High Precision Traffic Flow Reconstruction via Hybrid Method

Stefano Bilotta, Valerio Bonsignori, and Paolo Nesi, *Member, IEEE*

***Abstract*— Traffic management and sustainable mobility are the central topics for intelligent transportation systems (ITS). By means of modern technologies, it is possible to collect real-time traffic flow data to extract useful information to monitor and control vehicular traffic. On the other hand, costs to obtain this piece of information are high. It requires either direct measures in the network road by installing large number of sensors (more precise data) or acquiring data from international providers supplying data coming from onboard units, mobile app, navigators, etc. In current paper, this problem has been addressed providing a solution granting traffic flow data in each road segment of the whole network by reconstructing the computation by means of data from few scattered traffic sensors in fixed positions of the road network. The proposed approach combines the solution of nonlinear Partial Differential Equations (PDEs) with machine learning for improving the state-of-the-art solutions of PDE. The result has been a higher precision with respect to PDE-based solutions, and a strongly reduced execution time. Several different machine learning models have been compared for such a purpose, demonstrating the general viability of the hybrid architecture proposed. The research result has been obtained in the framework of both the Sii-Mobility national project on transport systems, and MOST, the National Center on Sustainable mobility (both funded by the Italian Ministry of Research), by exploiting the Snap4City platform.**

***Index Terms*— Traffic flow reconstruction, traffic flows, machine learning, hybrid architectures, machine learning PDE solution.**

I. Introduction

TRAFFIC flow computation consists in obtaining real time traffic flow state in each segment of a road network in a urban or rural area. Such a computation is fundamental for implementing a large number of smart services such as: dynamic route guidance, road digital signage, congestion detection, traffic reduction; fuel consumption and pollution emission monitoring, etc. [1], [2]. Often, traffic flow estimation is related to a monitored area based on few fixed points/sensors and thus no information is provided in other connected road segments free of sensors. Many contributions focus on this field of research as in [3], [4], [5], [6], [7], [8], and [9]. The usage of large number of traffic flow sensors can help in getting more precise estimations in the whole city (road network), but costs may become unaffordable. Traffic density measures are typically obtained by stationary sensors on fixed positions and they are usually of different kinds: TV cameras, road spires, etc. [10], producing measures in terms of traffic flow density, velocity and number of vehicles. Due to sustainability reasons, the number of deployed sensors has to be limited. Thus, it is mandatory to adopt some reconstruction algorithms to obtain the traffic flow condition in each road segment of the city in order to have dense traffic flows in the unmeasured road segments.

Surrogated traffic flow data can be obtained from: Mobile Apps, on board units (insurance black boxes for instance), social media app [11], and recently also from vehicle networks [12]. In [13], a deep Restricted Boltzmann Machine and Recurrent Neural Network, RNN, architecture has been used to predict traffic congestion evolution based on GPS data from taxis, and thus on their position and velocity, etc. In [14], a smartphone-based crowd sensing system for traffic detection and measure has been proposed, where data are gathered from the handheld devices. Data coming from navigator Apps (e.g., TomTom, Google map, Waze), at long term, could be very expensive for a municipality with respect to the installation of sensors. Those measures are not related to the actual counting of vehicles, since they are based on measuring single vehicle velocity, which does not directly relate to road traffic density. Vehicular Ad-Hoc Networks, VANET, are modeling communication among vehicles, thus creating a shared network of information which could be used to understand local traffic [12], [15]. On the contrary, the usage of TV Cameras located in specific critical points allows to perform direct measures, which reduces costs, while increasing precision in specific points. Then, multiple areas/lanes can be controlled with a single installation, so as to enable the control of a high number of traffic flows. Traffic flow sensors provide continuous measuring of traffic on selected roads at fine grain, and in most cases, they also provide information about the kinds of vehicles: busses, tracks, cars, bikes, etc. Generally speaking, to setup a network of traffic flow sensors in a city drastically avoids the costs of taking updated data from third parties such as Google or Navigator mobile Apps, which provide statistical data, instead of specific and direct measures.

A. Related Work

In the context of traffic flow theory, a distinction has to be done between traffic flow ***Predictions*** in specific points in urban contexts or highways (short or long terms in the

Manuscript received 19 December 2022; revised 26 July 2023 and 16 September 2023; accepted 30 October 2023. This work was supported by the CN MOST, National Center on Sustainable Mobility. The Associate Editor for this article was S. Ahn. *(Corresponding author: Paolo Nesi.)*

The authors are with the DISIT Laboratory, DINFO Department, University of Florence, 50139 Florence, Italy (e-mail: stefano.bilotta@unifi.it; valerio.bonsignori@phd.unipi.it; paolo.nesi@unifi.it).

Digital Object Identifier 10.1109/TITS.2023.3329544

future [9]) and traffic flow ***Reconstruction*** at small and large scale (few roads, highways' segments or whole city networks).

In the context of ***traffic flow predictions***, a large number of methods [7], [9], [16] use data-driven Machine/Deep Learning approaches for predicting traffic flow data or congestion levels in locations where those values are measured via sensors. The traffic flow prediction is performed by using a stacked autoencoder (SAE) model in [7], the author uses the powerfulness latent representation to build and infer the next flow traffic estimation in specific points. In [9], authors compared a large number of machine and deep learning solutions for traffic flow prediction computed on the basis of different data sources. In [16], authors combine convolution and LSTM (Long Short-Term Memory) to form a Conv-LSTM module which can extract the spatial-temporal information out of the traffic flow information. Additionally, they adopt a Bi-directional LSTM to analyze historical traffic data information and get traffic flow periodicity features. In [17], authors use a representation of the road network with a graph embedding: the encoded information is applied to a generative adversarial network to obtain road traffic state information in real-time.

The ***Traffic Flow Reconstruction***, *TFR,* is the process to estimate dense traffic density (flow) – e.g., vehicle per meter (or vehicles per second) – for each road segment within the road network by starting from a limited number of traffic flow sensors or data providing traffic density (flow) in the roads, or velocity in some cases and at the same time instant. It can be regarded as an extrapolation approach passing, for example, from 100 sensors data to 10.000 traffic flow data of road segments. TFR approaches can be classified into three main categories: *model-driven*, *data-driven* and *mixed* approaches.

TFR Model-driven approaches are those taking into account the physical model of traffic in the spatio-temporal domain, such as both agent-based and those solving differential equations. *Agent-based* solutions for the traffic flow reconstruction are substantially simulators which compute traffic flow by modeling vehicles as agents, thus showing typical problems of scalability for large road networks [18], [19], [20]. For example, InterSCSimulator [19] is an agent-based solution which may scale up to relevant networks at the expense of memory and computational time. Large scale simulators are often based on origin destination data (O-D) and population characteristics [21]. They focus on basic concepts and methods of discrete choice analysis. They describe the application of this methodology to travel demand modelling. Discrete choice models use the principle of utility and benefit maximization: operational models often consist in the characterization of parameterized utility functions via statistical inference [22]. Discrete choice models are usually applied to forecast trips by starting from origins-destinations data and considering different transport modalities [23]. Other simulators have been reviewed and compared in [24], identifying limitations when it comes to both traffic flow evolution and addressing large scale cases or small events. DEUS [15] is a Discrete-Event Universal Simulator used to simulate a Vehicular Ad-Hoc Network. VANET [12] has been used with SUMO (Simulation of Urban Mobility, http://sumo.dlr.de) to create microsimulations of traffic crossroad distribution. In those cases, the indeterminacy of vehicle behavior at junction is performed by using data coming from O-D or by making samples at the crossroads.

