IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 August 2022, accepted 17 October 2022, date of publication 26 October 2022, date of current version 2 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217240

RESEARCH ARTICLE

Short-Term Prediction of City Traffic Flow via Convolutional Deep Learning

STEFANO BILOTTA, ENRICO COLLINI, PAOLO NESI, (Member, IEEE), AND GIANNI PANTALEO

Distributed Systems and Internet Technologies Laboratory, Department of Information Engineering, University of Florence, 50139 Florence, Italy

Corresponding author: Paolo Nesi (paolo.nesi@unifi.it)

ABSTRACT Nowadays, traffic management and sustainable mobility are central topics for intelligent transportation systems (ITS). Thanks to new technologies, it is possible to collect real-time data to monitor the traffic situation and contextual information by sensors. An important challenge in ITS is the ability to predict road traffic flow data. The short-term predictions (10-60 minutes) of traffic flow data is a complex nonlinear task that has been the subject of many research efforts in past few decades. Accessing traffic flow data is mandatory for a large number of applications that have to guarantee a high level of services such as traffic flow analysis, traffic flow reconstruction, which in their turn are used to compute predictions needed to perform *what-if analysis, forecast routing, conditioned routing, predictions of pollutant, etc.* This paper proposes a solution for short-term prediction of traffic flow data by using a architecture capable to exploit Convolutional Bidirectional Deep Long Short Term Memory neural networks (CONV-BI-LSTM). The solution adopts a different architecture and features, so as to overcome the state-of-the-art solutions and provides precise predictions addressing traffic flow data in cities, which are tendentially very noisy with respect to the ones measured in high-speed roads, the latter being the validation context for the majority of state-of-the-art solutions. The proposed solution has been developed and validated in the city context and data via Sii-Mobility, a smart city mobility and transport national project and it is currently in use in other contexts such as in Snap4City PCP EC, TRAFAIR CEF, and REPLICATE H2020 SCC1, and it is operative in those areas.

INDEX TERMS Traffic flow, short-term predictions, machine learning, deep learning, CONV-BI-LSTM.

I. INTRODUCTION

Traffic flow measuring is central for intelligent transportation systems (ITS). According to recent technologies, real-time traffic flow data can be measured, collected and exploited. The knowledge of real-time traffic flow data enables the development of a large number of services such as congestion detection and reduction; computing of origin-destination matrices; incident management; optimization of existing infrastructures of public transport; dynamic network traffic control; improved information services (e.g., traffic information, dynamic route guidance, road digital signage, planned routing); plan for future investments on mobility solutions; reducing fuel consumption and emissions of both CO2 and NO2 that strongly depend on fossil combustion (and thus on traffic, as well); predicting NOX [1]. See for example, the European Commission 2008/50/EC Directive on Ambient Air Quality and Cleaner Air for Europe and 2004/107/EC Directive on heavy metals and polycyclic aromatic hydrocarbons in ambient air.

On this regard, traditional methods for traffic flow measuring via spire sensors (inductive loops) [2], as well as TV cameras to be located in several points of city roads can obtain equivalent measures in terms of traffic flow density, velocity, and number of vehicles. Surrogated traffic flow data can be obtained from App on mobile devices, as well as from on board units, and social media [3], etc. For example, in [3] and [4], a smartphone-based crowd sensing system for traffic detection and measure has been proposed, where data are gathered from handheld devices. In [5], a deep Restricted

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

Boltzmann Machine and Recurrent Neural Network, RNN, architecture have been used to predict traffic congestion evolution based on GPS data from taxis. Data coming from navigator Apps (e.g., TomTom, Google map, Waze) could be very expensive for a municipality with respect to the installation of sensors and measures, and they could be not related to the actual counting of vehicles. In [6], an interesting and comprehensive review of the different approaches has been provided.

The exploitation of TV Cameras located in specific critical points reduces costs and increase the precision of the measure in specific points, since multiple areas can be controlled with a single installation, thus enabling the control of a high number of traffic flows points. Traffic flow sensors provide continuous measuring of the traffic on selected roads at fine grain, while giving, in most cases, also information about the vehicle kind: busses, tracks, cars, bikes, etc.

On traffic flow sensors, a variety of dysfunctions can be experienced, thus causing a lack of data: network failure, broken device, wrong data production, and byzantine errors. When fault is temporary, an accurate short-term traffic prediction model could solve that problem by providing missed real-time data. An anomaly detection algorithm can be adopted to alert the municipality about the device failure to start fixing the problem [7]. An anomaly detection can trigger the activation of predictions in place of missing data. In alternative to predictions, typical time trends taking into account daily, weekly seasonality can be provided.

Most of the above-mentioned services consume traffic flow data with a high continuity to provide a needed level of quality for real-time services. Among them, (i) *traffic flow reconstruction* to compute a traffic flow estimation in each segment of the road network [8]; (ii) *conditioned and predictive routing* for rescue teams, fire brigade; *or* (iii) *what-if analysis* in critical conditions. Traffic flow data coming from stationary scattered sensors can be combined with short-term predictions of traffic flow to reduce discontinuities in the above-mentioned services, thus accepting a certain level of error that could maintain the needed service level and prevent any infringement of relevant constraints within service level agreements.

For the above-mentioned reasons, in literature, traffic flow predictions have attracted extensive research efforts. In [9], [10], [11], [12], and [13], traffic state analysis is related to the monitored areas in terms of short-term traffic flow prediction on fixed points. In [9] and [10], the theoretical bases for modelling univariate traffic condition data streams as seasonal autoregressive integrated moving average processes are considered. In [11] and [13], the problem of short-term prediction has been assessed in freeways through deep learning models exploiting historical information only. In [14], authors discussed the usage of Random Forest, RF, model for short-term prediction of traffic flow and achieved an accuracy of about 94%. In [15], neural networks, RF, gradient boosting machine, GBM, and a generalized linear model have been investigated for short-term prediction of traffic volume, speed, occupancy of a single roadway segment. Authors have applied the model only on 1.3-mile section of westbound Interstate 64 (I-64) in St. Louis, Missouri, in the United States and obtained an accuracy of about 92%, while exploiting historical information to predict traffic conditions. In [16], three methods for short-term traffic prediction of a single road have been compared, i.e., CNN (Convolutional Neural Network) [17], GRU (Gated Recurrent Unit) [18], GRU+STFSA (Spatio-Temporal Feature Selection Algorithm), based on exploiting historical information in a period of 40 working days. In [19], traffic volume is predicted on highway domain, using characteristics as weak time continuity, structural space topology and wider spatio-temporal correlation. In the literature, CNN architecture has been profitably used for person re-identification [20], [21].

In the above presented cases, only simple network areas such as freeways segments or rings have been considered to compute short-term predictions, based only on the historical data of the traffic flow. More extended cases are reported in Section 2 of current paper.

This paper presents a solution to compute short-term traffic flow sensors predictions up to 1 hour in advance, with a resolution of 10 minutes. The proposed results are innovative since the solution proposed:

- overcomes the state-of-the-art solutions in terms of precision and it is based on a never used architecture for the purpose: Convolutional Bidirectional Deep Long Short Term Memory neural networks, CONV-BI-LSTM (this solution has been compared with respect to RF, XGBOOST, and other 6 deep learning techniques as described in the following),
- clarifies which are the features actually relevant (historical, seasonality, weather, pollutant, etc.) in prediction computation, thus providing errors in all possible feature combinations for a large range of different machines and deep learning techniques vs the proposed solution, aiming at covering all cases reported in the literature for total of 512 combinations,
- has been validated in a complex urban network of a real-world road structure, which is an aspect totally different from most solutions only tested on high-speed roads which have less noisy and quite regular traffic flow conditions. The validation has been performed in the Florence area; Italy as accessible on Snap4City.org. The solution has been assessed in terms of impact of missing data and performance for training and execution.

The solution has been implemented in the context of Sii-Mobility project and infrastructure (national smart city project of Italian Ministry of Research for terrestrial mobility and transport, https://www.sii-mobility.org). Sii-Mobility is based on Km4City model and tools (https://www.km4city.org) [22], [23]. Sii-Mobility is at present covering the whole Tuscany region, Italy, which means 3.5M inhabitants and 40M of tourists per year. The proposed solution is at the basis of Snap4City on traffic flow analysis and reconstruction, and Trafair CEF for computing NOX production from traffic [22], and it is currently exploited in the Smart City Control Room for Florence area according to REPLICATE H2020 SCC1 project. Moreover, as to Florence, Pisa and Livorno municipalities in Tuscany region traffic flow data are used for traffic flow reconstruction and for other services [8].

The paper is structured as follows. Section II provides a description and comparison of the related deep learning works for short-term traffic flow prediction. Section III provides a description of data and features for traffic flow prediction. Traffic flow data have been analysed to see their typical behaviour and a large set of features has been considered. In Section IV, a set of machine and deep learning approaches has been adopted. The aim has been to identify the most effective short-term predictive models and the best features to be used. Section V contains a summary of the experimental and validation results conducted. The section focuses on the comparison of predictive models (short-term as 1 hour, every 10 minutes) exploiting data collected within Florence city area for traffic sensors, in order to identify the best resulting approach in terms of prediction error (the data used are accessible for all Snap4City open platform users and as Open Data on other portals). In this case, the considered traffic data are not collected from regular highways, as it often occurs in many states of the art cases; instead, they are data gathered from city sensors which are more complex to be managed for their variability and rapid changes. The impact of missing data has been analytics, as well as the performance in training and execution. Conclusions are drawn in Section VI.

