by Google in paper [14]. Google is currently testing the FL approach in project 'Gboard' [15]. In this paper, researchers have proposed a model for a mobile keyboard to predict the next word. They used LSTM (Long Short Term Memory) recurrent neural networks for client-side prediction. To find the average of the client updates Federated Averaging algorithm is proposed in [4]. In [16], researchers presented the FL algorithm for image classification. This paper proposed a decentralized learning framework that self balances itself against the bias introduced during training. In another work [17], the authors proposed an FL algorithm for anomaly detection, which works by training the models on the device. The papers [18-20] also discussed FL for optimization, strategies, and model averaging in detail. According to our best knowledge, most of the FL approach is used for image classification and anomaly detection. Very little work has been done on FL for Time-series data. One of the significant contributions for traffic prediction using FL is [21]. This paper's authors proposed the 'FedGRU' algorithm for privacy-preserving traffic flow prediction (TFP). Authors have worked mainly on secure averaging algorithms which aggregate the weights learned on client sites. The authors have used the K means algorithm for randomly selecting the participant. This is one of the pioneering works done on TFP using federated deep learning techniques.

4 Definition of Traffic flow and Decentralized Traffic Prediction

In simple words, traffic flow refers to the number of vehicles moving in a direction at a given time on a particular road. This work predicts traffic flow for a specific location for a given time. The proposed model is based on a local machine learning algorithm for local improvement of the model at the client-side. This improved model is then shared with a server present at the cloud, which later finds out an average of all models received from individual clients and again informed the client. The server restricts the number of clients that would participate in averaging the model.

Federated learning is designed to train a model using a decentralized approach to computation. In this approach, mobile devices themselves are used for training instead of transferring their private data to the server. In FL, mobile devices are responsible for training the model using their local data and communicating the model updates to the server. The updated weights from large numbers of participants (mobile devices) are aggregated and combined to form a better global model, which later communicated back to the participants, as

shown in Figure 1. FL is categorized into two categories based on the data characteristics distribution.

Let the data owner hold the data Matrix D^m , in which a row and the traffic feature represent the traffic sample is represented by a column. Many times these data sets may contain label data too. For example, for banking sectors, user's credit and debit are the labels. So, we represent the complete training dataset as $[I, X, Y]$, where X is features space, Y is label space, and I is sample ID space. So the FL is classified into vertical FL, horizontal FL, and federated transfer learning. The proposed work is using flat FL for the traffic prediction model.

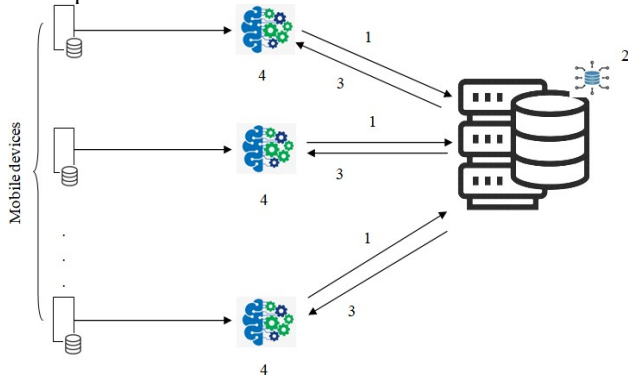


Fig. 1. Architecture for a horizontal federated learning system

4.1 Training steps for Horizontal FL

A classic architecture for a horizontal FL system used for traffic prediction is shown in Fig.1. In this system, k participants (mobile phones/ Sensors) with the same data structure collaboratively learn a model with the server's local data and initial model. We assumed that the mobile phone users are honest and the server is interested as well as simple. It is believed that no leakage of information to the server from any participants is allowed [22]. The training process of this system consists of the following four steps:

Step1: Each participant locally analyzes training gradients, encrypts computed gradients with encryption techniques, and sends encrypted results to the server.

Step2: Server then performs secure aggregation of the received encrypted gradients. Here server does not know information about any of the participants.

Step3: The server reverts the aggregated results to each participant.

Step4: Participants then update their model with the decrypted gradients.

These four steps are iterated until minimum change or no difference is observed in gradients.

