Urban Traffic Forecasting using Federated and Continual Learning

Chiara Lanza, Eduard Angelats, Marco Miozzo, Paolo Dini
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)
{chiara.lanza, eduard.angelats, marco.miozzo, paolo.dini}@cttc.es

***Abstract*—Smart cities are instrumented with several types of sensors, which allow to transmit, elaborate and exploit the collected data for different services. In this paper we focus on the urban traffic forecasting application. In such context, centralized learning (i.e., training a model in a central unit with data sent from the sensors) or having one model per sensor are the state-of-the art solutions. However, the transmission of such big amount of data, as those from a massive deployment of traffic intensity sensors, implies dense network architectures, long transmission delay, higher network congestion probability and significant energy consumption. On the other hand, training a model only with local data from each sensor lacks in generalization. In this paper we advocate Edge Intelligence and propose a federated peer-to-peer Continual Learning strategy, which applies two variants of Continual Learning principles on data from traffic intensity sensors deployed in a city with the aim to create collaboratively a single general model for all. The analysis of results, performed with real data from a district in Madrid, demonstrates that urban traffic forecasting can be successfully performed in a peer-to-peer fashion. Moreover, we prove that the proposed approaches have lower energy footprint (up to 87% less) and comparable accuracy with respect to state-of-the-art benchmarks.**

***Index Terms*—Machine Learning, Edge computing, Continual Learning, Urban traffic forecasting, Smart Cities, Sustainability.**

I. Introduction

Nowadays, almost six-in-ten people on Earth live in cities and predictions estimate that this percentage will grow till more than 70% by 2050 [1], [2]. To ensure that the benefits of urbanization are fully shared and inclusive, sustainable development of metropolitan areas is needed. In this context, urban traffic management is a key element to enable sustainability through planning and congestion control, which aim to decrease travel times, prevent accidents and safeguard environmental and noise pollution.

Our cities are becoming smart thanks to the deployment of sensors, Intelligent Transportation Systems (ITS) [3] and Internet of Things (IoT). Data captured by such instrumented environment are processed to provide different services, such as urban traffic forecasting. The use of Machine Learning (ML), and more in particular of Deep Learning (DL), for this aim relies on the availability of large computational resources. In fact, the conventional information processing strategy is to transmit the data from the IoT devices to cloud data centers. In such scenario, communication and computational bottlenecks may appear due to a high amount of connected clients to the same central server. For example, the amount of data shared in Barcelona in 2016 by the traffic sensors is around 8 GB per day [4]. The transmission of such big amount of data implies dense network deployments, long transmission delay, higher network congestion probability and significant energy consumption [5]. In fact, standard cloud data centres used for storage and running ML algorithms require a considerable amount of electricity, which corresponds to the 2% of all global CO_2 emissions and it is expected to increase up to 8% by 2030 [6]. Moreover, in the event of a server failure at the data centre, the training process can be disrupted or delayed. Similarly, a connection problem between the server and one of the deployed device implies the isolation of the sensor and its data. For these reasons, both research community and industry have started considering the Edge Computing (EC) paradigm, which consists on pushing the computation resources close to the edge of the network [7]. Furthermore, EC may be integrated with AI and create the so-called Edge Intelligence (EI) [8]. In this scenario, ML algorithms run directly on the edge device and build a sort of distributed data centre, in which learning tasks are executed closer to the data sources. Such approach provides the following benefits with respect to classic centralized AI approach [5]:

- lower latency: in the centralized solution, the data centre is usually located far from the data sources;
- privacy protection: sensitive data are maintained at the edge;
- lower communication overhead: no need to send big amount of data to train ML and DL models;
- smaller memory footprint: the necessary memory is distributed across the edge devices, instead of being concentrated in big energy-hungry central units;
- smaller energy requirements: reducing communication costs and using energy-efficient edge devices.

One of the most popular EI solutions is Federated Learning (FL), which is a distributed ML approach enabling model training on a large domain of distributed data sources using only local model updates and no data sharing. [9]. The global model is built collaboratively through a central server, which is in charge of collecting and merging the local model updates [10]. However, such solution still suffers the single point of failure problem [11]. In [12], an all-to-all scheme where each worker sends the local model updates to all the other workers is proposed to overcome this issue. Alternatively, in [13] a gossip based synchronization is adopted to exchange

the models among the data source nodes with peer-to-peer (p2p) transmissions and reduce the communication overhead of the previous mentioned proposal. The tasks of the central unit are assumed collaboratively by each node, which perform local learning plus merging with the received model.

More recently, FL has been extended with Continual Learning (CL) [14] (also known as Incremental Learning or Lifelong Learning). CL is a set of ML algorithms designed to learn a model for a large number of tasks. Starting from the human cognition capable to learn concepts sequentially, CL aims to increase the adaptation capabilities of the models in dynamic environments. Therefore, task incremental CL [15] represents a valuable tool for sequential learning in distributed data settings, featuring lower communication and computational costs, which, in turn, decrease the energy consumption of the training process, while maintaining high model accuracy.