II. RELATED WORKS

In literature, the problem of traffic flow predictions and related problems has been addressed through different approaches, most recent review works are using deep learning techniques [47], [48], [49], [50]. Many of the latter are used in time-series prediction, and in particular with deep recurrent neural networks because of their capability of using information at a certain instant, as well as past data from previous observations. In most cases, the adopted data have been collected from high-speed roads, showing typically more regular flows than urban flows.

In **Table** 1, a comparative summary of the state-of-the-art solutions is reported. Such comparison highlights the predictive target, the adopted features, the used technique, and the obtained results in terms of RMSE (Root Mean Square Error), MAPE (mean absolute percentage error), R2 (R Squared), and MAE (Mean Absolute Error), according to what has been published by authors.

Tian and Pan [24] used a LSTM (Long short-term memory) neural network for traffic flow predicting with targets of 15, 30, 45, 60 minutes in advance. The result has demonstrated that LSTM performed better than Random Walk (RW) [25], Support Vector Machine (SVM) [26], single-layer Feed-Forward Neural Network (FFNN) [27], and Stacked AutoEncoder (SAE) [28]. The used dataset is the Caltrans Performance Measurement System (PeMS) as in [2]. They used only traffic flow data for the prediction, without considering other factors such as spatial impact from neighbor observation stations, weather conditions, accidents, density, speed, which could have improved such predictive results.

Kang, Lv, and Chen [29] also used LSTMs for the PeMS dataset and focused their work on studying the effects of various inputs for short-term traffic flow prediction. They reported that, when including also speed and occupancy features, this led to better results. In their case study, results were improved by using other data collected by previous and subsequent stations, with respect to the target one, which is an approach unfeasible in city road networks.

Z. Wang et al. [30] compared different machine learning models for the short-term prediction of traffic flow in 3 datasets: PeMS, Kunming Regional Interchange Station traffic flow data provided by the China Yunnan Academy of Transportation Science and the Ireland's Nra traffic data. The LSTM architecture achieved better results compared to Backpropagation neural networks, SVR [31], RNNs [32], but one of the main pillars of their work is that the regularization of the LSTM with a recurrent dropout and also a max normalization weight constraint led to better results compared to the LSTM without regularization.

Mou et al. [33] proposed a temporal enhanced LSTM (T-LSTM) for short-term prediction of traffic flow in Bejing, reporting a comparison with several other architectures, i.e.: SAE, DBN (Deep Belief Networks) [34], GRU, LSTM, SVM, KNN (K-nearest neighbor) [35], FFNN [27], ARIMA (1,0,1) (Autoregressive Integrated Moving Average model) [36] and The dataset used for validation included traffic flow, speed, density and date. The proposed temporal enhanced LSTM architecture achieved its best results for predicting traffic flow 16 minutes ahead. Zhang et al. [37] compared ARIMA (0,1,1) [36], RNN, LSTM and GRU for the prediction of future 12 hours traffic flow. The considered dataset is the PeMS, but they included also meteorological features in the dataset such as average wind speed, weather types, average temperature and precipitation. Results demonstrated that 2 layers hidden GRU based on deep neural network achieved best predictive results. Traffic is one of the main sources causing air pollution in the cities. As to air quality prediction, much work is already available in literature, where air pollution has been predicted using road traffic data. Awan et al. [38] used LSTM networks for the task of 1 hour traffic flow prediction, in the case study based on Madrid, Spain. They included not only traffic measurements, but also meteorological and air pollution features in their dataset. This type of data inclusion could improve results on traffic flow prediction based on 16 considered sensors. Abduljabbar et al. [39], predicted speed and traffic flow on 2 Australian freeways: Pacific Motorway, Queensland, and Tullamarine Freeway, Melbourne. They also applied transfer learning to another freeway, the Southeastern Freeway, Melbourne, for traffic flow prediction with targets of 5', 15', 30', 45' and 60'. They implemented a Bidirectional-LSTM

TABLE 1. Related works table, focus on deep neural network solutions for traffic flow predictions based on sensor data.

Authors	Target	Features	Dataset	Model	Resutls
Y.Tian and L. Pan [23] (2015)	Traffic Flow: 15’, 30’, 45’, 60’	Only traffic flow data	PeMS 5’ intervals Highway	LSTM	RMSE 50.94 MAPE 6.49
D. Kang, Y. Lv and Y. Chen [29] (2017)	Traffic Flow 5’	Traffic flow, speed, occupancy	PeMS 5’ intervals Highway	LSTM	RMSE 23.45 MAE 15.97 MAPE 12.84
Z. Wang et al. [30] (2019)	Traffic Flow: 15’, 30’, 45’, 60’	Traffic flow, speed, occupancy, neighbor flows	PeMS 5’ Highway	LSTM with regularization	RMSE 7.9 MAE 4.1
Mou et al. [33] (2019)	Traffic flow 16’	Traffic flow and time label	data from Beijing Highway	T-LSTM	RMSE 50.54 MAPE 6,09
Zhang et al. [37] (2018)	Traffic flow 12h	average wind speed, precipitation, max temp., min temp., ice Fog, freezing Fog, smoke, the day is weekend, hour of the day, day of the week, vehicle miles travelled	PeMS 1h Highway	2 hidden layers GRU	RMSE 0.0019 MSE $3.76\times10{-}5$ MAE 0.0079
Awan et al. [38] (2020)	Traffic Flow 1h	Month, day, weekday, hour, CO, NO, NO2, NOx, O3, pressure, temperature, wind direction, wind speed, traffic flow	Data form Madrid, from 16 sensors Urban	LSTM 16 sensors	min / max MAE 0,061 / 0,214 MSE 0.009 / 0.60
Abduljabbar et al. [39] (2021)	Speed and traffic flow: 5’, 10’ 15’, 45’, 30‘, 60’	Speed and traffic flow	Australian freeways (PM) Pacific Motorway in Queensland; (TF) Tullamarine Freeway in Melbourne Highway	Bi-LSTM results for 15’	PM / TF Acc % 92.50 / 99.99 MAPE 7.49 / 0.01
C. Ma, G. Dai and J. Zhou [40] (2021)	Traffic flow: 5’, 10’, 15’, 20’	Traffic flow	Traffic flow Ningbo Meteorological Road Urban	LSTM-BILSTM Results for 5’	RMSE 16.72 R2 0.86 MAE 12.63 MAPE 6.25
Liu et al., [41] (2017)	Traffic flow 30’’	Traffic flow	PeMS 30’’ data Highway	CONV-LSTM BILSTM DNN	RMSE 6.419 MAE 4.408 MAPE 6.989
Essien et al., [45] (2021)	Traffic Flow 12 intervals: 5, 10, 15 ‘ up to 60’	Speed, Traffic Flow and Density, temperature, precipitation, ..	Greater Manchester Highway	Autoencoder BI-LSTM	MAE 5.5049 RMSE 6.8579 R2 93.45
Verma [46] (2021)	Traffic Flow 5’ 15’ 30’	Traffic Flow and Speed	PeMS Highway	CNN_LSTM with attention	MAE 2.4 RMSE 3.4

(BI-LSTM) model that achieved better results compared to other unidirectional recurrent RNN and LSTM architectures. Ma, Dai and Zhou [38] performed a time series analysis on traffic flow data and performed smoothing and standardization processing to obtain a stable time series for their training process of machine learning models. The case study is based on the urban road section Ningbo Meteorological Road, China. They proposed a model is composed by a LSTM layer followed by a BI-LSTM layer and another LSTM connected to a last Fully Connected layer. The LSTM-BI-LSTM neural network for short-term traffic flow prediction on this urban road section achieved results around 6% in terms of MAPE, as to 5 minutes traffic flow prediction. Liu et al., [41] used a deep architecture model with Convolutional-LSTM module combined with a BI-LSTM module to extract space-temporal features of traffic flow as input to a fully connected layer, so as to obtain short-term prediction of traffic flow. The used dataset is the PeMS dataset and the proposed model performed better than ARIMA, SAE, LSTM, SVM. Polson and Sokolov [42] developed a learning model to predict 40 minutes traffic flow from the Interstate I-55 Chicago. The sparse linear vector autoregressive model has been combined with a median data pre-filtering technique. They achieved the best results also in two events: a football game and a snowstorm situation. Liu et al [43] proposed a neural network architecture named DeepTSP (Deep Traffic State Prediction) that is made up of a convolutional network that extracts the features from the images of Berlin, Istanbul and Moscow combined with multi source data as weather condition, holiday, the day of the week and the time of day. The solution achieved better results than the ST-ResNet that represented the stoa solution for Spatio-Temporal prediction. The Spatio-Temporal solution proposed by Yao et al. in [44] to predict the traffic flow on the NYC taxi and bike-sharing datasets (not on traffic flow data), combines the CNN and LSTM architectures with a flow gating mechanism and a periodically shifted attention mechanism. Essien et al. [45] used in combination with the traffic and weather features also data retrieved from the Social Network Twitter. The Autoencoder BI-LSTM architecture has been used to predict the traffic flow using also the count of the Tweets from road traffic information users for the 12 temporal targets of 5, 10, 15, 20, ..., 60 minutes. Also, Verma [46] on the PeMS dataset achieved an improvement in the prediction results of Traffic Flow for the next 5, 15, and 30 minutes using an attention mechanism combined with a CNN-LSTM architecture.