According to a different approach with respect to the above described Agent Based, a traffic flow can be modeled as a fluid moving into the road network, and thus the TFR problem can be regarded as the classical solution of the LWR (Lighthill-Whitham-Richards) model [25], [26], which considers traffic density in terms of nonlinear Partial Differential Equations, PDEs, and it is used to estimate traffic flow using scattered observations, location of sensors and so forth. In this context, the estimation of traffic distribution at junctions plays a crucial role on the effectiveness of the LWR model application in real contexts and its related solution is not trivial for large networks, so called macroscale [27], [28], [29], [30], [31]. Moreover, traffic distribution at junctions may change over time during day and week, and thus, its computational costs may be very high.

On this line, a scalable traffic flow reconstruction approach at macroscale has been proposed and applied in real-world contexts of (large) city road networks [32]. Such an approach is based on LWR model where the indeterminacy of traffic distribution at junctions has been solved by means of a stochastic relaxation technique which reduced system errors at the expense of computational cost, while resulting more scalable and effective with respect to agent-based solutions. Limitations of this approach are related to the precision of the estimation and on the execution time that could be improved.

TFR Data-driven approaches should derive traffic state by means of the dependences learned from observed data using statistical or machine learning methods. They should rely on real time and historical data in each segment to extrapolate data in each and every segment. This means that it should not require a priori knowledge of traffic models and laws, as it occurs with model driven solutions. Machine and deep learning solutions can provide predictive capabilities for nonlinear phenomena *as long as historical data about dense traffic flow are available*. Data driven approaches have been also used for traffic flow analysis. For example, in [33], authors have proposed machine learning tasks to analyze road networks to perform vehicle speed limit classification. Thus, in current literature, there are many data-driven approaches without a specific address of the traffic reconstruction over the entire road traffic network [34].

In order to overcome such limitations belonging to the above-described TFR Data Driven solutions, some ***TFR Hybrid*** Approaches have been proposed in literature, as well as in present paper. TFR Hybrid approaches combine model-driven and data-driven methods to achieve more accurate and efficient results for TFR computing. In [35], authors have investigated the use of a model-based neural network for traffic prediction problems, using noisy measurements coming from Probe Vehicles. Designing a single optimization model, they developed a solution using a deep neural network to reduce both identification process and other processes like reconstruction, prediction, and noise rejection. The *physics-informed deep learning* (PIDL) framework has been proposed for solving PDEs and recently it has been applied to various physical models [36]. In the context of TFR, PIDL can

describe the LWR model and it has been only applied to simple road networks, like single road or road ring [37], [38], [39]. In such studies, no road crossing modeling has been considered to address the indeterminacy at junctions, which would involve the solution of the so-called Riemann's problem.

Basically, when it comes to large road networks having a limited number of traffic measurements (traffic data sensors), the usage of data-driven or hybrid approaches is considered prohibitive in terms of accuracy to capture the vehicular traffic state of the whole network by assuming as input data the observed data only, without any additional information.

Moreover, non-toy solutions are affected by data *discontinuities* on observed sensor data in real-time, caused by malfunctions of sensors and/or communication, thus reducing the accuracy of the whole network data and TFR.

B. Article Aim and Structure

In this paper, two TFR Hybrid approaches have been provided combining models based on machine learning approaches and PDE solution, so as to solve indeterminacy at junctions. The indeterminacy is due to the fact that measuring the traffic flow arriving at a crossroad without measuring the traffic produced in the output roads, the corresponding out flows can be locally undetermined. On the other hand, globally the distribution of flows on the outputs can estimated by exploiting the knowledge of flow in the other parts of the road network.

The proposed solutions aimed at producing accurate TFR of large road network, main problems are related to: (i) density of traffic flow estimations on large graph roads from scattered data sensors (agent based solutions are unsuitable and do not address the indeterminacy at junctions, PDE based model driven solutions are computationally expensive and may provide dense reconstructions in small scale, data driven solutions at large scale are not available in the literature), (ii) indeterminacy of the traffic partition at junctions (difficult to be solved by any approach, see model driven stochastic relaxation approach of [32]), (iii) high computational complexity and thus complexity of execution time (demanding a new estimation of TFR at each new sample of the sensors for the whole network), which can be more easily addressed by machine/deep learning solutions rather than agent based and model based solutions in general, and (iv) producing TFR also in the event of sensor data showing discontinuities, e.g., missing observed data.

In more details, in this paper, a Hybrid TFR is proposed by integrating machine learning with a data driven solution based on PDE for large scale TFR computation in order to: (a) improve the estimation accuracy of the TFR with respect to the results of data driven based on PDE solutions [32], (b) speed up the execution time needed for computing TFR with respect to the performance in order to make the solution more scalable for very large networks. The indeterminacy of the traffic flow distribution at junctions has been solved at level of TFR model. Therefore, the proposed Hybrid approach is based on combining the TFR model based on PDE solution with machine learning approaches by means of two possible innovative hybrid architectures, which are identified in the paper as Case (i) and Case (ii). Data driven models for TFR have been trained on the history of traffic reconstruction data to learn the traffic dynamics behavior in a large network. This approach leads to solve the problem related to traffic distribution at the junctions which is generally very expensive from a computational point of view, via model-driven approach. Then, the proposed hybrid approach reduces the execution time for both the entire TFR process and the whole road network. Moreover, the combination of data dependency obtained with such data-driven approach, together with the understanding of physical model and its related traffic distribution through the PDEs solution, allows to produce more accurate TFR solutions than those attainable from model driven solutions at present state of the art, e.g., [32]. Moreover, data imputation methods have been also considered to solve the problem of missing sensors data, which may cause, as to model-driven solutions, the impossibility of computing the solution when the number of missing sensor data is large in space and time. This has brought a general improvement of the TFR accuracy.

To this end, a range of Machine Learning, ML approaches (Adaboost [40], RF [41], XGboost [42], Bayesian [43], Decision Tree [44], ExtraTree [45], MLP [46]), have been compared to identify the best model to improve the PDE based solutions in terms of TFR. The results presented in this paper have been validated in a context of data related to the actual traffic network of Florence city metro area. This current study and its related outcomes have been produced and validated by exploiting the Snap4City framework for smart city, mobility and transport and data analytics, also using Km4City/Sii-Mobility graph model and tools. The project has been funded by the national Ministry of research [47] and by MOST, Italian National Center on Sustainable Mobility of the national Ministry [48]. Algorithms have been put in execution by exploiting semantic model [31], [49] and the Snap4city Platform [50], [51].