In this paper, we investigate on the usage of task incremental CL for urban traffic forecasting. In particular, we propose a Federated peer-to-peer Continual Learning strategy (FpC), which applies CL on data from traffic intensity sensors deployed in a city with the aim to create collaboratively a single general model for all. The global model is trained incrementally across the different data sources participating in the training phase. Moreover, we extend the proposed FpC solution with an early stopping condition in the local training (named FpC with early stopping, FpCes) to further decrease the computational energy. To the best of our knowledge, this is the first time that edge intelligence and Continual Learning are applied to the urban traffic flow prediction problem.

We compare our proposals with two benchmark solutions: i) a model based on classic centralized AI architecture, which processes all the data at the central unit simultaneously and ii) a set of single-sensor prediction models, based only on local data. The study is performed on an open dataset containing the vehicular intensity collected by sensors deployed in Madrid [16] and a Long Short Term Memory (LSTM) [17] model. The analysis of results presented here demonstrate that urban traffic forecasting can be successfully performed in a peer-to-peer fashion. Moreover, we prove that the proposed approaches based on FpC have lower energy footprint and comparable accuracy with respect to the benchmark solutions.

As a result, the contributions of the paper are summarized in the following list:

- Design of an urban traffic prediction framework based on edge-intelligence with reduced energy consumption and high accuracy;
- Design of a distributed ML model based on FL and CL to solve the traffic flow intensity prediction problem;
- Performance evaluation of the proposed distributed solutions both in terms of accuracy and consumed energy in a district of Madrid. Moreover, we include also a comparison with two benchmarks (centralized and one-model-per-sensor approaches).

The paper is organized as follows. Section II reviews the state-of-the-art on urban traffic forecasting techniques and architectures, including also FL and CL approaches. In Section III, we present the used dataset and its exploratory data analysis. The details of the two FpC proposals together with the two benchmarks are presented in Section IV. In this section the adopted energy model is also included. Section V presents the achieved results. Finally, Section VI concludes the paper and introduces possible future research directions.

II. Background

The problem of urban traffic forecasting has been studied through many different approaches [18]. Kalman filters and the Auto-Regressive Integrated Moving Average (ARIMA) have been widely adopted. They require large datasets, and reach lower accuracy with respect to ML and DL methods [19]. Moreover, they present difficulties in representing spatial relationships in traffic flow forecasting tasks [18]. To overcome these issues and to manage the non-linear and stochastic behavior of the traffic, ML techniques, such as Support Vector Machines (SVM) [20] and K-nearest neighbors (KNN) [21] have been applied. Recently, DL based solutions have been also used to exploit their capability to catch space-time dependencies in heterogeneous datasets from multiple sources, as in the case of vehicular traffic [22]. DL has been usually applied to datasets containing information from the whole city for obtaining a single model [20], or using data from a small dedicated area (e.g. a road or one sensor) [21], with the drawback of having a model that is not general and needs to be replicated in many instances.

In this work we propose a novel approach, which exploits edge computing and combine the distributed nature of data sources (i.e., traffic intensity sensors) with the sequential characteristics of CL to train a global model in a peer-to-peer fashion. These characteristics are essentially missing in our literature search for urban traffic forecasting.

As for the usage of CL for collaborative training in distributed settings, an extension to classic federated scenario has been presented in [23]. They propose two different solutions based on knowledge distillation, which involve the central server as teacher model. On the other hand, Huang et al. [24] implement a single visit Continual Learning (SVCL) for p2p Federated Learning for metastasis identification, and they compare the results with a standard Federated solution. In [25], the authors maximize the knowledge transfer between clients while minimizing the inter-client interference and communication costs. In particular, they tackle this problem by decomposing the model parameters and using selective transferring techniques: each edge device will selectively update the parameters at each step. A similar approach has been used in [26].

In this work we are interested in evaluating server-less FL solution based on the peer-to-peer paradigm, similarly to [24]. In addition, we evaluate the early stopping condition to reduce the computational energy of the local training. Moreover, we focus our evaluation on both model accuracy and energy consumption, following the Green AI principles [27]. To the best of our knowledge, the evaluation of the energy vs. accuracy trade-off for Federated p2p Continual Learning have

not been discussed in the literature and have not been applied to the urban traffic forecasting problem.

III. Dataset

Urban traffic information can be derived using Traffic Data from Infrastructures, Trajectory Data from Vehicle, Automatic Fare Collection (AFC) Records from Transit Systems or other non-categorized sources [28]. We use the vehicle flow intensity information to characterize it. In particular, we consider the traffic data from infrastructure, which in our case is defined as the number of the vehicles passing through the portion of the road where the vehicle detection sensors are placed.