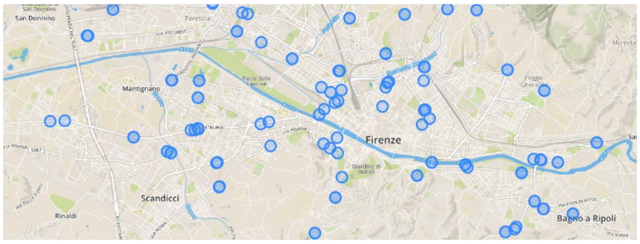


FIGURE 1. Map of the traffic sensors' locations in Florence municipality.

III. DATA FOR TRAFFIC FLOW

A. TRAFFIC FLOW DATA

As mentioned in the introduction, our main goal was to find a solution to predict traffic flow in the locations of traffic sensors. In the considered scenario, traffic flow is typically registered every 10 minutes by each traffic sensor. The exploited data refers to 135 devices located in the municipality of Florence area as depicted in **Figure** 1. Please note that, each device sensor location may measure traffic flow on both sides of the road and on multiple lanes. Therefore, each location may correspond to two distinct device logic sensors.

Trends of traffic flow data are strongly dependent on a number of road features: road relevance (primary, secondary, etc.), number of lanes, speed limits, presence of speed meters, distance from road crossing, etc. Moreover, a certain class of roads (e.g., the so-called primary/main roads of the open street map), may provide higher capability with respect to local, single lane cases. In order to characterize the typical time trend H24 of the whole traffic flow sensors located in the city, a clustering was carried out. This approach allowed us to aggregate device sensors with the same behaviour over time. The range of considered data goes from September 2019 to February 2020. A wider data set, over a larger time period, has been also used without obtained any better estimation performance, since traffic evolves over time.

As a first step, we have tested cluster tendency by measuring the probability that a given data set has been generated by a uniform data distribution using the Hopkins statistics [51]. The Hopkins statistic value resulted to be equal at 0.86, therefore proving significantly the data set cluster-ability. As a second step, K-means clustering method has been applied to identify clusters of traffic flow sensors. Please note that, K-means assigns each item to the cluster having the nearest centroid. In K-means clustering, there is an ideal centre point representing a cluster [52]. The clustering has been performed on the basis of the time trend H24, considering the normalized traffic flow measures. The optimal number of clusters turned out to be 3 and it has been identified by using gap statistic criteria [53]. In **Figure** 2, the identified clusters have been represented on map, assigning a different colour pin for each cluster.

The representative sensor for cluster 1 (primary roads) is METRO775, for cluster 2 (downtown area) is METRO707 and for cluster 3 (suburban area) is METRO714.

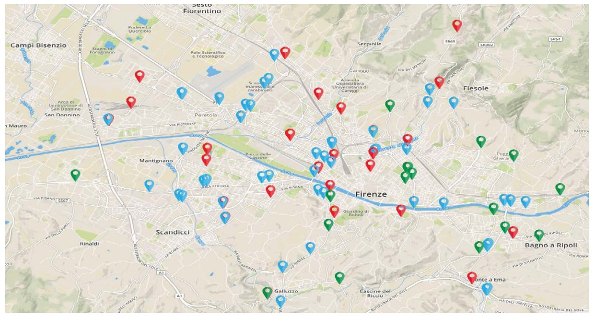


FIGURE 2. Map of the traffic sensors' locations per cluster in Florence municipality (blue pins: Group 1; red pins: Group 2; green pins: Group 3).

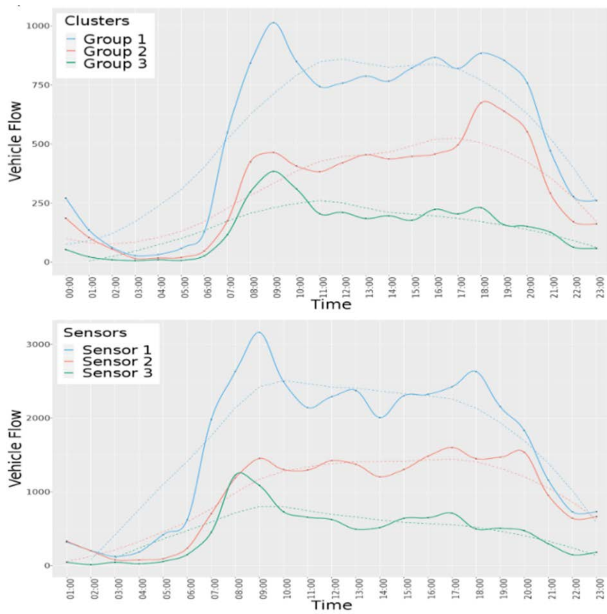


FIGURE 3. (a) Hourly median traffic flow trends per cluster (Group 1, Group 2, Group 3) and (b) hourly average traffic flow trends per representative sensor (Sensor 1, Sensor2, Sensor 3) in each cluster.

Figure 3 (a) depicts the hourly median traffic flow trends for each cluster and **Figure** 3 (b) shows the average traffic flow trends of the three most representative traffic flow sensors for each cluster. The three trends are mainly describing situations where: (1) a peak is registered in the morning and a second peak is also present in the evening and this cluster is characterized by a high flow of vehicles; (2) an almost stable traffic is present in the whole day working hours, characterized by medium flows; (3) the peak of traffic is registered in the morning, from 7:00 to 9:00 while along the rest of the day the traffic flow tends to decrease.

B. OTHER FEATURES

One of the goals of this work has been conquer a general understanding above the factors that are more relevant for predicting traffic conditions in the city. Based on the related works, a set of data composed of temporal variables, traffic-related features, weather information, and air pollution has been considered in present paper. Therefore, the identified larger set of features has been classified and presented in **Table 2**. The data related to traffic measurements have been divided into two categories. The *Traffic* category includes the TrafficFlow metric at the observation time that refers to the number of vehicles detected by sensors, while *TrafPlus includes* other measures coming from traffic sensors, such as: the vehicles' AverageSpeed (km/h) and the Concentration which is a punctual measure expressed in percentage.

The *DateTime* category includes the timeOfTheDay metric, encoded with a number that ranges from 1 to 144, since traffic flow data are measured and collected every 10 minutes. Typically, these values are used to also consider the data seasonality that may have different trends, e.g., working days with respect to weekends. Usually, the trend related to the vehicle number is similar on the same day of the week (e.g., Monday of current week with respect to Mondays in past weeks). Features related to the *Seasonality* are dayOfTheYear, dayOfTheWeek, Weekend, and Year.

In the *Temporal* category, three time-based features have been included in the model in order to take into account:

(i) dP: the difference between the vehicle number in the observation day d at time t and the available vehicle number during the previous time slot $t\text{-}1$ of the previous day, $d\text{-}1$.

$$dP = VehicleFlow_{d,t} - VehicleFlow_{d-1,t-1}$$

(ii) dS: the difference between the vehicle number in the observation day, d, at time t and the available vehicle number during the successive time slot $t+1$ of the previous day, $d\text{-}1$.

$$dS = VehicleFlow_{d,t} - VehicleFlow_{d-1,t+1}$$

(iii) PwVF: the vehicle number of the previous week $d\text{-}7$ in the same time slot, t :

$$PwVF = VehicleFlow_{d-7,t}$$

Features belonging to the *Weather* category are also collected every 10 minutes (i.e., *Air Temperature, Humidity, Pressure, WindSpeed).*

In order to complete any possible descriptive features for the phenomena and thus for the input dataset, we included data regarding the measured pollutants, *AirPoll* category, such as: CO, NO2, O3, PM10, and PM2.5.

IV. SHORT-TERM PREDICTION MODELS

In this section, machine learning techniques are compared with the aim of creating a solution to predict the traffic flow for the most representative sensors resulted from the previously referred clustering process (see **Section** III, A) with a temporal target of 1h, which is the most critical short-term prediction slot. Ensemble learning techniques such as Random Forest (RF) and Extreme Gradient Boosting Machines (XGBOOST) [54] are powerful techniques that must be considered for this type of problem. Regarding the deep learning techniques for this work, we have compared the Deep Neural Network (DNN), Deep LSTM [55], Deep BI-LSTM Neural Network [56], Autoencoder BI-LSTM, and an attention-based CONV-LSTM, with respect to a architecture defined in this paper as CONV-BI-LSTM. Therefore, these different approaches have been compared to consider any best results in both the state of the art and our solutions: RF (as in [14]), XGBOOST, DNN, LSTM (as in [23], [29], and [38]), BI-LSTM (as in [39]), Autoencoder BI-LSTM (as in [45]), attention-based CONV-LSTM (as in [46]), and CONV-BI-LSTM (which is the approach we have proposed in this paper). Such different solutions have been applied in 64 combinations of feature category, as presented in **Table** 1, with the aim of determining which one could be the most relevant for current purpose and for which architecture. Some combinations of features and solutions correspond to the ones

TABLE 2. **Overview of the features used in the short-term prediction models.**

Category	Feature	Description
Traffic Trafplus	*Ytaffic Flow*	Real number of vehicles recorded every 10 minutes
	AverageSpeed	Average speed of vehicles (Km/h)
	Concentration	Number of vehicles in terms of road occupancy (%)
DateTime	*timeOfTheDay*	Time of the day {1, 144}
	dayOfTheYear	Day of the year {1, 366}
seasonality	*dayOfTheWeek*	Day of the week {1,7}
	Weekend	0 for working days, 1 else
	Year	The year of the observation
Temporal	Previous observation's difference of the previous week (*dP*)	the difference between the number of vehicles in the observation day (d) at the time slot t and the number of available vehicles during the previous time slot (t-1) of the previous day (d-1)
	Subsequent observation's difference of the previous week (*dS*)	the difference between the number of vehicles in the observation day (d) at the time slot t and the number of vehicles during the successive time slot (t+1) of the previous day (d-1).
	Previous week observation (*PwVF*)	the number of vehicles of the previous week (d-7) in the same time slot (t).
Weather	*Air Temperature*	City temperature one hour earlier than *Time* (°C)
	Humidity	City humidity one hour earlier than *Time (%)*
	Pressure	City pressure one hour earlier than *Time (millibar mb)*
	Wind Speed	City wind speed one hour earlier than *Time (KM/h)*
AirPoll	*CO*	Concentration of CO one hour earlier than *Time*
	NO2	Concentration of NO2 one hour earlier than *Time*
	O3	Concentration of O3 one hour earlier than *Time*
	PM10	Concentration of PM10 one hour earlier than *Time*
	PM2.5	Concentration of PM2.5 one hour earlier than *Time*

in literature, resulting as a systematic comparison, as shown in **Table 6**.