The paper is organized as follows. In Section II, the computation of traffic Flow reconstruction via PDE solution is recalled, together with the approach for result assessment. In Section III, the proposed hybrid architecture and solution to improve precision are described. In Section IV, both context and data used for these experiments are presented. Section V describes in detail the hybrid solution exploiting the machine learning approaches and the first obtained results, Case (i). In Section VI, an improved version of the solution proposed in Section V is presented, by data analysis and exploiting temporal information on data, thus focus is on Case (ii). In a subsection of Section V, outcomes are compared one another and with respect to model driven solutions. Conclusions are drawn in Section VII.

II. Traffic Flow Reconstruction

Before discussing the proposed hybrid solutions, an overview of model driven approaches for TFR based on PDE solution is needed. In the latter, TFR computation is performed by solving a nonlinear equation based on vehicle conservation, which is described by the following scalar

hyperbolic conservation law in a single road segment:

$$\frac{\partial \rho\,(t,x)}{\partial t}+\frac{\partial f\,(\rho\,(t,x))}{\partial x}=0, \tag{1}$$

where, $\rho\,(t,x)$ is the traffic density of vehicles, with values from 0 to ρ_{max}, where $\rho_{max}>0$ is the max traffic density; $f\,(\rho(t,x))$ function is the vehicular flow which is defined by means of the product $\rho\,(t,x)\,v\,(t,x)$, where $v\,(t,x)$ is the vehicle speed; and boundary conditions $\rho\,(t,h)=\rho_h(t)$, $\rho\,(t,k)=\rho_k(t)$, initial values $\rho\,(0,x)=\rho_0(x)$, with $x\in(h,k)$. In the case of first order approximation, we assume that $v\,(t,x)$ is a decreasing function, depending on the density, then the corresponding flow is a concave function. Thus, we consider the local speed of the vehicles as $v\,(\rho)=v_{max}(1-\frac{\rho}{\rho_{max}})$ and then $f\,(\rho)=v_{max}\left(1-\frac{\rho}{\rho_{max}}\right)\rho$, where v_{max} is the speed limit on a given road segment (these assumptions are known in the literature as the *Greenshield's Model*). Equation (1) may be solved by means of an iterative process at finite differences applied to each road segment of the whole network. As proposed in [32], the achievable solution is grounded on a Stochastic Relaxation Approach based on the measures of Traffic Flow data in a limited number of sensor points at each time instant. In this paper, the solution presented in [32] is denoted as SRA4TF.

At each timestamp, SRA4TF solution produces a value of traffic flow density in each road segment of the network, typically of 20 meter, as unit, that is the TFR. The accuracy of SRA4TF solution mainly depends on the stochastic relaxation approach for estimating Traffic Distribution Matrices (TDMs), which are the traffic flow distributions at junctions. A TDMs describe the percentage of vehicles getting out of each outcoming road with respect to those getting in from each incoming road of the junction. Thus, the TDM is defined as $TDM=\{w_{ji}\}_j=n+1,\ldots,n+m, i=1,\ldots,n$ so that $0<w_{ji}<1$ and $\sum_{j=n+1}^{n+m}w_{ji}=1$, for $i=1,\ldots,n$ and $j=n+1,\ldots,n+m$, where w_{ji} is the percentage of vehicles arriving from the i-th incoming road and taking the j-th outcoming road (assuming that, on each junction, the incoming flow coincides with the outcoming flow). The values of w_{ji} depend on the time of the day and of day of the week, etc., on the road size, cross light settings, etc., and thus, it is unknown a priori. The values of w_{ji} are estimated by giving the lower mean error by means of this stochastic relaxation technique as described in [32]. The computing of TFR is progressively performed on a parallel architecture. The estimation of traffic flow density for a city (e.g., in Florence there are more than 30.000 road segments or units) at time instant t would depend on traffic flows at time *t-1* in the whole network, and on the new measures coming from sensors at time t.

Once $TDM(t)$ are estimated (or initially guessed), the SRA4TF solution computes the TFR in the road network and verifies the Root Mean Square Error, RMSE, (or Mean Absolute Error, MAE) with respect to actual values in sensor locations. This is performed by computing the solution excluding data from each different sensor (all of them) by means of a Leave-One-Out Crossing-Validation approach (LOOCV), so as to estimate the deviation from the reconstructed traffic density $\rho^R(t)$, with respect to the observed density by the sensor $\rho^O(t)$, for each time t in T. In the rest of the paper, we refer to R and O to denote reconstructed and observed traffic flow densities (number of vehicles for space unit), respectively. Then, in a road network having m traffic sensors, the LOOCV approach consists in the application of the model to the set of the observed data at time t, that is $\boldsymbol{O}(t)=\{O_1(t),\ldots,O_m(t)\}$, by excluding the k-th observation $O_k(t)$ from $\boldsymbol{O}(t)$, for each $k=1,\ldots,m$. Then, the model is applied to the remaining set of $m-1$ sensor observations and the reconstructed density $R_k(t)$ in the road segment (unit) where the k-th sensor is located, can be estimated and compared with $O_k(t)$ via *RMSE* or *MAE* estimation as follows:

$$RMSE(k)=\sqrt{\frac{\Sigma_{t=1}^{T}\left(R_k\,(t)-O_k\,(t)\right)^2}{T}}, \tag{2}$$

$$MAE(k)=\frac{\Sigma_{t=1}^{T}\left(|R_k(t)-O_k(t)|\right)}{T}. \tag{3}$$

The *RMSE* and *MAE* are used to measure error values when the perfect fit by 0. The unit of measure of *RMSE* and *MAE* is the same of R and O number of vehicles for space unit. Therefore, a value of 0.5 represents a 1/2 of a vehicle in the space of 20 meter. For each round, the stochastic relaxation may produce a new minimum of the *RMSE* that is taken as a reference status, together with the produced new values of the $TDM(t)$, for next iterations. At each timestamp, the *RMSE*(k) for each sensor in the LOOCV is measured and the *RMSE(system) (* average value of the *RMSE* on all the m LOOCV sensors) is considered:

$$RMSE(system)=\frac{1}{m}\sum_{k=1}^{m}RMSE(k). \tag{4}$$

The value of *RMSE* is higher when traffic density is high and thus it reflects the hourly behavior of incoming and outcoming vehicle flow in the city having its maximum in the morning, at about 8 am. On the other hand, the ratio from *RMSE* and traffic flow density is almost constant in daily time and it is in the range of 25% (see for details [32]). The computation of the *MAE(system)* is estimated in similar manner on the basis of Eq. (3).

The computational complexity of SRA4TF depends on the dimension of the road network. As to the metropolitan network of Florence, it includes 1390 nodes (or intersections, junctions), 130 traffic sensors and 31217 road segments (units) of 20 meter. Once $TDMs$ are estimated, then the computational complexity of traffic reconstruction at each time t is an *O(H(V+U))* where: V is the number of nodes, U is the number of road segments and H is the number of iterations (generally H is equals to 250). Since U is much larger than V, then we definitively have a complexity of an *O(HU)*. The stochastic relaxation approach randomly assigns TDM values depending on road featuring. Then, at each attempt, if the local error is lower than the previous one, weights are confirmed. The procedure continues to try new *TDM* until the computed *RMSE*(*system*) is minimized, by means of a sort of Simulated Annealing. The procedure is computationally heavy, and it is typically sporadically performed to update the TDMs. Typically, the solution converges in 600 iterations.