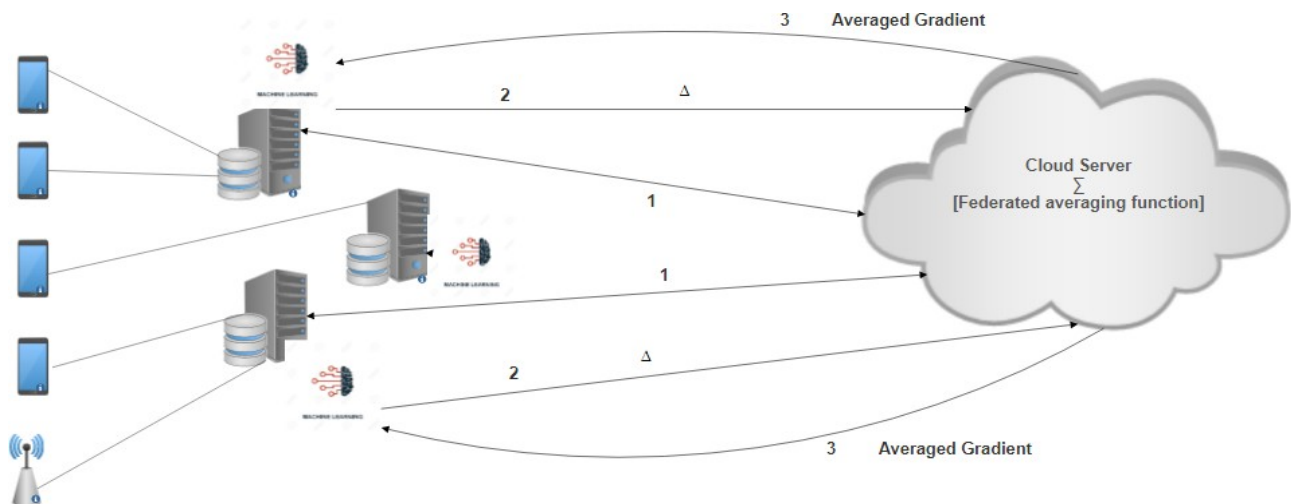


Fig. 2. Architecture of decentralized Vehicular Traffic Prediction

5 The architecture of the proposed Decentralized Vehicular Traffic Prediction

Figure 2 shows an architecture for Decentralized Vehicular traffic prediction. Every sensor updates its entry in the registered Trusted Third Party (TTP). We call it a client. This work has introduced TTP because the vehicles are constantly moving in vehicular networks, so connectivity loss during the communication could be a big issue. The clients may be private firms like cab providers, RSU sensors. Every TTP has m_j devices registered with it, and each of these clients C_j maintains respective database D_j . This work aims to calculate a local traffic prediction model for each client C_j using the D_j and share it with the server on the cloud in a confidential manner. For encryption, this work will use homomorphic encryption [22]. The server available on the cloud completes the handshake process with the selected TPA. Again to restrict the number of clients to participate in model training proposed method used a clustering algorithm that makes the selection

of the client based on spatial and temporal dependency. The server also shares the initial model with the selected TPAs. TPAs use local data and update the model transmitted by the server and upload it back to the server. The server available on the cloud then uses an averaging algorithm [20] to average all the models obtained from the clients and resend them to clients again for the next iteration. We will use Spatio-temporal data for better performance.

6 Methodology

For secure decentralized training, we are using two algorithms. Federated averaging is responsible for securely averaging models received from the participants. For this decentralized traffic flow prediction, we need an aggregation function having the following properties: -

- It should be able to handle high-dimensional data vectors.
- It should provide security to data even on unauthenticated networks.
- It should be robust enough to handle failure in communication.
- It should be communication efficient for managing new sets of participants.

To address this issue, B. McMahan et al. [23] have given a protocol for securely computing sum client input vectors in a constant number of iterations, minimum communication overhead, more robust to communication failure, and secure. The protocol has a low overhead. This algorithm is based on cryptographic techniques in which individual phones update cannot be inspected earlier to the averaging process.

6.1 Fed - Averaging algorithm

The Fed-Averaging algorithm [4] is used on the server-side for averaging updates received from participants and then finding a new global update for all the participants. At training iteration i , a global model weight W_i is sent to subset K out of C clients. In the initial case when $i=0$, i.e., participants start training the model from a universal model that has either been arbitrarily initialized or already trained on proxy data. Each of the participants in i^{th} iteration has local data. This local data varies from participant to participant. With this local data, each participant finds the average gradient G_k with current model W_i . For this step, participants use the Stochastic Gradient Descent Algorithm (SGD). With a learning rate α of a participant, the local update W_{i+1}^k is given as follows:

$$W_i - \alpha \cdot G_k \rightarrow W_{i+1}^k \quad (1)$$

The server then aggregates this client-side weight update to compute a new global model, which is given as follows:

$$\sum_{k=1}^K \left(\frac{i_k}{M} W_{i+1}^k \right) \rightarrow W_{i+1} \quad (2)$$

Here, $M = \sum_k i_k$, summation of local updates calculated by participants.