Numerous datasets on urban traffic intensity are available on-line, such as open data distributed by the city municipalities (Madrid [16], Barcelona [29], Gdansk [30], Turin [31]), or others used for previous works in this field (IARAI [32], UTD19 [33]). We select Madrid open data based on our spatial and temporal coverage requirements. We need a high spatial density to cover pervasively the area under study; and we want two years of data to train over one year and test the results over the other. Moreover, the period should not consider 2020 to avoid any influence of the mobility restrictions due to COVID-19 pandemic. Small time data granularity (5 or 15 minutes) is also important, since our aim is short term traffic forecasting. Traffic data in Madrid have been collected using 7.360 vehicular detectors, 5.886 of them are electromagnetic sensors based on the floor spread in more then 4000 measurement sites, detecting cars with a time granularity of 15 minutes.

Madrid is divided in 21 districts [16], as shown in Fig. 1. In this paper, we consider data coming from sensors in the Salamanca district, highlighted in Fig. 1 (blue). The data used are enough to train and test our Continual Learning proposal in a wide area and compare it with our benchmarks. Our results should be considered as a preliminary study for using edge intelligence in smart cities scenarios and possible extensions can be easily adopted to cover the whole city, as discussed in Section VI. Salamanca district is a perfect match for our purposes, since no big changes have been made by the municipality in this zone in term of mobility between 2017 and 2019. We note here, instead, that we did not consider the central district due to the implementation of *Madrid Central* project [16] at the end of 2018. With this policy, the municipality defined a low-emission zone with vehicular mobility restrictions affecting traffic flow significantly in 2019.

A. Exploratory Data Analysis

We consider data from 2017 as training set, and 2019 as test set. The data from 2018 has not been used due the many missing data, which cover even complete months.

Sensors can be classified in different levels based on their intensity range. In this work, we exploit the classification proposed in [34], where 4 different flow levels are defined i.e., low, medium, high and very-high. Salamanca district data contains only three classes, since sensors with very-high intensity flow are not present, as depicted in Fig. 2. As done in

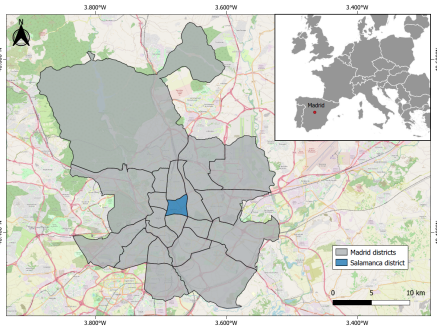


Fig. 1: Location of Madrid and its districts. Salamanca district (area under study) is highlighted

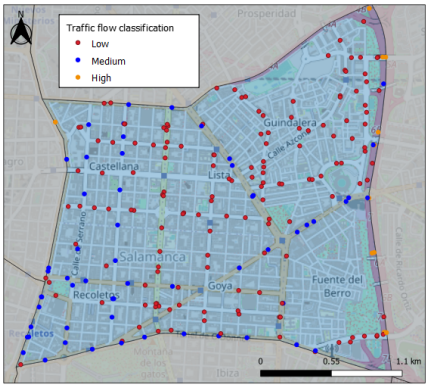


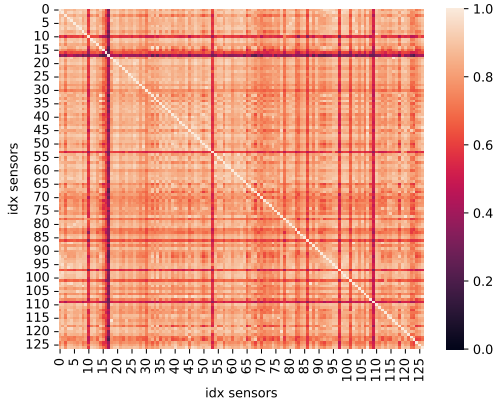
Fig. 2: Salamanca district and traffic flow sensors. Sensors marked in yellow, blue and red are belonging to high, medium and low intensity class, respectively.

the previous literature for urban traffic forecasting (e.g., [35], [36] and [37]), we focus our study on a limited number of sensors, i.e., low level class, which is, however, the most dense deployed and allow a more pervasive study of the district. After discarding sensors with more than 20% of missing data, the total amount of considered sensors is 153 (2017) and 151 (2019). Among them, we select the sensors that have data for both 2017 and 2019 as well as sharing the same sensor ID. The resulting set is made of 127 eligible sensors, that are considered for the prediction task of this work.