At each iteration, the LOOCV approach via parallel structures considers more than 4 million of road segments/units (of 20 meter).

III. Hybrid Architecture for Improving Precision

As stated in the Introduction, in this paper, we are presenting a hybrid solution able to improve model driven solutions, such as SRA4TF, by using machine learning. More precisely obtaining: *(i) improvement of precision in dense traffic flow estimation, reduction of RMSE(system), (ii) reduction of execution time.* With this aim, a number of ML techniques have been tested as listed above. In [37], [38], and [39], small road segments/networks were studied by using ML approaches exploiting the knowledge of road traffic physical model in the loss function. The proposed hybrid solution overcomes this limitation covering the whole network with a new hybrid architectural solution, which could be also used in solving/improving other PDE solutions. The hybrid solution proposed in this paper consists in using ML together with the exploitation of knowledge about the road network traffic and the SRA4TF solution; it can be regarded as a neuro-symbolic approach.

Indeed, this paper has its focus on two different approaches to tackle problems (improving precision, and performance of TFR estimation from sensors data), which are called ***Case (i)*** (see Section V), and ***Case (ii)*** (see Section VI). Each of them shares the same architecture in terms of data flow for the phases of training and execution of the ML solution (passing from the former to the latter with the trained model and parameter). In *Case (i),* the ML model is trained by taking in input the traffic flow densities observed in the locations of sensors and the corresponding values of TFR estimated by the SRA4TF. *Case (ii)* is improved by adding some features related to temporal information in order to model in terms of feature the seasonality of sensors data and traffic flows. Such temporal information is used to distinguish days from festive, pre-festive and working days and consider the related time slots.

In the following, $\boldsymbol{O}(t)$ means the vector of the observations (measures) from sensors at time t, while $\boldsymbol{R}(t)$ is the vector of the traffic density reconstructed in other segments of the road network at time t. The SRA4TF produces a traffic density for the whole road network which can be regarded as vector $\boldsymbol{R}(t)$ as follows:

$$SRA4TF(\boldsymbol{O}(t-1), \boldsymbol{R}(t-1), \boldsymbol{O}(t), RoadGraph) \rightarrow \boldsymbol{R}(t).$$

Having m traffic sensors in a road network, we obtain that the total road segments (units) in the road network is $m + n$ considering $\boldsymbol{O}(t) = \{O_1(t), \ldots, O_m(t)\}$ and $\boldsymbol{R}(t) = \{R_1(t), \ldots, R_n(t)\}$.

A. Hybrid Architecture, Case (i)

The hybrid architecture for TFR computation of *Case (i)* is reported in **Figure 1**, where both training and execution data flows are reported. The training data flows are reported as *dashed lines*, while the execution data flows are represented as *dotted lines*. The training phase is fed by using data

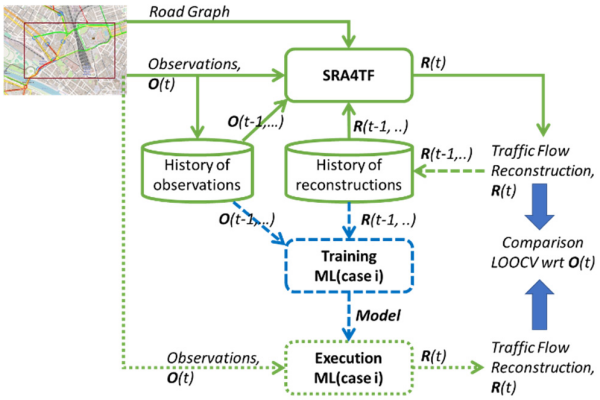

Fig. 1. Hybrid training (continuous and dashed lines) and ML data driven execution (dotted line) for traffic flow reconstruction. Continuous lines describe the model-based traffic flow reconstruction flows.

produced by both observation and SRA4TF solution (green lines). ML approach in *Case (i)* learns a *Model* able to produce a full set of traffic flow densities on the basis of observations, that is the TFR, at each time instant, disregarding its temporal evolution.

The ML is trained without considering the temporal information related to the evolution of time series:

$$\hat{f}(\boldsymbol{O}(t)) \rightarrow \boldsymbol{R}(t) \qquad Case(i)$$

Thus, the SRA4TF is used for generating dense traffic flow training data with respect to observed values, for the ML function $\hat{f}(.)$. Moreover, function $\hat{f}(.)$ learns how to compute the TFR according with $\boldsymbol{R}(t)$ on the basis of the observed values $\boldsymbol{O}(t)$. Once trained, the ML solution could be used at run time to produce dense traffic flow results in faster manner (if compared to PDE iterative solution). The resulting $\boldsymbol{R}(t)$ can be compared with the measured values obtained by sensors by using the LOOCV approach in specific $\boldsymbol{O}(t)$ locations, thus estimating the *RMSE* as depicted by means of the bold arrows in **Figure 1**. This has allowed us to assess the precision of the produced results by using *Case (i)* proposed.

IV. Urban Context and Assessment

In order to assess the accuracy of the estimated $\boldsymbol{R}(t)$ from SRA4TF and from ML solutions, beyond the training period, the estimated $\boldsymbol{R}(t)$ has to be compared with respect to the $\boldsymbol{O}(t)$ by using the LOOCV approach and thus estimating the *RMSE* as described in Section II. In terms of performance, the main advantages of a data-driven model usage have to do with time efficiency with respect to the SRA4TF solution which is iterative.

A. Data vs City Scenario

As to the assessment of the proposed solution, a small road network (subnet of the whole metropolitan traffic network of Florence) included in the bounding box in **Figure 2** has been taken under exam; 7 traffic sensors are located and denoted as: METRO707, METRO709, METRO740, METRO741, METRO756, METRO757 and METRO814 (in the context of https://www.snap4city.org Florence knowledge

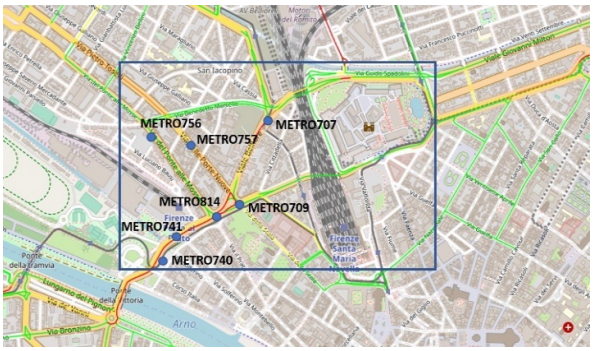

Fig. 2. Representation of real-time traffic flow reconstruction data over the network within the city of Florence. The bounding box delimits the subnetwork where data have been taken from to be analyzed in present work.

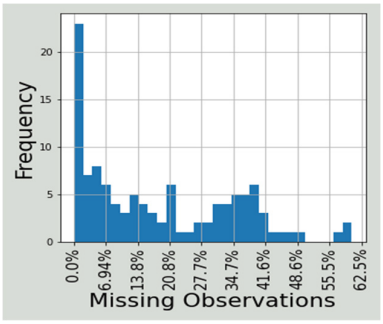

Fig. 3. Histogram of missing observations during the selected time period for each day under consideration.