These models were evaluated in terms of statistical measures such as R-squared (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE) [57]. The MAPE metric is the one used to compare techniques to choose the best model architecture for the task of short-term predictions. The R^2 is calculated as follows:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} obs_i$$

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{\sum_{i=1}^{n}(obs_i - \bar{y})^2}\right)$$

The MAE is calculated as follows:

$$MAE = \frac{\sum_{i=1}^{n}|obs_i - pred_i|}{n}$$

The MAPE is calculated as follows:

$$MAPE = \frac{\sum_{i=1}^{n}|\frac{obs_i - pred_i}{obs_i}|}{n} * 100$$

The RMSE is calculated as the Root square of the Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{n}$$

$$RMSE = \sqrt{MSE}$$

The MASE is calculated as follows:

$$MASE = mean\left(|q_t|\right), \quad t = 1, \ldots, n$$

and

$$q_t = \frac{obs_t - pred_t}{\frac{1}{n-1}\sum_{i=2}^{n}|obs_i - obs_{i-1}|}$$

where:

$$obs_i = observation\ at\ time\ i,$$
$$pred_t = prediction\ at\ time\ t,$$

n is the number of the values in the test set.

As to the implementation of ensemble learning techniques, the number of trees parameter for the RF was set to 300, with a minimum sample split set equal to 2, minimum number of samples allowed for a leaf equal to 1, without limits on both the maximum number of features considered to split a node and the number of leaves, with the construction of bootstrapped datasets for creating the trees.

XGBOOST regressor uses the least-squares loss function with learning rate optimized with values 0.1, 0.01, and 0.001 with max depth equal to 3 and minimum sample split, minimum sample leaf, maximum number of features equal to the ones chosen for the RF.

As to DNN, the developed deep neural networks have a 4 layers deep architecture: the first three layers are fully connected with a Leaky-Relu activation, and the final layer provided one neuron to make the prediction 1 hour in the future and has a sigmoid activation to obtain a value in the range [0,1]. Indeed, the input data has been normalized using a Min Max scaler. The adopted hyperparameters have been tuned via Random Search among the ones reported in **Table** 3, by using the minimum RMSE on the validation set to select the best configuration.

TABLE 3. Hyperparameter optimized for the DNN for the prediction target of 60 minutes.

Hyperparameter	Values
num_unit_first_layer	256,128,128,256,512
num_unit_second_layer	128,64,128, 256, 512
num_unit_third_layer	64,32,64,128,256
dropout_rate_fist_layer	0.1, 0.25, 0.5
dropout_rate_intermediate_layer	0.1, 0.25, 0.5
dropout_rate_final_layer	0.1, 0.25, 0.5
learning_rate	0.05,0.005,0.0005,0.00005
batch_dim	32,64,128,256,512

TABLE 4. Hyperparameter optimized for the LSTM/BI-LSTM for the prediction target of 60 minutes.

Hyperparameter	Values
num_LSTM/BI-LSTMs_units	14, 21, 28
dropout_rate_fist_6_layers	0.1, 0.2, 0.3
batch_dim	32,64,128,256,512
learning_rate	0.05,0.005,0.0005,0.00005

The LSTMs and BI-LSTMs networks have a 7-layers architecture structured such as: the first 6 layers are made of LSTMs or BI-LSTMs units depending on the considered network. The last layer is a fully connected one with one neuron and sigmoid activation to obtain the prediction. The implementation of recurrent neural networks is stateful with a number of timesteps considered equal to 6 that corresponds to the data of one hour prior to the observation/prediction time. The training process has been made with early stopping with patience set to 100 and weights restored to the best model. The hyperparameters that have been optimized through a Random Search are reported in **Table** 4. In the training process, the minimum RMSE on the validation set has been used to select the best configuration.

The structure of the defined CONV-BI-LSTM network is reported in **Figure**4 and it is made up of 3 components:

- The first component is made up of a Convolutional 1-dimensional layer with 48 filters and a kernel size of 16, and a Max Pooling layer of 2 × 2 and stride equal to 1.
- The second component is the BI-LSTMs layers, in particular 6 layers with 32 units per layer and dropout of 0,25. This value refers to the final version.
- The last one is made of 3 fully connected layers with number of neurons of 32-16-1. The last one has a sigmoid activation to produce the prediction.

TABLE 5. Autoencoder BI-LSTM network architecture.

Layer	Number of units
1) BI-LSTM	*n_unit*
2) BI-LSTM	*n_unit / 2*
3) BI-LSTM	*n_unit / 4*
4) BI-LSTM	*n_unit / 4*
5) BI-LSTM	*n_unit / 2*
6) BI-LSTM	*n_unit*
7) Fully Connected	*n_unit*
8) Fully Connected	1

The used optimizer is Adam Optimizer with learning rate between 0.005 and 0.008. MSE was selected as the loss function to be monitored during optimization. The batch size has been set to 512 and the number of epochs was set to a maximum value of 1000, because the training strategy used the Early Stopping method with patience parameter set to 100 to determine the optimum epoch number minimizing the RMSE of the validation set, restoring the weights of the best model at the end of the learning process. For the CONV-BI-LSTM the only parameter optimized has been the learning rate obtaining as final value 0.005; thus not dropout optimization in the final training. The model proposed has been trained with the two possible learning rates and based on the minimum RMSE on the validation set it has been chosen the best configuration. The rationale behind the addition of a convolutive layer was the advantage of combining powerful feature extraction of CNN with LSTM capability in capturing temporal dependencies. CNN are useful for learning local features in time series [58], since they can perform an optimized smoothing of noisy input data, while maintaining the underlying data trend. Therefore, convolutive layers can improve performances of subsequent LSTM layers in learning temporal dependencies [59]. In [46] and [60], CNN-LSTM architectures have been employed for traffic speed prediction. As to the adoption of BI-LSTM layer, a BI-LSTM model has the characteristic of being trained twice, first by feeding input data to an LSTM layer, and then feeding the same input dataset (but on reverse order) to another LSTM layer. This has shown improvements in time-series analysis and forecasting [61].

The structure of the Autoencoder BI-LSTM network is made up of 8 layers as for the work in [45] and the structure of the network (reported in the **Table** 5) depends on the hyperparameter *n_unit* that has been optimized between 16, 24, 28. The other hyperparameters optimized have been the batch size, the dropout rate for the first 6 layers and the learning rate with the same values reported in **Table** IV. In this case, the Sklearn Random Search has been used [62].

The Attention-based CONV-LSTM, similarly to the work in [46], is made up of 9 layers: the first one is an Attention Layer followed by a Fully connected layer with 32 neurons. Then there are 3 Convolutional 1-dimensional layers with 32 filters followed by 3 LSTM layers with number of units optimized using the values of **Table** 4 as for the other 2 hyper-parameters optimized that are the batch size, the learning rate, using Sklearn Random Search [62]. The dropout rate of the first Fully connected and for the LSTM layers has been set to 0.2.

V. EXPERIMENTAL RESULTS

According to data and remarks reported in previous sections, the identified challenge was not only to find the best architecture to predict the traffic flow with a resolution of 10 minutes for the next hour, but also to discover the most informative set of features for the analysed models. The training set considered included data from 09/09/2019 to 02/02/2020. The two weeks from 03/02/2020 to 16/02/2020 have been used as follows: the first week as validation set and the second one as test set, for all techniques. The approach reduces the problem of the so-called cold start period. We have much longer time periods of data into Snap4City platform and service. On the other hand, we have tested larger data sets without obtaining better results due to data variability over long time periods. As shown in the following, with the above-described limited amount of data it is possible to train deep learning models as well as Ensemble Learning techniques for short-term traffic flow predictions and obtain satisfactory results with respect to the state of the art.

According to the state-of-the-art, to derive short-term predictions of traffic flow by exploiting only historical traffic flow data is not always the best solution. On the other hand, a large set of features may not always produce better results even in big data deep learning approaches.