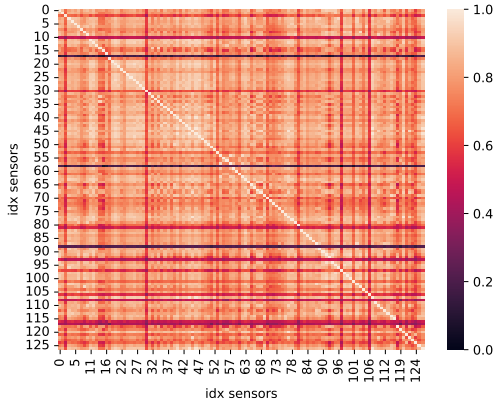
In Fig. 3a and Fig. 3b, we show the correlation matrices of the time series data collected in 2017 and 2019, respectively. The x-axis and the y-axis indicate the index of the sensors, assigned with an incremental order for the sake of readability. Data correlation is normally high for both 2017 and 2019 data; sensors that show lower correlation in 2017 are mostly the same as in 2019, which implies that data distribution in the two considered years is similar. Moreover, in Fig. 3c, we present the correlation matrix between data from the same sensor in 2017 and 2019. Here, values are lower, i.e., there exists a small drift in time from 2017 to 2019.

IV. System Models

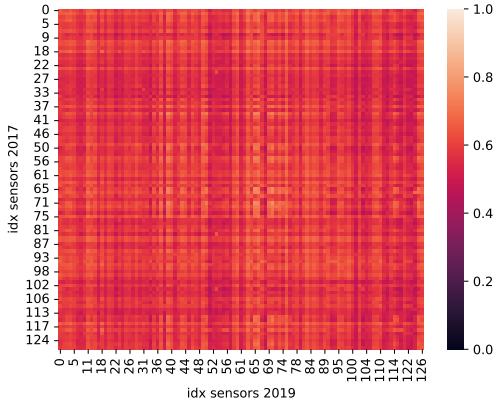
Our distributed learning environment is composed of a set of traffic sensors deployed in the studied area: $S =$



(a) 2017



(b) 2019



(c) 2017 vs 2019

Fig. 3: Correlation matrices of the studied sensors

$\{s_1, s_2, \cdots, s_n\}$ where s_i is the sensor with index i and n is the total number of sensors. We call d_i the data collected by the sensor s_i. In particular for this work, in the training phase, d_i will be the data collected during the 2017. Therefore, the whole dataset is defined as $D = \{d_1, d_2, \cdots, d_n\}$. We define $\boldsymbol{x}(t)$ as the vector containing the traffic flow intensity measurements. We perform *data normalization* in the interval $[-1, 1]$ to scale data. We apply *data windowing* to the sequence $\boldsymbol{x}(t)$, to create the input of our prediction models. At every prediction step h, the input sequence $\boldsymbol{x}(h)$ can be expressed as

$$\mathbf{x}(h) = [\boldsymbol{x}(h), \boldsymbol{x}(h+1), \ldots, \boldsymbol{x}(h+p)]$$

being p the amplitude of the observation window.

A. Federated peer-to-peer Continual Learning

In the proposed FpC approach, model parameters are passed sequentially from one sensor to the next, till exploring the whole set and following a specific path, as illustrated in Fig. 4. We are assuming here a (logical) mesh topology, meaning that every sensor can reach all the others.

Alg.1 describes the proposed FpC strategy. After initializing the model weights w_0, we choose one path for the Continual training through the function $generateRandomSequence$. It returns a sequence of sensors to be visited for training among all the possible combination over the total $n!$. Then, we go through all the sensors in the selected path and execute the function $train(w, d_k, E)$, which updates the weights of the model w according to the training performed on the local dataset d_k during E epochs.

We also propose a variant of FpC, called FpCes, to further reduce the computational energy of the training process. This approach includes an *early stopping* condition in the training function, as defined in Alg.2. By doing so, at each epoch it is evaluated whether the loss has not decreased in the last iteration and, in case this condition is verified for *max_wait* consecutive epochs, the training is stopped. The *early stopping* condition is motivated by the idea that the data collected by the sensors present similar distributions (as explained in Sec. III). Thus, when the sensors detect that local data are not significant for the training phase, the local process is stopped and the model passed to the next node in the sequence. In such a way, computational energy is reduced, since the sensors can perform less training epochs with respect to the scheduled E. It is to be noted that, the resulting number of epoch per sensors is evaluated automatically during the FpCes algorithm execution and it does not need to be known a-priori.

B. Benchmarks

The proposed models are compared with two solutions that are based on architectures relying on opposite paradigms: a classic centralized approach and a method training one model per sensor.