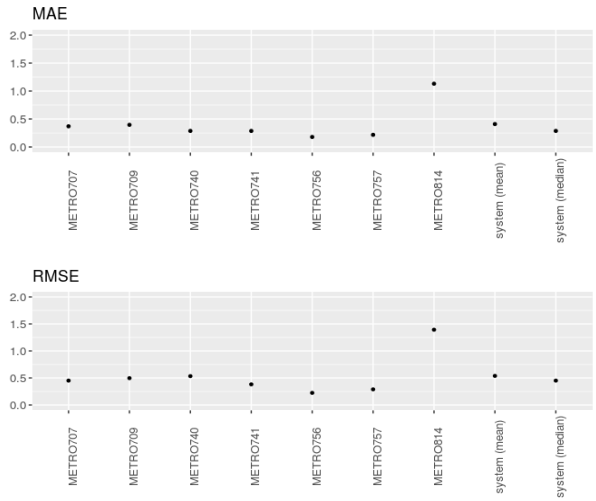

Fig. 4. MAE and RMSE at traffic sensors for SRA4TF.

base and Organization). The considered area is constituted by 735 road segments (units) and 103 intersections/junctions or nodes, thus TDMs for SRA4TF.

The area of **Figure 2** is part of the road network of the metropolitan area of Florence where the SRA4TF solution for computing TFR has been operative since many years. By means of incoming/outcoming traffic flows observations on the border, the selected subnet satisfies the traffic flow conservations in the area leading to a correct SRA4TF model application. Moreover, the selected subnet is relevant in terms of traffic flow in the city of Florence, since it constitutes one of the high traffic areas and includes one of the main accesses to city downtown and main railway station.

The training set is based on traffic sensor data updated every (about) 10 minutes (144 measures should be observed per day per sensor) during the weeks from 2019 November 1st to 2020 February 29th, i.e., 24 (hours) per 121 (days). The entire dataset is composed by 13208 observations $\boldsymbol{O}(.)$ from 7 sensors, while 13208 reconstructions $\boldsymbol{R}(.)$ of the traffic density can be computed in 728 units composing the selected subnet of 735. During the day some observations may be missing from some or all the sensors due to a given number of reasons (lack of connectivity, faults, maintenance, etc.): when many observations are missing, SRA4TF does not produce results.What may physically happen is that one or more sensors would not provide data for some samples or even days, and this occurrence can be regarded as *local missing* Spatial and Temporal at the same time. In some special cases, the whole area may lack of data when the gateway is under maintenance; therefore, a complete global missing data for the whole area is obviously spatial and temporal together. As to the time period taken for training and test, most days did have all the correct 96 observations (each sensor produces 4 measures per hour, and thus 96 per day). In fact, from **Figure 3**, 23 days show all the values from sensors, while the remaining days have some missing values. Most days have more than 60% of their traffic sensors values. Therefore, local missing and short time global missing are solved as discussed hereafter. Global missing for long periods may be covered with the so-called typical time trends computed on statistical basis and long terms predictions. Sensitive analysis on missing rates for short terms predictions has been carried out in [9].

B. Estimated Traffic Flow Reconstruction via SRA4TF

Please note that one of the aims of this proposed solution is to reduce errors in computing the TFR produced by SRA4TF being the basis of *Case (i)* architecture, as reported in **Figure 1**. For this reason, in this section, we are recalling a description of $\boldsymbol{R}(t)$ produced by SRA4TF with respect to observations $\boldsymbol{O}(t)$. The SRA4TF solution has turned out to be one of the best solutions in the state of the art in [32].

Thus, the assessment reported in **Figure 4** depicts both *MAE* and *RMSE* (at level of sensor location using LOOCV), over 3500 timestamps (which constituted about 30% of the above-described dataset). The reported errors are associated with each traffic sensor where its actual value is also estimated and its average estimation, in terms of (mean and median) system error, is also considered in the selected subnet.

According to LOOCV approach, the positions of sensors present *MAE* and *RMSE* values close to (or less than) the vehicular density of 0.5 cars/20m, except for METRO814. In such a location sensor measured traffic data are very high and they are typically 2 or 3 times greater than others, therefore traffic volume is affecting model accuracy. Yet, normalized errors, with respect to the traffic volume in each sensor location, allow a similar behavior as described in [32].

C. Assessment Metrics

The above presented approach has been validated by taking into account different aspects: (i) the observed data are only available on sensor locations, (ii) the validation can be performed using LOOCV scheme, (iii) the aim is to reduce the general error of SRA4TF in computing TFR. Once the training with ML approaches has been performed, both *MAE* and *RMSE* between the left-out target sensor and the estimated value in the observation for both cases are viable.

Therefore, the new hybrid approach of **Figure 1** and original SRA4TF can be compared on the basis of *MAE* and *RMSE* via LOOCV on single sensor position or at system level *MAE(system), RMSE(system).* In addition, we can compare the new hybrid approach and the original SRA4TF by comparing the whole TFR in all segments; we have to be sure that convergence on sensors location, aiming at reducing the error in those locations, does not degenerate the precision in the other segments. More precisely, the mean TFR deviation ΔR is computed as:

$$\Delta R = \frac{1}{T}\sum_{t=1}^{T} \Delta R(t).$$

where instant deviation is:

$$\Delta R(t) = \frac{1}{n}\sum_{z=1}^{n} |\hat{R}_z(t) - R_z(t)|$$

and where: $R_z(t)$ is the traffic density value of the z^{th} reconstructed unit at timestamp t using SRT4TF and $\hat{R}_z(t)$ is the reconstructed traffic density value by the data-driven model of the z^{th} unit at timestamp t by the data-driven model.

V. Machine Learning Approaches Case (i)

In this section, we report the outcomes related to the usage of the above-mentioned machine learning techniques in the architectural context described in **Figure 1**. Different ML solutions have been compared, with the aim of identifying the best possible solution to learn and compute TFR. To this end, we have considered ensemble learning techniques such as **Adaboost** [40], Random Forest, **RF** [41], and **XGboost** [42]. However, we took into account also more concise and interpretable models such as a **Bayesian** regressor [43], a Decision Tree, **DT** [44], **ExtraTree** [45], and multi-layer perceptron, **MLP** [46]**.** Other ML and deep learning, DL, architectures could be applied: the value of what is proposed in this paper is not in the specific adopted model, but in the hybrid architecture. In fact, we could demonstrate that a number of ML approaches can be used for the same purpose, maybe with some adaptations according to the ML/DL adopted models.