In order to better understand the influence of each feature category, we have collected a large set of features, as described in **Table** 2: traffic, datetime, seasonality, temporal, weather and air pollutants. Assuming the *Traffic* category mandatory as input for the construction of any predictive model, the number of combinations of the other 6 feature categories reaches 64. Thus, we have trained, tested, and validated all 64 combinations vs the traditional ML and deep learning and CONV-BI-LSTM) (see **Table** 6), which included also the ones used in literature. The aim was to identify the best model, and at the same time to understand which are the most relevant features. From the analysis of results reported in **Table** 6, it seems that RF gets benefit for the presence of the DateTime (**Table** 6 rows from C1 to C32) feature which explicitly describes the time series over the day. On the other hand, CONV-BI-LSTM obtains very similar results. Moreover, CONV-BI-LSTM model also includes time series modelling, since it requires temporal sequences as input data, obtained from the original time series by using a sliding temporal window. The same approach is used in LSTM and BI-LSTM. In fact, in absence of the DateTime (**Table** 6 rows from C33 to C64) they obtain in most cases the better results with respect to the other methods.

The best results have been achieved by the predictive models presenting a convolutional layer which efficiently extracts local features in noisy data (**Table** 6 rows C32 – C6 – C28).

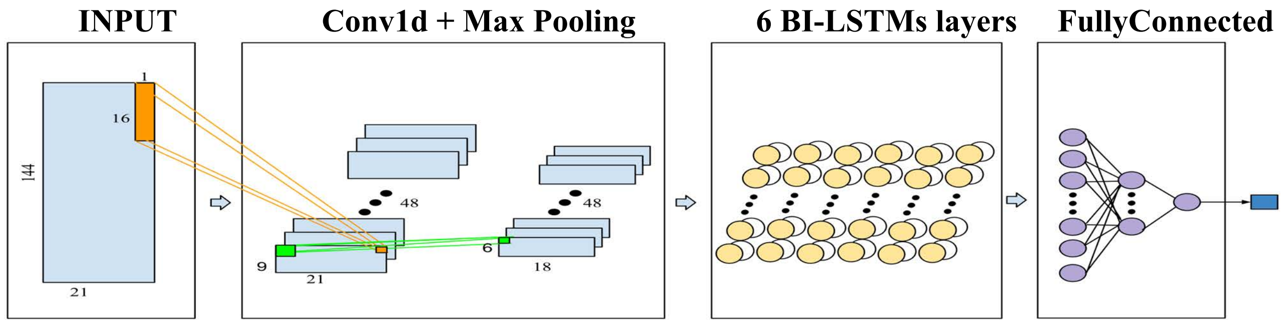


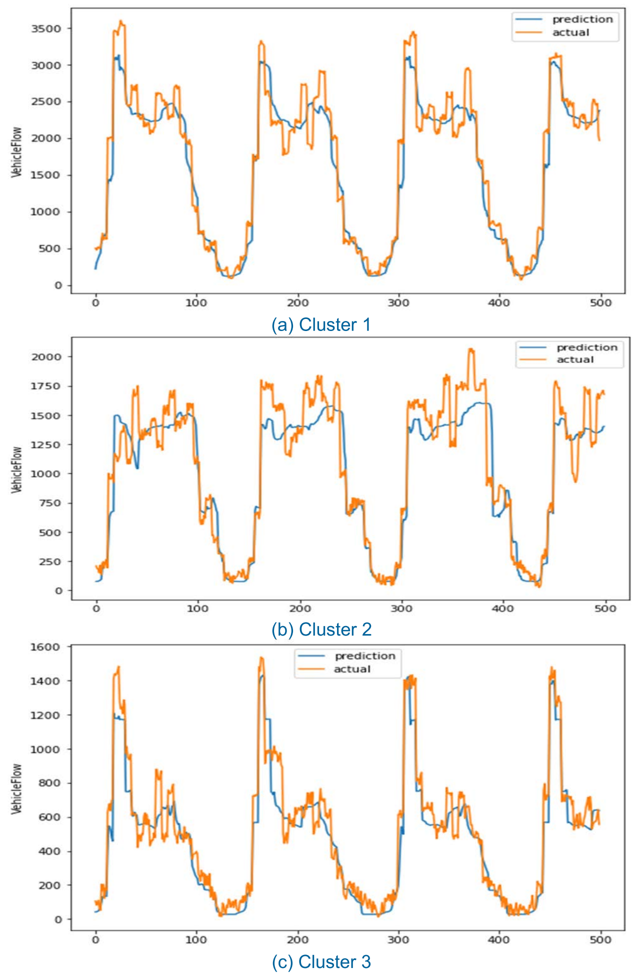
FIGURE 4. Graphical representation of the CONV-BI-LSTM network used for the traffic short-term prediction.

While a bidirectional approach improved, in most cases, the classic LSTM by performing an additional training on reversed order input dataset, which seems to lead to a better understanding of the underlying context also in time series data [61]. CONV-BI-LSTM approach worked in a quite satisfactory manner without considering Weather and Airpoll features, whereas RF generically benefits from such presence. Moreover, other features positively could contribute to the precision in terms of MAPE, but the impact of categories Trafplus, Temporal, Seasonality is not as evident on **Table** 4 as for the Weather and Datetime. For this reason, it has been conducted a detailed analysis reported in the next Section V.A.

In order to identify best prediction model, machine learning solutions were compared based on the MAE, MAPE and RMSE for short-term prediction of TrafficFlow 1 hour in advance, for every possible combination of feature categories as in **Table** 2. Typical trends for prediction are reported in **Figure 5** according to different clusters using the CONV-BI-LSTM referring to the configuration with minimum MAPE in **Table** 6 C32.

Overall, the best predictive model architecture for 1-hour short-term prediction of TrafficFlow for the representative sensors is CONV-BI-LSTM, as it can be observed from **Table 6**. CONV-BI-LSTM achieved minimum errors for the evaluated metrics if compared with the ensemble learning methods RF and XGBOOST, and the Deep Learning architectures DNN, LSTM and BI-LSTM. Noteworthy is this aspect: best three results are the ones obtained without weather features (**Table** 6 rows C32 – C6 – C28).

Best results per cluster are reported in **Table** 7 in terms of R^2, MAE, RMSE, MAE, MAPE for the CONV-BI-LSTM referring to the configuration with minimum MAPE in **Table** 6 C32.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

FIGURE 5. Plots of the predictions made by the CONV-BI-LSTM of the traffic flow for (a) Cluster1, (b) Cluster2, (c) Cluster3.

A. FEATURE CATEGORY IMPORTANCE ANALYSIS

The problem of short-term prediction in traffic flow measurements can be tackled with a univariate approach or with multivariate features of different categories. To achieve best results from the evaluated predictive models, we tested models on every possible combination of feature categories as reported in **Table 6**, assuming the *Traffic* category as always present as input for models.

According to the above presented results, it is interesting to assess the relevance of different features to explain model impact and minimize model costs. To this end, an analysis of the CONV-BI-LSTM for the representative sensor of Cluster-1 has been performed. The analysis calculated the MAPEs using all the features except the specific considered category: for example, the MAPE of the CONV-BI-LSTM for 1h prediction target of the TrafficFlow, considering all the

TABLE 6. The MAPE estimated for 64 combinations of features for all the identified techniques as the median value on the sensors in the 3 clusters described above. The order is based on the combination of features. In bold, best results/configurations. In bold with citation: results obtained taking into account solutions from the state of the art. Please note that CONV-BI-LSTM overcomes all of them in the same feature conditions.