In the centralized approach (Fig. 5), the data collected by all the sensors are sent to a central unit (e.g., a cloud data centre) to training a global model. Alg.3 describes the centralized

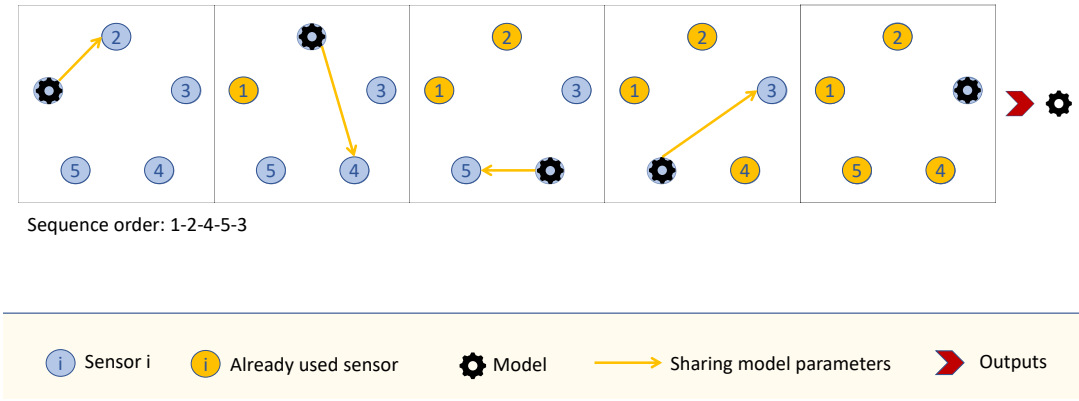


Fig. 4: Diagram of FpC and FpCes models

Algorithm 1 FpC Algorithm

Initialize w_0
$path = generateRandomSequence()$

for k in $\{1, 2, \cdots, n\}$ **do**
  $d_k = path[k]$
  $w \leftarrow train(w, d_k, E)$
end for

return w

Algorithm 2 Train function for FpCes

function TRAIN(w,d_k, E)
  wait←0

  for e in $\{1, 2, \cdots, E\}$ **do**
    $w_e \leftarrow$ update(w_{e-1}, d_k)
    validLoss$_e \leftarrow$ validation(w_e, d_k)

    if validLoss$_e$ >=validLoss$_{e-1}$ **then**
      wait = wait+1

      if wait==max_wait **then**
        break
      end if
    else
      wait=0
    end if
  end for
return w

training process. Note that the input of the $train(w, D, E)$ function is the dataset $D = \{d_1, d_2, \cdots, d_n\}$.

In one-model-per-sensor approach, instead, each model is trained using only the local data of that specific sensor $S = \{s_1, s_2, \cdots, s_n\}$ (Fig. 6). Alg.4 describes the procedure followed. Note that in this case, a set of S models is generated since there is one model w^i for each sensor s_i.

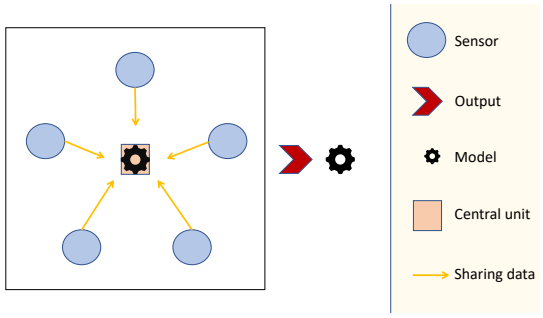


Fig. 5: Diagram of the centralized approach

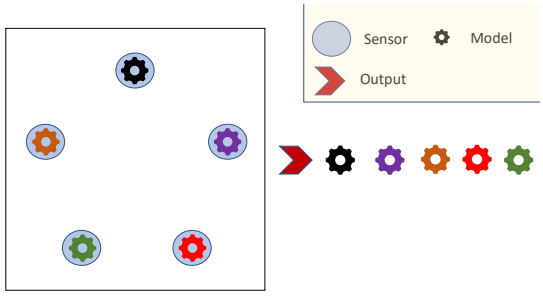


Fig. 6: Diagram of one-model-per-sensor approach

Algorithm 3 Centralized Algorithm

Initialize w_0
$w \leftarrow train(w, D, E)$
return w

Algorithm 4 One-model-per-sensor Algorithm

Initialize a w_0^i for each sensor s_i

for i in $\{1, 2, \cdots, n\}$ **do**
  $w^i \leftarrow train(w^i, d_i, E)$
  $W[i] = w^i$
end for
return $W = \{w^1, w^2, \cdots, w^n\}$

C. Energy Models

The energy consumed during the training process is given by two main components: the computations for updating the model with the information from the data (local or global depending on the approach) and the communication to send the updated model to the next sensor in the sequence (FpC and FpCes) or to send sensor data to the central unit (centralized approach). The computational energy has been computed by Carbontracker [38], a tool developed in Python for tracking and predicting the energy consumption of DL models. It accounts for the energy consumed by the memory and the processing units.

Regarding the communication energy, we assume that the connections between sensors and the connections between sensor and the central unit are IEEE 802.11ax wireless links. The amount of data to transmit differs with the learning solution. In FpC and FpCes, each sensor sends the model parameters only; in the centralized approach, the sensors send the entire local dataset. No transmission is performed in the one-model-per-sensor approach. We calculate the time spent to send data T_{tx} based on [11]. Then, we evaluate the energy consumption of a transmission as $E_{tx} = T_{tx}P_{tx}$, where P_{tx} is the transmission power used by the sensors (assumed to be of 9 dBm).