All the models have been trained with the same training set and validated on the same validation data set, so as to perform the comparison on the same conditions. As mentioned in Section IV-C, validation data set is constituted of about 30% of the entire dataset which is described in Section IV-A. The remaining 70% is devoted to the training phase. The selection has been random. As to experiment settings, we adopted the following parameters. For **Adaboost** a maximum of 50 decision trees has been used with a maximum depth of 3 to improve the error. **RF** had 100 decision trees each as base estimator that is grown to minimize the absolute error with the target without limitation in depth. **XGBoost** model enclosed decision trees built with a max depth of 6 using all features to improve the previously fitted tree. **Bayesian** model employed a Gamma distribution prior for the estimation of the parameters, having as hyperparameters alpha and lambda equal to $1e^{-6}$. The **DTs** had no limits in depth and have been trained to minimize the absolute error with the target. **ExtraTrees** have been built using a maximum of 100 decision trees using all features to find the best split. **MLP** had a single hidden layer composed by 100 neurons using as activation function ReLu, and the activation function of the output layer has been ReLu; the optimizer used to train the network has been Adam, the network has been trained by using MAE as loss function.

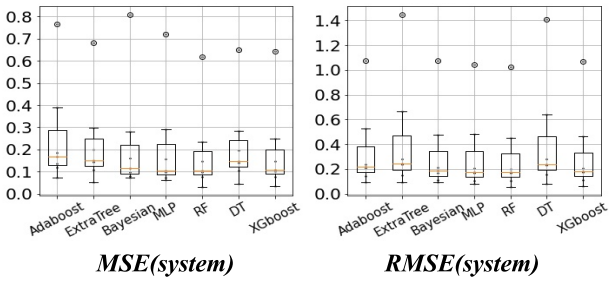

Fig. 5. MAE(system) and RMSE(system) of TFR for Case (i), with their confidence intervals. For STR4FT, MAE(system) = 0.4, and RMSE(system) = 0.53, have been registered.

A. Experimental Results for Case (i)

According to the above-described assessment, for *Case (i) MAE* and *RMSE* have been computed for SRA4TF method and compared with the results obtained by using the above presented ML techniques. Both *MAE(system)* and *RMSE(system)* estimated for the above described ML models by using LOOCV approach are reported in **Figure 5**.

According to the results reported in **Figure 5**, **RF** model turned out to be the best, as it provides the smallest *MAE(), RMSE()* and confidence interval values. Moreover, almost all ML approaches tested in the context of *Case (i)* could improve the STR4FT solution (the reconstruction of the last unseen 3500 timestamps). The mean improvement has been in the range of *delta MAE(system)* of 0.22, which is a reduction of more than the 50% of the STR4FT error in estimating traffic flow. The performance of ML solutions in terms of ΔR are listed in **Table I**. Also in this case, **RF** turned out to be the best model in reducing the difference in all TFR segments with respect to STR4FT solution.

In terms of execution time for TFR, we used the test set partition, composed by 3500 timestamps. The results are listed in **Table II**. The executions have been conducted on a GPU board NVIDIA Quadro GV100 with 32GByte Ram, which has 5120 CUDA Cores, FP64 perf as 7.4 TFLOPS. Therefore, in terms of execution time, RF improved the execution time of SRA4TF, providing a speed up of about 2. A better compromise can consist in the adoption of MLP which is not the best solution in terms of error reduction (see **Table I**), and

TABLE I
TFR DEVIATION ΔR ACCORDING TO THE DIFFERENT MODELS IN CASE (i)

Model	ΔR
Bayesian	0.0942
Adaboost	0.0848
MLP	0.0676
ExtraTree	0.0552
DT	0.0519
XGboost	0.0467
RF	0.0435

TABLE II
EXECUTION TIMES FOR THE TFR PERFORMED WITH SRA4TF ONLY, AND VIA ML MODELS FOR CASE (i)

Model	Test Time (s)
SRA4TF	3685.15
RF	1627.30
XGboost	744.50
Adaboost	43.66
DT	19.52
ExtraTree	18.47
Bayesian	4.69
MLP	0.22

it provides a **speed up** of about 16000 times with respect to the SRA4TF execution time.

VI. CODING TEMPORAL INFORMATION: CASE (ii)

A further analysis has been performed to improve the precision of the ML phase in terms of system *MAE*, *RMSE*. In this *Case (ii)*, the model of *Case (i)* has been enriched by adding in inputs different kinds of coding for temporal information, while addressing problems related to discontinuous input data. This latter problem was not addressed in Case (i) which produced results only based on input data presence, while the rate of missing in the realistic case produced also some sporadic missing in the output. The problem can be largely overcome with some imputation via predictions or typical time trends [9]. When temporal data are adopted in the model, the impact of missing data can be much higher and thus has to be addressed.

Moreover, temporal information has to be encoded in features so as to consider data seasonality, which can be daily and weekly mainly. To this end, we (i) tagged days as festive, pre-festive and working days; and (ii) we added the time slot of the day.The added features can be estimated a priori by knowing the day conditions and its related time slot. To this end, an analysis has been performed to identify those classes and make the addition of this information easier at each time stamp. In some cases, a weekday in the middle of the week may present a traffic flow profile like a festive one, for example in the event of national or religious festivities. For instance, the days of the 1st November, 24th, 25th, 26th, 30th and 31st of December, despite being working days, have a traffic flow profile that resemble much more a festive/pre-festive day according to the clustering, being indeed related to Christmas, Christmas Eve and New year Eve festivities. The day tag has been assessed automatically by using the

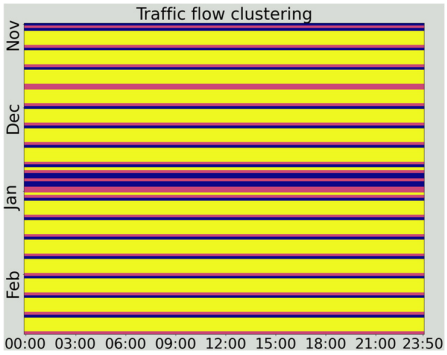

Fig. 6. Horizontal clustering identifying the festive, pre-festive, and working days: in dark purple the festive days, in dark pink the pre-festive days and in yellow the working days.

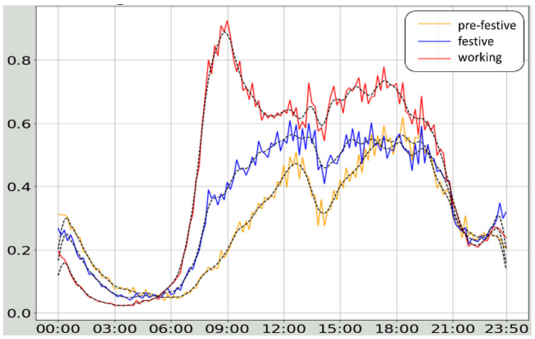

Fig. 7. Example of trends for the observed traffic flow density (cars/20m) in the different clusters, identified as festive, pre-festive, and working days.

K-Means unsupervised clustering technique with k equal to 3 (the festive, pre-festive, and working days). Thus, traffic flow data trends over 24 hours have been clustered. Similarity metric for the clustering has been the Euclidean distance with the nearest neighbours. As a result, days providing similar trends have been grouped (see **Figure 6**).

Typical trends are reported in **Figure 7**. We denoted this feature as $\boldsymbol{d}(t)$, which is an additional info for each observed sensor data. The feature $\boldsymbol{d}(t)$ is defined as the average value (cars/20m) estimated over a range of days at a given sensor location at time t, thus being a typical time value/trend.

According to the day specification we have a typical trend for festive, pre-festive and working days, respectively, as sketched in **Figure 7**.