ID	Features adopted in the model						Median value of MAPE for prediction results by technique								min
	Date time	Traf plus	Temp oral	Season ality	Airpoll	weath er	RF	XGBOOST	DNN	LSTM	BI-LSTM	Autoencoder BI-LSTM	Attention CONV-LSTM	CONV-BI-LSTM	
C1	Y	Y	Y	Y	Y	Y	29.342	34.552	42.754	49.407	34.865	34,708	37,059	31.365	29.342
C2	Y	Y	Y	Y	Y	N	29.682	35.545	43.400	49.832	35.870	35,707	39,506	35.613	29.682
C3	Y	Y	Y	Y	N	Y	28.782	34.441	35.465	36.824	31.555	32,998	33,179	30.894	28.782
C4	Y	Y	Y	Y	N	N	30.935	35.373	38.942	35.383	30.564	32,969	35,713	32.485	30.564
C5	Y	Y	Y	N	Y	Y	29.776	34.469	33.425	42.301	39.865	37,167	35,161	36.897	29.776
C6	**Y**	**Y**	**Y**	**N**	**Y**	**N**	**29.598**	**35.547**	**33.865**	**36.792**	**35.097**	35,322	29,923	**25.981**	**25.981**
C7	Y	Y	Y	N	N	Y	29.421	33.711	31.377	34.736	40.510	37,110	30,741	30.106	29.421
C8	Y	Y	Y	N	N	N	31.245	34.414	32.026	37.823	40.662	37,538	31,263	30.500	30.500
C9	Y	Y	N	Y	Y	Y	29.626	36.919	42.187	**37.068 [38]**	34.297	35,608	36,651	**31.115**	29.626
C10	Y	Y	N	Y	Y	N	29.964	35.802	47.201	41.334	34.743	35,272	40,658	34.116	29.964
C11	Y	Y	N	Y	N	Y	29.785	35.976	45.451	44.756	41.620	38,798	37,345	29.240	29.240
C12	Y	Y	N	Y	N	N	31.262	35.792	36.040	37.228	32.727	34,259	32,701	29.363	29.363
C13	Y	Y	N	N	Y	Y	29.431	35.935	34.448	35.829	34.619	35,277	32,287	30.126	29.431
C14	Y	Y	N	N	Y	N	29.764	36.374	36.203	43.510	35.744	36,059	33,015	29.827	29.764
C15	Y	Y	N	N	N	Y	29.972	35.423	31.526	46.201	37.209	36,316	32,919	34.313	29.972
C16	Y	Y	N	N	N	N	**30.960 [14]**	34.235	30.338	**37.068 [23]**	**38.082 [39]**	**34,235[45]**	**29,455[46]**	**28.573**	28.573
C17	Y	N	Y	Y	Y	Y	29.281	34.503	72.909	64.557	48.685	41,594	51,026	29.144	29.144
C18	Y	N	Y	Y	Y	N	30.184	35.350	59.458	68.127	46.874	41,112	44,810	30.163	30.163
C19	Y	N	Y	Y	N	Y	28.711	34.316	45.679	46.211	33.404	33,86	37,125	28.571	28.571
C20	Y	N	Y	Y	N	N	31.211	34.784	51.603	45.188	48.643	41,713	40,862	30.122	30.122
C21	Y	N	Y	N	Y	Y	30.689	35.774	36.428	48.608	40.092	37,933	34,801	33.175	30.689
C22	Y	N	Y	N	Y	N	30.505	36.165	37.337	61.168	34.420	35,292	34,385	31.434	30.505
C23	Y	N	Y	N	N	Y	30.036	34.779	37.583	64.341	51.063	42,921	33,455	29.328	29.328
C24	Y	N	Y	N	N	N	32.629	34.312	36.849	53.854	41.912	38,112	33,257	29.665	29.665
C25	Y	N	N	Y	Y	Y	28.766	35.906	71.829	65.565	54.403	45,154	52,023	32.218	28.766
C26	Y	N	N	Y	Y	N	30.008	37.317	67.870	49.366	46.880	42,098	53,256	38.642	30.008
C27	Y	N	N	Y	N	Y	28.986	35.218	57.938	50.333	59.419	47,318	43,298	28.658	28.658
C28	**Y**	**N**	**N**	**Y**	**N**	**N**	**31.068**	**35.878**	**66.634**	**50.957**	**55.096**	45,487	47,097	**27.561**	**27.561**
C29	Y	N	N	N	Y	Y	29.301	37.532	38.325	40.677	50.303	43,917	35,554	32.784	29.301
C30	Y	N	N	N	Y	N	29.323	37.284	37.149	48.801	55.064	46,174	34,721	32.294	29.323
C31	Y	N	N	N	N	Y	29.964	36.331	34.638	56.157	45.016	40,673	35,293	35.949	29.964
C32	**Y**	**N**	**N**	**N**	**N**	**N**	**29.281**	**34.574**	**33.028**	**57.961**	**44.977**	39,775	29,320	**25.612**	**25.612**
C33	N	Y	Y	Y	Y	Y	61.579	71.245	77.572	82.634	49.253	60,249	62,308	47.044	47.044
C34	N	Y	Y	Y	Y	N	63.153	71.786	82.539	40.695	47.843	59,814	58,809	35.080	35.080
C35	N	Y	Y	Y	N	Y	61.337	67.098	74.538	48.810	39.500	53,299	54,460	34.383	34.383
C36	N	Y	Y	Y	N	N	59.836	69.497	73.069	51.573	43.162	56,329	51,884	30.699	30.699
C37	N	Y	Y	N	Y	Y	62.579	70.836	78.562	52.011	44.275	57,555	58,888	39.215	39.215
C38	N	Y	Y	N	Y	N	65.235	74.594	71.483	47.473	47.431	61,012	53,134	34.786	34.786
C39	N	Y	Y	N	N	Y	60.648	67.809	65.560	50.268	45.742	56,775	49,816	34.072	34.072
C40	N	Y	Y	N	N	N	65.146	68.154	62.328	52.430	38.560	53,357	46,883	31.439	31.439
C41	N	Y	N	Y	Y	Y	87.662	87.685	96.116	48.239	55.073	71,379	65,153	34.190	34.190
C42	N	Y	N	Y	Y	N	92.600	91.702	97.397	45.643	38.839	65,270	66,528	35.660	35.660
C43	N	Y	N	Y	N	Y	81.587	87.461	91.794	98.094	74.307	80,884	64,471	37.149	37.149
C44	N	Y	N	Y	N	N	88.379	93.141	97.063	48.135	43.457	68,299	66,708	36.354	36.354
C45	N	Y	N	N	Y	Y	89.694	86.278	93.713	47.987	39.295	62,786	64,921	36.130	36.130
C46	N	Y	N	N	Y	N	95.933	90.567	97.018	43.781	55.774	73,170	65,662	34.307	34.307
C47	N	Y	N	N	N	Y	81.423	82.424	84.088	57.282	44.650	63,537	59,109	34.131	34.131
C48	N	Y	N	N	N	N	105.358	88.863	83.781	57.603	39.214	64,038	56,526	29.271	29.271
C49	N	N	Y	Y	Y	Y	66.155	70.646	99.553	71.170	55.400	63,023	73,459	47.366	47.366
C50	N	N	Y	Y	Y	N	70.520	75.833	92.552	64.248	47.862	61,847	63,637	34.723	34.723
C51	N	N	Y	Y	N	Y	68.477	65.479	84.188	74.767	57.531	61,505	60,418	36.648	36.648
C52	N	N	Y	Y	N	N	75.340	68.767	77.498	68.524	49.812	59,289	53,901	30.304	30.304
C53	N	N	Y	N	Y	Y	63.759	69.235	79.907	65.290	54.281	61,758	63,021	46.135	46.135
C54	N	N	Y	N	Y	N	69.726	69.803	71.915	46.227	54.910	62,356	53,797	35.679	35.679
C55	N	N	Y	N	N	Y	64.300	64.550	67.123	56.830	83.575	74,062	49,193	31.264	31.264
C56	N	N	Y	N	N	N	75.239	67.306	69.582	57.716	49.640	58,473	52,173	34.764	34.764
C57	N	N	N	Y	Y	Y	98.969	91.366	112.23	66.553	80.339	85,852	74,096	35.963	35.963
C58	N	N	N	Y	Y	N	101.122	96.404	99.889	60.807	84.501	90,452	66,802	33.715	33.715
C59	N	N	N	Y	N	Y	86.663	88.952	85.747	66.840	94.029	91,490	63,157	40.567	40.567
C60	N	N	N	Y	N	N	86.249	89.043	95.480	59.384	57.615	73,329	63,936	32.392	32.392
C61	N	N	N	N	Y	Y	97.382	91.550	102.56	45.337	43.743	67,646	71,048	39.538	39.538
C62	N	N	N	N	Y	N	98.401	91.546	99.451	60.519	54.513	73,029	70,780	42.109	42.109
C63	N	N	N	N	N	Y	83.403	81.393	84.196	51.608	51.096	66,244	62,604	41.012	41.012
C64	N	N	N	N	N	N	88.430	87.844	85.450	56.906	44.187	66,015	60,793	36.137	36.137

TABLE 7. CONV-BI-LSTM results for the representative sensor of the clusters.

Representative sensor	MAE	MAPE	RMSE	R2	MASE
Cluster-1	161,42	15,35	221,84	0.95	0.51
Cluster-2	138,98	23,86	182,48	0.90	0.60
Cluster-3	81,86	25,73	124,82	0.89	0.57

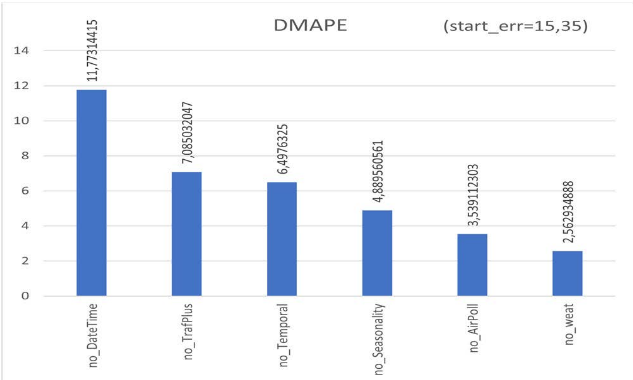


FIGURE 6. Bar plot of the feature categories importance.

features categories except the *DateTime,* was 27,12%. After calculating all MAPEs obtained excluding recursively each single feature category, the DMAPE has been calculated. The DMAPE is defined as the difference of MAPE with respect to the minimum MAPE registered for the CONV-BI-LSTM such as:

$$DMAPE_i = MAPE_{all-cat_i} - minMAPE$$

where: i = 1, ..., number of categories-1 (all, except the traffic, for a total of 6).

Categories with a higher DMAPE are the most relevant ones, since they do not cause larger differences / errors. Results are reported in the bar plot of **Figure** 6. The feature category with the highest DMAPE is the *DateTime* followed by the *Trafplus,* and the *Temporal* feature category. Additional information on data seasonality for short-term prediction has been ranked 4th, ahead of Air Pollution feature category which in turn beats also Weather features. On the basis of **Table** 6 Min values obtained for all possible combinations of features, the usage of weather feature marginally influences the MAPE, and in most cases the resulting MAPE is better without taking into account weather conditions. This fact can be different for different methods.