V. Numerical Results

This section is divided in two parts. Section V-A defines the setup used in the performance evaluation. Section V-B presents the achieved results and a comparison with the two benchmarks.

A. Evaluation setup

We considered a LSTM-based model [17] to predict the traffic flow intensity, trained with Mean Squared Error (MSE) as loss function. We adopt the same LSTM architecture for all the studied solutions to perform a fair comparison among the different learning solutions. In fact, the goal is to study the applicability of Continual Learning to train collaboratively a global model over distributed data, with the main aim to reduce the energy consumption, while at the same time maintaining the highest accuracy. We state here that a more complex DL model might have been designed to increase the accuracy, at the cost of an increase of its computational energy and jeopardizing fairness in comparing the different models.

We select the hyperparameters applying the Python library Optuna [39] on a LSTM model implemented in Pytorch [40]. The final resulting parameters are reported in Tab. 1. We run the experiments on a computer with an AMD Ryzen 7 3700X 8-Core Processor, with a GPU NVIDIA GeForce RTX 2080 Ti.

TABLE 1: Setup Parameters

Hyperparameter	Value
optimizer	Stochastic gradient descent
learning rate	0.088
window used to predict	11
no. of LSTM layer(s)	1
no. of unit for the layer	25
no. of necessary epochs	75
batch size	208

B. Results Analysis

We evaluate the achieved performance considering the following indicators:

- **Mean Absolute Percentage Error** (MAPE): the measure of the prediction accuracy, calculated as: $MAPE = \frac{100\%}{n}\Sigma_{i=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$ where A_t is the actual value ground-truth and F_t is the forecast value.
- **Coefficient of determination** R^2: is a statistical calculation that measures the degree of interrelation and dependence between two variables. It ranges from 0 to 1: the closer is to 1, the better the model is making the prediction.
- **Training time**: is the time spent to train the global model. In the case of one-model-per-sensor approach, we consider the cumulative local training time.
- **Computational energy**: is the energy consumed to train the model.
- **Communication energy**: is the energy consumed for the communication, as explained in Sec. IV-C.
- **Communication overhead**: is the total amount of data sent during the training phase.

In Tab. 2 we report the performance indicators obtained using FpC, FpCes and the two benchmarks, averaged over 40 runs. In particular, each trial of FpC and FpCes is the results of a different sequence of sensors. One-model-per-sensor is the best in terms of both MAPE and R^2, since it is tailored on the specific data collected by that given sensor. Of course, this approach lacks in generalization. FpCes reaches, however, very similar performance in accuracy as one-model-per-sensor, and outperforms both FpC and centralized approach. Moreover, FpCes saves up to 87%, 85% and 85% of training time compared to centralized, FpC and one-model-per-sensor, respectively. Consequently, FpCes presents also the lowest computational energy; instead FpC consumes the same amount of computational energy as the one-model-per-sensor. The centralized approach is the most energy-hungry. Communication overhead is higher for the centralized approach, since it requires the transmission of the entire local datasets from all the sensor at the central unit (380MB). On the other hand, FpC and FpCes have to share only the model parameters, e.g., 2200 B for each sensor, which correspond to a total of 0.279 MB transmitted. This is reflected into the communication energy consumption figures, being that of the centralized approach by far the biggest. It is worth to highlight here that the total energy reduction using FpCes (FpC) is of 87% (18%) compared to the centralized solution.

Fig. 7 shows the average MSE and its variance considering different training paths for both FpCes and FpC. We can appreciate that FpCes has both lower MSE and variance, which implies that this approach is able to better generalize the

TABLE 2: Performance Comparison

	MAPE [%]	R^2	training time [mm:ss]	comp. energy [kWh]	comm. energy [kWh]	comm. overhead [MB]
FpCes	0.77	0.82	07:13	0.011	7.93e-10	0.279
FpC	0.92	0.78	46:43	0.079	7.93e-10	0.279
centralized	0.94	0.78	56:40	0.096	1.13e-6	380
one-model-per-sensor	0.75	0.81	46:53	0.079	0	0

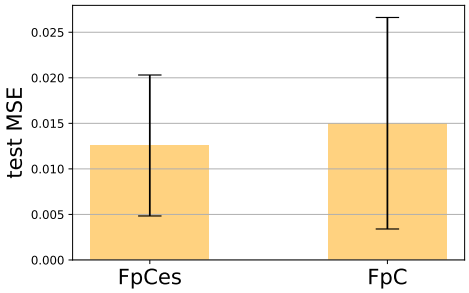


Fig. 7: Average test MSE for FpCes ad FpC.