An analysis over time has been performed likewise on traffic flow data. Final choice has been to encode the temporal information into hours as additional temporal feature $\boldsymbol{h}(t)$ added to the input. The feature $\boldsymbol{h}(t)$ defines different time slots where data are performed, thus $\boldsymbol{h}(t)$ can be in the range [0-23] and successively normalized to stay on [0.0-1.0]. Alternatively, daily hours can be coded in 4 time slots as typically happening in many transport system applications. The adopted partition in 4 time slots has been [start, end] as follows: [00:00, 05:59], [06:00, 11:59], [12:00, 17:59], [18:00, 23:59]. These slots correspond to a quite uniform traffic behavior.

A. Assessing Results by Comparison)

Therefore, the model enriched with temporal explicit information with $\boldsymbol{h}(t)$ and $\boldsymbol{d}(t)$ assumes the form:

$$\hat{f}(\mathbf{O}(t), \mathbf{h}(t), \mathbf{d}(t)) \rightarrow \mathbf{R}(t) \qquad Case(ii)$$

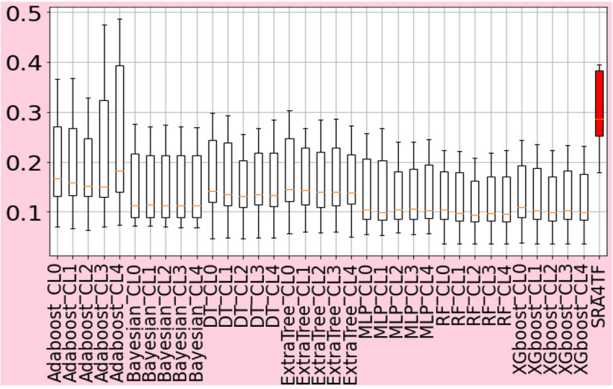

Fig. 8. Results in terms of MAE(system). For each ML model, 5 values represent the results for cases CL0-CL4. The last box on the right in red depicts the distribution of the error produced by SRA4TF.

The above presented temporal information improved both LOOCV approach in terms of *MAE(system)* or *RMSE(system),* and TFR deviation ΔR with respect to *Case (i)* which does not take into account temporal information. Hereafter we refer to 4 different cases.

1) CL0 is *Case (i)* as described in Section V.
2) CL1 is the case where the $\boldsymbol{d}(t)$ are 3 classes and $\boldsymbol{h}(t)$ are in 4 time slots. They are coded together into $3\times4=12$ possible values in a unique input data encoded together.
3) CL2 is the case where $\boldsymbol{d}(t)$ are 3 classes and $\boldsymbol{h}(t)$ are in 24 time slots. They are separately encoded.
4) CL3 is the case where $\boldsymbol{d}(t)$ are 3 classes and $\boldsymbol{h}(t)$ are in 4 time slots, while they are separately encoded, which makes it different from CL1.
5) CL4 is the case where $\boldsymbol{d}(t)$ are 3 classes and $\boldsymbol{h}(t)$ are in 48 time slots. They are separately encoded.

For this reason, *Case (ii)* has been substantially implemented in 4 different encoding cases. As a result, temporal information of $\boldsymbol{d}(t)$ and $\boldsymbol{h}(t)$ led to a performance improvement for every model. Generally, CL1-CL4 cases improved CL0 cases, except for Adaboost model which admitted CL3 and CL4 larger than CL0, while CL1 and CL2 are better than CL0. The improvement in the reconstruction error of real observed traffic flow data has been observed in terms of *MAE(system)* (see **Figure 8**) for almost all the techniques. The cases passing from CL0 to CL4 provide an increment of the complexity of the input data. In these conditions Adaboost provided a decrement of performance since it is less capable to learn non-linear models than the others and it is more sensitive to noise (the reduction of time slots reduce the noise). The best results have been obtained by **RF**.

Since changes are not easily observable in Figure 8, in **Figure 9** a differential representation is presented. The largest improvement has been obtained by the DT model, while a detrimental effect has been observed for the Adaboost Model. Taking into account all the different time encoding in Case (ii), the best has been CL2.

Thus, best results have been obtained by RF for CL2, which takes into account 3 classes clustering and 24 hours coding. In **Figure 10,** distributions of delta MAE of TFR of model

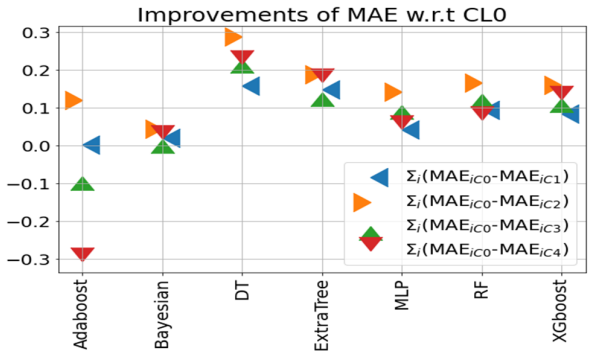

Fig. 9. The differences of performances in terms of MAE(system) of TFR for Cases (ii) with respect to Case (i) (i.e., CL0).

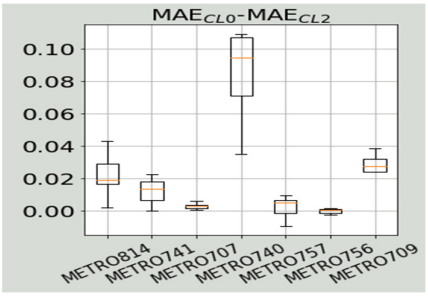

Fig. 10. The delta MAE of TFR for case RF CL2 with respect to CL0, Case (i), for all traffic flow sensors.

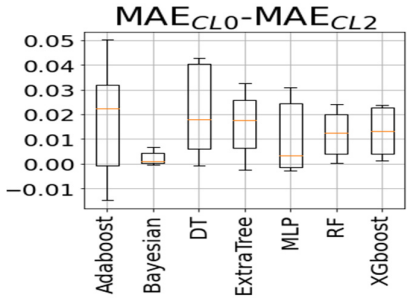

Fig. 11. Comparison of improvements expressed as the differences of MAE(system) obtained for TFR estimation by using different ML models for case CL2 (with temporal information) with respect to CL0 (without temporal information).

CL2 with respect to CL0 (case (i)) for the traffic flow sensors where the error is assessed. The largest improvement has been observed in the reconstruction of the observed traffic flow data for sensor METRO740.

In **Figure 11,** the improvements obtained by considering temporal information are reported as distributions of *delta MAE(system)* of the TFR of model CL2 with respect to CL0 (*case (i)*) for the different ML models. With a median improvement of 0.0129 for *MAE*, the additional information could generate the highest improvement for DT model. According to **Figure 8**, RF CL2 is the best solution in terms of *MAE(system).*

In **Figure 12**, TFR deviation ΔR obtained by CL2 with respect to SRA4TF method is reported. Almost all ML models provided marginal changes.