B. IMPACT OF DATA MISSING ON PRECISION

Data missing is an inevitable problem when dealing with real-world IoT sensor networks and of course, the traffic data from the real traffic system scenario of this study are affected by this problem. Traffic sensors may suffer of problems such as detector malfunction and communication failure, while there could be also some problems during the data acquisition process. All these problems can affect the monitoring of traffic and may constrain the predictive capability of the predictive models at runtime. The approaches of data imputation for

TABLE 8. Data missing analysis based on different missing rates on the clusters representative sensors of Table 7.

Representative sensor	Missing Rate	MAE	MAPE	RMSE	R2
cluster 1 METRO775	0%	161.42	15.35	221.84	0.95
	10%	173.19	16.12	241.86	0.94
	25%	177.36	17.17	258.88	0.93
	50%	176.98	16.77	258.26	0.93
	75%	173.92	16.67	248.51	0.93
cluster 2 METRO707	0%	138.98	23.86	182.48	0.90
	10%	147.49	25.36	194.64	0.88
	25%	146.77	24.90	193.56	0.88
	50%	145.72	24.52	193.34	0.88
	75%	146.10	24.58	193.46	0.88
cluster 3 METRO714	0%	81.86	25.73	117.37	0.89
	10%	83.73	27.99	119.32	0.87
	25%	83.01	27.15	119.11	0.87
	50%	85.00	28.92	122.33	0.87
	75%	82.18	26.89	118.42	0.88

producing surrogate data may help in creating dense data in training and execution [63], while actual data are preferrable. Therefore, in training, we overcome the occurrence of missing data cases by considering only complete samples/sequences according to the architecture.

On the other hand, the presence of missing data samples in making predictions (execution of the predictive model) may impact on the precision, up to make impossible to produce the prediction. In the literature, several approaches have been proposed depending on the machine or deep learning architecture adopted [63], [64].

In order to assess the impact of missing data on prediction a set of experiments have been conducted on the best solution identified in **Table 6** for the CONV-BI-LSTM using the best dataset configuration (*Traffic* and *DateTime*) as reported in **Section IV,** on the test dataset from 10/02/2020 to 16/02/2020 randomly setting to missing the *Traffic Flow* of a percentage of the total dataset based on the missing rates chosen (10%, 25%, 50%, 75%) and then imputing the missing data. The approach used for imputation has been based on the so-called Hot Deck on the basis of which the missing data are imputed with a previously observed data value from a "similar" unit [64]. Thus, thanks to data seasonality (daily and weekly) an identical time slot of a previous period or the median of a set of identically positioned slots in time, have been taken. The results are reported in **Table 8**.

The imputation strategy proposed to handle missing data reports valid results for the missing rates of 10%, 25%, 50%, 75% on all the representative sensors of the three clusters. In particular, taking into consideration the MAE as evaluation metric, the strategy proposed for the METRO775 achieved a minimum additional MAE, to the 0% missing rate test set MAE of 161.42, of 11.77 and a maximum additional MAE of 15.94 for the missing rates considered. The results on the METRO707 are even better in terms of MAE starting from the 0% missing rate of the test set of 138.98 ranging from

TABLE 9. Comparative performance for machine learning techniques in the context of traffic flow predictions.

	Training execution		Prediction
Processing time	Duration (s)	Max GPU	execution (s)
RF	14.681	On CPU	0.023
XGBOOST	4.352	On CPU	0.002
DNN	748.431	25%	0.056
LSTM	527.623	40%	0.017
BI-LSTM	681.874	42%	0.021
Autoencoder BI-LSTM	3240.564	38%	0.033
Attention-based CONV-LSTM	2579.248	41%	0.023
CONV-BI-LSTM	**353.672**	**39%**	**0.102**

a minimum additional value of 6.74 to a maximum of 8.51. The METRO714 handled the different missing rates with an additional MAE ranging from a minimum of 0.32 to a maximum of 3.14.

As a conclusion, the presence of missing data in execution of the prediction marginally impacted the quality of results for CONV-BI-LSTM. This result is very similar to what has been obtained in [63] for similar data sets.

C. PERFORMANCE ANALYSIS

To assess the performance analysis, we have evaluated the required computational capability and the execution time obtained by the RF, XGBOOST, DNN, LSTM, BI-LSTM, Autoencoder BI-LSTM, attention-based CONV-LSTM, and CONV-BI-LSTM, which provided the best MAPE, see **Table** 6. The performance analysis is made of two parts as reported in **Table** 9 with the training described in **Section IV**. The first one is based on typical train execution time required for a single sensor/cluster considering the above-described temporal window and considering all the features of **Table** 2. The second one analyses the execution time required to compute predictions. Please note that, the ensemble learning techniques RF and XGBoost have been trained on CPU, while deep learning techniques have been trained on the GPU. The CPU computations have been performed on 8 core XEON at 2.3 GHz, while deep learning solutions have been executed on GPU as NVIDIA Quadro GV100 with 32GByte Ram, which has 5120 CUDA Cores, FP64 perf as 7.4 TFLOPS.

The training time required by the ensemble learning techniques is much lower than the ones of deep learning techniques, even though for the training of the latter GPU has been used. The time needed for training the deep neural networks tested depends on multiple factors. The time reported included the hyperparameter optimization as described in Section IV. This explains why Deep Learning based solutions require more training time, and also other factors related to the optimization process, could influence this: for instance, the time required for the networks to converge on each iteration of the optimization process. Moreover, the use of Random Search optimization method implies the random selection of hyperparameters values from the hyperparameters search space, and this can lead to different total training times for different optimized models. On this regard, Autoencoder BI-LSTM and Attention-based CONV-LSTM have been trained by using a different implementation of the optimization search space, with respect to the CONV-BI-LSTM. As described in Section IV, for the CONV-BI-LSTM the only parameter optimized on reported training time has been the learning rate, see Section IV. In fact, since the CONV-BI-LSTM provided the best MAPE, the value reported for training time refers to the final training values obtained during manual optimization as described in Section IV. Regarding GPU load, it settles at a max of 25% while training DNNs, up to a max load of around 40% for the other architectures. As a final consideration, the training time can further be improved by: (i) reducing the ranges of the hyperparameters or passing to a manual optimization, (ii) executing the training once per week/month, and/or (iii) performing the training per cluster.

According to **Table 8,** the required time to compute predictions on a test set, XGBOOST is the fastest solution, and it takes a few seconds, followed by the RF, DNN, LSTM, BI-LSTM requiring hundredths of seconds, and Attention-based CONV-LSTM, Autoencoder BI-LSTM which are much slower. The solution taking longer time is the CONV-BI-LSTM, performing predictions in 1/10 of second, while, as presented above it is an acceptable performance, and it is the best in terms of MAPE. Thus, in most cases, in 10 minutes it is possible to compute at least 580 estimations, remarking the suitability for short terms predictions. Please note that in mid-size cites, as in Florence, there are 150 sensors, which results in a cheap solution.

VI. CONCLUSION

In this paper, we have proposed a predictive approach and solution for short-term predictions of traffic flow data in a urban context, being typically much more noisy than high speed roads segments, which appears to be the main target of most solutions in current state of the art. Accessing precise traffic flow data is mandatory to guarantee high level of services such as: traffic flow reconstruction, which in turn is used to perform what-if analysis, conditioned routing, etc. They have to be reliable and precise for possible rescue teams and fire brigades.

The considered case refers to city traffic data being noisier and more complex than more regular traffic data from high-speed roads. This paper proposes a solution and an approach for short-term traffic flow prediction by using traditional machine learning as RF and XGBOOST and comparing them with deep learning techniques as DNN, LSTM, BI-LSTM, Attention based CONV-LSTM, Autoencoder BI-LSTM, and the proposed CONV-BI-LSTM. In the paper a comparative analysis has been performed, taking into account a large number of solutions and features, thus analysing the precision of such different techniques. Best solution turned out to be the solution, namely CONV-BI-LSTM which in most cases produced better results with respect to other solutions already in the state of the art and even better results could be obtained with the proposed feature combination. The current study evaluated which type of feature categories, based on the related state-of-the-art works, improved the models' results. On this regards a specific analysis on the motivation and feature relevance has been discussed, considering a very large number of combinations of both features and methods. Also, RF could produce quite precise results and is computationally lighter than CONV-BI-LSTM. This solution has been developed in the context of Sii-Mobility/Km4City smart city mobility and transport national project and it is in use in other solutions such as Snap4City, TRAFAIR CEF, and REPLICATE smart city control room for Florence area.

ACKNOWLEDGMENT

The authors would like to thank the MIUR, the University of Florence, and companies involved for co-funding Sii-Mobility national project on smart city mobility and transport. Km4City is an open technology and research of the DISIT Laboratory. Sii-Mobility is grounded and has contributed to Km4City open solution. The present solution is also adopted in Snap4City platform and Trafair CEF project.

REFERENCES

[1] L. Po, F. Rollo, J. R. R. Viqueira, R. T. Lado, A. Bigi, J. C. Lopez, M. Paolucci, and P. Nesi, "TRAFAIR: Understanding traffic flow to improve air quality," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Oct. 2019, pp. 730–737.

[2] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, "Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs," *Sensors*, vol. 19, no. 9, p. 2206, May 2019.

[3] E. Alomari, I. Katib, A. Albeshri, T. Yigitcanlar, and R. Mehmood, "Iktishaf+: A big data tool with automatic labeling for road traffic social sensing and event detection using distributed machine learning," *Sensors*, vol. 21, no. 9, p. 2993, Apr. 2021.

[4] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "SmartRoad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Trans. Sensor Netw.*, vol. 11, no. 4, pp. 1–27, Dec. 2015.