This is how the participants use the SGD algorithm to compute the local updates. These updates are then sent to the server and aggregated to find a global model. We need to tune the model to improve the performance by working on the number of participants, epochs, and participant's batch size. This decentralized approach gives lesser security and privacy violation risk than the traditional approach. This approach also gives users more control over their sensitive data. The most challenging task for training the model is non-IID (not independent and identically distributed) data. The data used for the local training by each participant highly depend on the application usage pattern of that participant. This data may be unbalanced and statistically skewed. Heavily used applications generate more data than less to average used applications. Data generated by the population will not have a similar distribution. Again each participant will use a varying amount of data for learning the model, which will affect the accuracy of results effectively. So consideration of non-IID data is essential for decentralized learning. McMahan et al. [23] have verified that the FedAvg function for secure aggregation works with specific non-IID data, but it is still challenging to prove it works for time series data. The local updates shared by the client to the server are short-lived and not saved on the

server. The new global model is processed, communicated to clients, and then destroyed immediately. An improved global model is computed using updates received from participant devices.

6.2 Federated Machine Learning using LSTM

For on-device training, this work regards Long Short Term Memory (LSTM). LSTM is a particular type of recurrent neural network (RNN). It was designed to learn patterns from input data sequences. This algorithm has already been proven for the traffic prediction model [25]. LSTM uses the concept of gates. It decides which information to forget using 'forget gate,' new data is saved using 'input gate,' and 'output gate' decides which information to be passed to the next layer. LSTM RNN helps to remember the error during the learning process, and later those errors can be backpropagated through time and layers.

6.3 Spatiotemporal Clustering of clients

The proposed system uses a clustering algorithm on the client's data. This work uses a decentralized K-means clustering algorithm [24] to select participants for local training. The system then uses the centroid of the cluster for calculating errors. This selective model training will reduce the communication as selected numbers of models are shared with the server.

7 Discussion

This section will discuss the existing system and the advantages and limitations of the proposed approach. Although lots of work has been done in ITS, there are many open issues like privacy and security. Traditional methods have not handled the matter of privacy of user's sensitive data in the ITS system. This novel approach uses the concept of FL, which allows for data collection instead of sharing the data with the server. Selection of the participants by clustering on spatial information will restrict the number of participants in the process and thus minimize communication overhead. This approach will also help to improve results as location is an essential parameter for traffic prediction.

The handshake process for the selection of the client is an overhead for the system. Again synchronization during the model update is very vital. As vehicles are constantly moving, link failure can be a significant issue. For handling, this issue is proposed to store the data in the TTP database. This will minimize the risk of communication link failure.

8. Conclusion

This paper proposed a decentralized approach for the traffic prediction model. This model uses the concept of FL, which protects the privacy of user-sensitive data. This approach provides security to the user data as it is not directly shared with the server. The shared models are also deleted from the server after specific iterations. The clustering algorithm was used to restrict the number of participants in training the models. As the clustering uses the location parameter (latitude and longitude), only spatially related clients are selected for the training. This prediction model reduces communication overhead and data uploading as the client shares the trained weights only.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License.