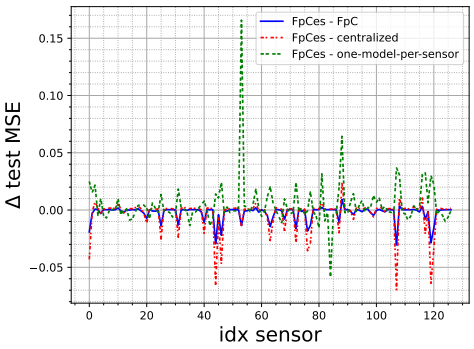


Fig. 8: Test MSE comparisons sensor by sensor.

prediction and is less conditioned by the order of the sensors in the sequence.

Finally, in Fig. 8 we report the difference Δ between the test MSE using FpCes and the other three solutions at each sensor, respectively. Negative values indicate that FpCes architecture has lower MSE. This figure shows the generalization characteristics of the FpCes and FpC proposals. In fact, the final global models trained with the proposed Continual Learning principles have higher accuracy than the centralized approach for all the deployed sensors and very similar performance as the one-model-per-sensor. An outlier is identified in node 53, which has a very different flow behavior with respect to the other 126 and, for this reason, presents higher errors.

VI. Conclusions

In this work we have investigated the application of Edge Intelligence to smart city scenarios. In particular, we have analysed the usage of Continual Learning principles for urban traffic forecasting to overcome the classical problems of centralized approaches, i.e., transmission of big amount of data, long transmission delay, higher network congestion probability and significant energy consumption. We have proposed a Federated peer-to-peer Continual learning strategy (FpC), which applies CL on data from traffic intensity sensors deployed in a city with the aim to create collaboratively a single general model for all. The global model is trained incrementally across the different data sources participating in the training phase. Moreover, we have extended the proposed FpC solution with an early stopping condition in the local training (FpCes) to further decrease the computational energy. The tests performed using real data from a district of Madrid demonstrate that urban traffic forecasting can be successfully performed in a peer-to-peer fashion. In fact, we have proved that the proposed approaches have lower energy footprint (up to 87% less) and comparable accuracy with respect to state-of-the-art benchmarks.

Our work can be considered as a preliminary step towards federated peer-to-peer continual learning for urban traffic forecasting, and opens several issues and possible research directions. First, a deeper analysis is needed to consider realistic sensor network topologies and algorithms for the most appropriate training path selection. Furthermore, other communication technologies can be considered together with IEEE802.11ax, such as LoRa, Sigfox or LTE-NB. Then, other continual learning principles (e.g. a replay strategy [41] or the Elastic Weight Consolidation [42]) can be investigated to achieve higher accuracy and model generalization. Finally, extending the analysis to the whole set of sensors deployed in a city would be also interesting to be explored.

Acknowledgment

This publication has been partially funded by the European Union Horizon 2020 research and innovation programme under Grant Agreement No. 953775 (GREENEDGE) and the Spanish grant PCI2021-122043-2A/AEI/10.13039/501100011033.

References

[1] "Demographia world urban areas 18th annual edition- july 2022 - (built up urban areas or world agglomerations)." Available at: http://www.demographia.com/db-worldua.pdf, (Accessed: 14th December 2022).

[2] "World Urbanization Prospects - Population Division - United Nations." Available at: https://population.un.org/wup/, (Accessed: 14th December 2022).

[3] R. J. Weiland and L. B. Purser, "Intelligent transportation systems," *Transportation in the new millennium*, 2000.

[4] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, E. Marin-Tordera, J. Cirera, G. Grau, and F. Casaus, "Estimating smart city sensors data generation," in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, (Vilanova i la Geltru, Spain), pp. 1-8, IEEE, Jun 2016.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," May 2019. arXiv:1905.10083 [cs].

[6] A. Andrae and T. Edler, "On global electricity usage of communication technology: trends to 2030," *Challenges*, vol. 6, no. 1, p. 117-157, 2015.

[7] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, p. 85714-85728, 2020.

[8] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," Feb 2020. arXiv:1909.00560 [cs].

[9] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Kone, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," Mar 2019. arXiv:1902.01046 [cs, stat].

[10] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," Feb 2017. arXiv:1602.05629 [cs].

[11] E. Guerra, F. Wilhelmi, M. Miozzo, and P. Dini, "How much does it cost to train a machine learning model over distributed data sources?," 2022. arXiv:2209.07124 [cs].

[12] P. Patarasuk and X. Yuan, "Bandwidth optimal all-reduce algorithms for clusters of workstations," *J. Parallel Distributed Comput.*, vol. 69, pp. 117-124, 2009.

[13] J. Daily, A. Vishnu, C. Siegel, T. Warfel, and V. Amatya, "Gossipgrad: Scalable deep learning using gossip communication based asynchronous gradient descent," *CoRR*, vol. abs/1803.05880, 2018.

[14] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, p. 54-71, May 2019.

[15] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1, 2021.

[16] "Madrid open data portal." Available at: https://datos.madrid.es/portal/site/egob, (Accessed: 14th December 2022).