[5] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0119044.

[6] B. Alsolami, R. Mehmood, and A. Albeshri, "Hybrid statistical and machine learning methods for road traffic prediction: A review and tutorial," *Smart Infrastructure and Applications*. Cham, Switzerland: Springer, 2020, pp. 115–133.

[7] P. Bellini, D. Cenni, P. Nesi, and M. Soderi, "Anomaly detection on IoT data for smart city," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Sep. 2020, pp. 416–421.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[9] S. Bilotta and P. Nesi, "Traffic flow reconstruction by solving indeterminacy on traffic distribution at junctions," *Future Gener. Comput. Syst.*, vol. 114, pp. 649–660, Jan. 2021.

[10] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[11] Y. Kamarianakis and P. Prastacos, "Space-time modelling of traffic flow," *Comput. Geosci.*, vol. 31, no. 2, pp. 119–133, 2005.

[12] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, no. 2, pp. 114–121, Feb. 2006.

[13] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[14] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[15] C. H. E. N. G. Zheng and C. H. E. N. Xian-Fu, "The model of short-term traffic flow prediction based on the random forest," *Microcomput. Appl.*, vol. 35, no. 10, pp. 46–49, 2016.

[16] O. Mohammed and J. Kianfar, "A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in Missouri," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2018, pp. 1–7.

[17] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[18] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10. 1995, p. 1995.

[19] W. Ding, X. Wang, and Z. Zhao, "CO-STAR: A collaborative prediction service for short-term trends on continuous spatio-temporal data," *Future Gener. Comput. Syst.*, vol. 102, pp. 481–493, Jan. 2020.

[20] S. U. Khan et al., "Deep-ReID: Deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance," *Multimedia Tools Appl.*, pp. 1–22, 2021, doi: 10.1007/s11042-020-10145-8.

[21] S. U. Khan, I. U. Haq, N. Khan, K. Muhammad, M. Hijji, and S. W. Baik, "Learning to rank: An intelligent system for person reidentification," *Int. J. Intell. Syst.*, vol. 37, no. 9, pp. 5924–5948, Sep. 2022.

[22] P. Nesi, C. Badii, P. Bellini, D. Cenni, G. Martelli, and M. Paolucci, "Km4City smart city API: An integrated support for mobility services," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 18–20.

[23] C. Badii, E. G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, M. Soderi, and I. Zaza, "Snap4City: A scalable IoT/IOE platform for developing smart city applications," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 2109–2116.

[24] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City (SmartCity)*, Dec. 2015, pp. 153–158, doi: 10.1109/SmartCity.2015.63.

[25] E. Fama, "Random walks in stock market prices," *Financial Analysts J.*, vol. 21, no. 5, pp. 55–59, 1965. [Online]. Available: http://www.jstor.org/stable/4469865

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Oct. 1995.

[27] S. Haykin and R. Lippmann, "Neural networks. A comprehensive foundation," *Int. J. Neural Syst.*, vol. 5, pp. 363–364, May 1994.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[29] D. Kang, Y. Lv, and Y.-Y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6, doi: 10.1109/ITSC.2017.8317872.

[30] Z. Wang, R. Zhu, M. Zheng, X. Jia, R. Wang, and T. Li, "A regularized LSTM network for short-term traffic flow prediction," in *Proc. 6th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Dec. 2019, pp. 100–105, doi: 10.1109/ICISCE48695.2019.00030.

[31] H. Drucker et al., "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9. 1996.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[33] L. Mou, P. Zhao, H. Xie, and Y. Chen, "T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction," *IEEE Access*, vol. 7, pp. 98053–98060, 2019, doi: 10.1109/ACCESS.2019.2929692.

[34] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[35] E. Fix and J. L. Hodges, "Discriminatory analysis. Nonparametric discrimination: Consistency properties," *Int. Stat. Rev.*, vol. 57, no. 3, pp. 238–47, 1989, doi: 10.2307/1403797.

[36] G. E. Box, G. M. Jenkins, and G. C. Reinsel, "Forecasting," *Time Ser. Anal.*, vol. 12, pp. 137–191, Jun. 2008.

[37] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: A GRU-based deep learning approach," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 578–585, Sep. 2018, doi: 10.1049/IET-ITS.2017.0313.

[38] F. M. Awan, R. Minerva, and N. Crespi, "Improving road traffic forecasting using air pollution and atmospheric data: Experiments based on LSTM recurrent neural networks," *Sensors*, vol. 20, no. 13, p. 3749, Jul. 2020, doi: 10.3390/s20133749.

[39] R. L. Abduljabbar, H. Dia, and P.-W. Tsai, "Unidirectional and bidirectional LSTM models for short-term traffic prediction," *J. Adv. Transp.*, vol. 2021, pp. 1–16, Mar. 2021.

[40] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5615–5624, Jun. 2022, doi: 10.1109/TITS.2021.3055258.

[41] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with Conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6, doi: 10.1109/WCSP.2017.8171119.

[42] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017, doi: 10.1016/j.trc.2017.02.024.

[43] Y. Liu, C. Lyu, Y. Zhang, Z. Liu, W. Yu, and X. Qu, "DeepTSP: Deep traffic state prediction model based on large-scale empirical data," *Commun. Transp. Res.*, vol. 1, Dec. 2021, Art. no. 100012, doi: 10.1016/j.commtr.2021.100012.

[44] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5668–5675, doi: 10.1609/aaai.v33i01.33015668.

[45] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from Twitter," *World Wide Web*, vol. 24, no. 4, pp. 1345–1368, Jul. 2021, doi: 10.1007/s11280-020-00800-3.

[46] B. Vijayalakshmi, K. Ramar, N. Jhanjhi, S. Verma, M. Kaliappan, K. Vijayalakshmi, S. Vimal, and U. Ghosh, "An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city," *Int. J. Commun. Syst.*, vol. 34, no. 3, Feb. 2021, Art. no. e4609, doi: 10.1002/dac.4609.

[47] J. Mena-Oreja and J. Gozalvez, "A comprehensive evaluation of deep learning-based techniques for traffic prediction," *IEEE Access*, vol. 8, pp. 91188–91212, 2020, doi: 10.1109/ACCESS.2020.2994415.

[48] N. Kumar and M. Raubal, "Applications of deep learning in congestion detection, prediction and alleviation: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 133, Dec. 2021, Art. no. 103432, doi: 10.1016/j.trc.2021.103432.

[49] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022.

[50] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Netw.*, vol. 32, no. 4, pp. 40–46, Jul. 2018, doi: 10.1109/MNET.2018.1700411.

[51] A. Banerjee and R. N. Dave, "Validating clusters using the Hopkins statistic," in *Proc. Int. Conf. Fuzzy Syst.*, vol. 1, Jul. 2004, pp. 149–153.

[52] R. Xu and D. C. Wunsch, *Clustering*. Hoboken, NJ, USA: Wiley, 2008.

[53] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, 2001.

[54] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[56] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[57] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, Oct./Dec. 2006.

[58] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585, doi: 10.1109/IJCNN.2017.7966039.

[59] H. Xie, L. Zhang, and C. P. Lim, "Evolving CNN-LSTM models for time series prediction using enhanced grey wolf optimizer," *IEEE Access*, vol. 8, pp. 161519–161541, 2020, doi: 10.1109/ACCESS.2020.3021527.

[60] M. Cao, V. O. K. Li, and V. W. S. Chan, "A CNN-LSTM model for traffic speed prediction," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9129440.

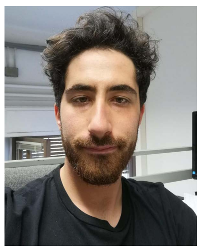
[61] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2012.

[63] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. C, Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102674.

[64] R. R. Andridge and R. J. Little, "A review of hot deck imputation for survey non-response," *Int. Stat. Rev.*, vol. 78, no. 1, pp. 40–64, 2010, doi: 10.1111/j.1751-5823.2010.00103.

STEFANO BILOTTA was born in Arezzo, Italy, in 1983. He received the master's degree in mathematics from the University of Siena, Italy, in 2009, and the Ph.D. degree in computer engineering and automation from the University of Florence, Italy, in 2013. From 2013 to 2016, he was a Postdoctoral Researcher at the University of Florence, and then, at the DISIT Laboratory, DINFO Department. His research interests include traffic flow reconstruction algorithms, parallel solution, dynamic systems, machine learning, languages, and coding theory. He has been involved in projects, such as Sii-Mobility and Trafair.



ENRICO COLLINI is currently pursuing the Ph.D. degree with the DINFO Department, University of Florence. He is also an Engineer. His research interests include deep learning, mobility, and data models.

PAOLO NESI (Member, IEEE) is currently a Full Professor at the DINFO Department, University of Florence, and the Chief of the DISIT Laboratory. His research interests include machine learning, massive parallel and distributed systems, physical models, the IoT, mobility, big data analytic, semantic computing, formal model, and machine learning. He has been the chair of several international conferences. He is and has been the coordinator of several research and development multipartner international research and development projects.

GIANNI PANTALEO is currently an Aggregated Professor at the DINFO Department, University of Florence. His research interests include knowledge engineering, the IoT, visual analytics, mobility, NLP, and semantic computing. He has been the coordinator of a number of WP in international research and development projects.

Open Access funding provided by 'Consiglio Nazionale delle Ricerche-CARI-CARE' within the CRUI CARE Agreement