References

1. Levin M., and Tsao Y.(1980), 'Forecasting freeway occupancies and volumes', in Transportation Research Record, Volume 773, pp.47-49.
2. EU. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND THE COUNCIL on General Data Protection Regulation. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT>, accessed June 2020.
3. McMahan, H., Moore, E., Ramage, D., Hampson, S., & Arcas, BA (2017). 'Communication-Efficient Learning of Deep Networks from Decentralized Data' in *AISTATS*.
4. Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth(2017), 'Practical Secure Aggregation for Privacy-Preserving Machine Learning,' in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '17),p 1175-1191.
5. F.H. Tseng, J.H. Hsueh, C.W. Tseng, Y.T. Yang, H.C. Chao, and L.D.Chou(2018), 'Congestion prediction with big data for real-time highway traffic,' in *IEEE Access*, vol. 6, 201; p. 57311-57323.
6. Nicholas G. Polson, Vadim Sokolov(2017), 'Deep Learning for Short-Term Traffic Flow Prediction in Transportation Research Part C Emerging Technologies, pp. 1-17 DOI:10.1016/j.trc.2017.02.024.
7. Z. Liang, Y. Wakahara(2013), 'City traffic prediction based on real-time traffic information for Intelligent Transport Systems, in 13th International Conference on ITS Telecommunications (ITST), Tampere, pp. 378-383.
8. Rehborn H., Klenov S.L.(2009), 'Traffic Prediction of Congested Patterns,' in *Encyclopedia of Complexity and Systems Science*' Springer, DOI:10.1007/978-0-387-30440-3_564.
9. Goves C., North R., Johnston R., Fletcher G(2016), 'Short term traffic prediction on the UK motorway network using neural networks, in *Transp. Res. Procedia* 13,pp 184-195 DOI:10.1016/j.trpro.2016.05.019.
10. Stylianos Kolidakis, George Botzoris, Vassilios Profillidis, Panagiotis Lemonakis(2019), 'Road traffic forecasting — A hybrid approach combining Artificial Neural Network with Singular Spectrum Analysis, Economic Analysis and Policy,' in *Springer*, Volume 64, pp 159-171 DOI: 10.1016/j.eap.2019.08.002.
11. Essien A., Petrounias I., Sampaio P, Sandra Sampaio(2020), 'A deep learning model for urban traffic flow prediction with traffic events mined from Twitter,' in *World Wide Web*, springer, pp.1-24 DOI:10.1007/s11280-020-00800-3,
12. Yang Gang, Wang Yunpeng, Yu Haiyang, Ren Yilong, Xie Jindong(2018), 'Short-Term Traffic State Prediction Based on the Spatiotemporal Features of Critical Road Sections,' in *Sensors* 18(7):2287, vol- 18;p.2287-2315.
13. Zhang, S., Yao, Y., Hu, J., Zhao, Y., Li, S., & Hu, J. (2019). Deep Autoencoder Neural Networks for Short-Term Traffic Congestion Prediction of Transportation Networks. *Sensors (Basel, Switzerland)*, 19(10), 2229. <https://doi.org/10.3390/s19102229>
14. Hard, Andrew & Rao, Kanishka & Mathews, Rajiv & Beaufays, Françoise & Augenstein, Sean & Eichner, Hubert & Kiddon, Chloé & Ramage, Daniel. (2018) 'Federated Learning for Mobile Keyboard Prediction' arXiv:1811.03604.
15. Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, Françoise Beaufay. (2018), 'Applied federated learning: Improving google keyboard query suggestion' arXiv preprint arXiv:1812.02903.
16. Duan, Moming & Liu, Duo & Chen, Xianzhang & Tan, Yujuan & Ren, Jinting & Qiao, Lei. (2019). 'Astraea: Self-Balancing Federated Learning for Improving Classification Accuracy of Mobile Deep Learning Applications'.pp 246-254. 10.1109/ICCD46524.2019.00038.
17. Rei Ito, Mineto Tsukada, Hiroki Matsutani. (2020), 'An on-device federated learning approach for cooperative anomaly detection,' arXiv:2002.12301.
18. Jakub Konecny, H. Brendan McMahan, Daniel Ramage, Peter Richtarik(2016), 'Federated Optimization: Distributed Machine Learning for On-Device Intelligence'; arXiv:1610.02527.
19. Jakub Konecny, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, Dave Bacon(2017), 'Federated Learning: Strategies for Improving Communication Efficiency,';arXiv:1610.05492.
20. H. Brendan McMahan, Eider Moore, Daniel Ramage, Blaise Aguera y Arcas. (2016), 'Federated Learning of Deep Networks using Model Averaging,' arXiv:1602.05629.
21. Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato and S. Zhang. (2020), 'Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach,' in *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751-7763, DOI: 10.1109/JIOT.2020.2991401.
22. Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai.(2018), 'Privacy-Preserving Deep Learning via Additively Homomorphic Encryption,' in *IEEE Trans. Information Forensics and Security*, vol. 13, no. 5,p. 1333-1345.
23. McMahan H. B., Ramage D., Talwar K., Zhang L.(2018), 'Learning differentially private recurrent language models,' in *International Conference on Learning Representations(ICLR)*, arXiv:1710.06963.
24. Soliman A., Girdzijauskas S., Bouguelia MR., Pashami S., Nowaczyk S(2020), 'Decentralized and Adaptive K-Means Clustering for Non-IID Data Using HyperLogLog Counters,' in *Lecture Notes in Computer Science*, vol 12084. Springer, *Advances in Knowledge Discovery and Data Mining*, pp.343- 355.
25. Yang Bailin, Sun Shulin, Li Jianyuan, Lin Xianxuan, Tian Yan(2018), *Traffic Flow Prediction Using LSTM with Feature Enhancement*, *Neurocomputing*, Vol- 332., p.320-327.