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," vol. 9, p. 1735-1780, Nov 1997.

[18] B. Medina-Salgado, E. Sanchez-DelaCruz, P. Pozos-Parra, and J. E. Sierra, "Urban traffic flow prediction techniques: A review," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100739, Sep 2022.

[19] T. Sun, C. Yang, K. Han, W. Ma, and F. Zhang, "Bidirectional spatialtemporal network for traffic prediction with multisource data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, p. 78-89, Aug 2020.

[20] A. Ata, M. A. Khan, S. Abbas, M. S. Khan, and G. Ahmad, "Adaptive iot empowered smart road traffic congestion control system using supervised machine learning algorithm," *The Computer Journal*, vol. 64, p. 1672-1679, Nov 2021.

[21] S. Cheng, F. Lu, P. Peng, and S. Wu, "Short-term traffic forecasting: An adaptive st-knn model that considers spatial heterogeneity," *Computers, Environment and Urban Systems*, vol. 71, p. 186-198, Sep 2018.

[22] X. Wang, C. Chen, Y. Min, J. He, and Y. Zhang, "Vehicular transportation system enabling traffic monitoring: A heterogeneous data fusion method," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, p. 1-7, 2018.

[23] A. Usmanova, F. Portet, P. Lalanda, and G. Vega, "Federated continual learning through distillation in pervasive computing," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, p. 86-91, Jun 2022.

[24] Y. Huang, C. Bert, S. Fischer, M. Schmidt, A. Dorfler, A. Maier, R. Fietkau, and F. Putz Apr 2022. arXiv:2204.13591 [cs].

[25] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," Jun 2021. arXiv:2003.03196 [cs, stat].

[26] M. A. Hussain, S.-A. Huang, and T.-H. Tsai, "Learning with sharing: An edge-optimized incremental learning method for deep neural networks," *IEEE Transactions on Emerging Topics in Computing*, p. 1-13, 2022.

[27] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun. ACM*, vol. 63, p. 54-63, nov 2020.

[28] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Network*, vol. 32, no. 4, pp. 40-46, 2018.

[29] "Traffic state information in the city of barcelona." Available at: https://opendata-ajuntament.barcelona.cat/data/en/dataset/itineraris, (Accessed: 14th December 2022).

[30] "Permanent traffic counting stations - expressway s6 in gdansk (dataset containing 5-min aggregated traffic data and weather information) - open research data - most wiedzy." Available at: https://mostwiedzy.pl/en/open-research-data/permanent-traffic-counting-stations-expressway-s6-in-gdansk-dataset-containing-5-min-aggregated-traf,923120743943369-0, (Accessed: 14th December 2022).

[31] G. Albertengo and W. Hassan, "Short term urban traffic forecasting using deep learning," in *Proceedings of 3rd International Conference on Smart Data and Smart Cities*, (Delft, Netherlands), pp. 3–10, Copernicus, Oct. 2018.

[32] M. Kopp, D. Kreil, M. Neun, D. Jonietz, H. Martin, P. Herruzo, A. Gruca, A. Soleymani, F. Wu, Y. Liu, J. Xu, J. Zhang, J. Santokhi, A. Bojesomo, H. A. Marzouqi, P. Liatsis, P. H. Kwok, Q. Qi, and S. Hochreiter, "Traffic4cast at neurips 2020 - yet more on the unreasonable effectiveness of gridded geo-spatial processes," in *Proceedings of the NeurIPS 2020 Competition and Demonstration Track* (H. J. Escalante and K. Hofmann, eds.), vol. 133 of *Proceedings of Machine Learning Research*, pp. 325–343, PMLR, 06–12 Dec 2021.

[33] "Utd19. largest multi-city traffic dataset publically available." Available at: https://utd19.ethz.ch/, (Accessed: 14th December 2022).

[34] A. H. Aja, "Madrid centro: division en 'barrios funcionales'," *Cuadernos de Investigación Urbanística*, vol. 0, no. 50, 2007. Available at: http://polired.upm.es/index.php/ciur/issue/view/70, (Accessed: 14th December 2022).

[35] Z. Shen, W. Wang, Q. Shen, S. Zhu, H. M. Fardoun, and J. Lou, "A novel learning method for multi-intersections aware traffic flow forecasting," *Neurocomputing*, vol. 398, p. 477-484, Jul 2020.

[36] H. Lu, D. Huang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "St-trafficnet: A spatial-temporal deep learning network for traffic forecasting," *Electronics*, vol. 9, p. 1474, Sep 2020.

[37] E. L. Manibardo, I. Lana, and J. D. Ser, "Deep learning for road traffic forecasting: Does it make a difference?," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, p. 6164-6188, Jul 2022.

[38] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models." ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, July 2020. arXiv:2007.03051.

[39] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), p. 8024-8035, Curran Associates, Inc., 2019.

[41] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[42] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.