Finally, according to **Figure 8,** *MAE(system)* obtained by the data driven models with respect to SRA4TF has generally

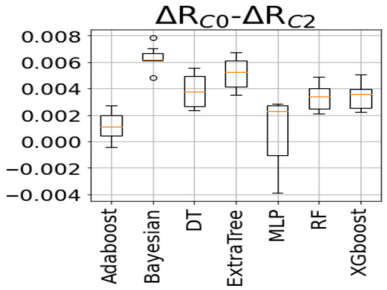

Fig. 12. Changes in terms of ΔR for Case(i), CL0 and CL2 (case (ii)). The additional information has slightly changed the related variation in terms of ΔR.

improved the estimation precision. More precisely, RF model, according to CL2 configuration, has provided a *MAE(system)* of 0.16, against the value of 0.4 of SRA4TF. The time needed for the LOOCV computation by data-driven models is lower if compared to SRA4TF. Therefore, the goal of both improving accuracy of traffic flow reconstruction and providing faster execution times has been achieved.

As a remark, we can finally assert that the speed up obtained by RF in terms of execution time is lower than SRA4TF, which is one of the faster estimators in literature [32]. On the other hand, a good compromise in term of performance and speed-up could be MLP with a speed up of 16000 with respect to the SRA4TF execution time and a precision comparable to the one of RF.

VII. Conclusion

In this work, a new architecture for computing traffic flow reconstructions from sensors data has been presented. The solution is based on a hybrid architecture combining model and data driven approaches. It starts from a model driven SRA4TF to compute dense traffic flow data in the road network, while any other estimators could be used for the same purpose. Machine learning models have been used to improve dense traffic flow data resulting from SRA4TF. The paper presented and compared various solutions for machine learning models. This comparison allowed us to identify the best possible solution based on RF, Random Forest. The current solution has improved results in terms of: (a) **precision** of the TFR as *MAE(system)* of more than 0.2, (b) **speed up** the computational time needed for TFR estimation, thus allowing the computation to be more sustainable on large networks. Our outcomes did demonstrate that the integration of model driven, and data driven is possible. Several ML approaches have been used, and results have proven that many of them can be profitably used to improve precision and speed-up. As a final remark, the approach proposed for improving the computed TFR could be applied to any kind of traffic flow estimation models, no matter which way they are computed. Furthermore, both approach and architecture could be also used to improve the results produced by other PDE resolutions. This could be done with finite element approaches, which are always very time consuming. On this regard several applications in the fluid dynamic and hydraulic fields can be easily foreseen.

Acknowledgment

The authors would like to thank MIUR, both University of Florence and companies involved for co-funding Sii-Mobility National Project on Smart City Mobility and Transport, the National Center on Sustainable Mobility (MOST, Centro Nazionale per la Mobilità Sostenibile) (both of the Italian Ministry of Research), and the National Ph.D. Program in Artificial Intelligence on AI for Society. Snap4City is an open technology and research of DISIT Laboratory. Sii-Mobility is grounded and has contributed to Snap4City open solution and infrastructure where the discussed results are operative.

References

[1] S. Bilotta and P. Nesi, "Estimating CO_2 emissions from IoT traffic flow sensors and reconstruction," *Sensors*, vol. 22, no. 9, p. 3382, Apr. 2022.
[2] C. Badii et al., "High density real-time air quality derived services from IoT networks," *Sensors*, vol. 20, no. 18, p. 5435, Sep. 2020.
[3] 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.
[4] Y. Kamarianakis and P. Prastacos, "Space–time modeling of traffic flow," *Comput. Geosci.*, vol. 31, no. 2, pp. 119–133, Mar. 2005.
[5] 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.
[6] 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.
[7] 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.
[8] S. Bilotta, P. Nesi, and I. Paoli, "Real-time system for short- and long-term prediction of vehicle flow," in *Proc. IEEE Int. Conf. Smart Data Services (SMDS)*, Oct. 2020, pp. 97–104.
[9] S. Bilotta, E. Collini, P. Nesi, and G. Pantaleo, "Short-term prediction of city traffic flow via convolutional deep learning," *IEEE Access*, vol. 10, pp. 113086–113099, 2022.
[10] 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.
[11] 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.
[12] M. Y. Darus and K. A. Bakar, "Congestion control algorithm in VANETs," *World Appl. Sci. J.*, vol. 21, pp. 1057–1061, Jun. 2013.
[13] 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.
[14] 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.
[15] M. Picone, M. Amoretti, and F. Zanichelli, "Simulating smart cities with DEUS," in *Proc. 5th Int. Conf. Simul. Tools Techn.*, 2012, pp. 172–177.
[16] 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.
[17] D. Xu, C. Wei, P. Peng, Q. Xuan, and H. Guo, "GE-GAN: A novel deep learning framework for road traffic state estimation," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102635.
[18] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[19] E. Santana et al., "Interscsimulator: Large-scale traffic simulation in smart cities using Erlang," in *Proc. Int. Workshop Multi-Agent Syst. Agent-Based Simul.* Cham, Switzerland: Springer, 2017, pp. 211–227.
[20] P. A. Lopez et al., "Microscopic traffic simulation using sumo," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.

[21] D. T. Hunt and A. L. Kornhauser, "Assigning traffic over essentially-least-cost paths," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1556, no. 1, pp. 1–7, Jan. 1996.
[22] M. Ben-Akiva and S. R. Lerman, *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA, USA: MIT Press, 1985.
[23] J. de Dios Ortùzar and L. G. Willumsen, *Modelling Transport*. Hoboken, NJ, USA: Wiley, 2011.
[24] T. Alghamdi, S. Mostafi, G. Abdelkader, and K. Elgazzar, "A comparative study on traffic modeling techniques for predicting and simulating traffic behavior," *Future Internet*, vol. 14, no. 10, p. 294, Oct. 2022.
[25] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.
[26] P. I. Richards, "Shock waves on the highway," *Operation Res.*, vol. 4, pp. 42–51, Feb. 1956.
[27] G. Bretti, R. Natalini, and B. Piccoli, "A fluid-dynamic traffic model on road networks," *Arch. Comput. Methods Eng.*, vol. 14, no. 2, pp. 139–172, Jun. 2007.
[28] S. K. Godunov, "A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics," *Mathematics Sbornik*, vol. 47, pp. 271–290, Oct. 1959.
[29] P. Kachroo and S. Sastry, "Travel time dynamics for intelligent transportation systems: Theory and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 385–394, Feb. 2016.
[30] P. Bellini et al., "Real-time traffic estimation of unmonitored roads," in *Proc. IEEE 16th Intl. Conf. Dependable, Autonomic Secure Comput., 16th Intl. Conf. Pervasive Intell. Comput., 4th Intl. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr.*, Aug. 2018, pp. 935–942.
[31] P. Bellini et al., "Knowledge modelling and management for mobility and transportation applications," in *Proc. Workshop Technol. Converg. Smart Cities*, 2018, pp. 1–8.
[32] 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.
[33] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Graph convolutional networks for road networks," in *Proc. 27th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2019, pp. 460–463.
[34] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annu. Rev. Control*, vol. 43, pp. 128–151, 2017.
[35] J. Liu, M. Barreau, M. Čičić, and K. H. Johansson, "Learning-based traffic state reconstruction using probe vehicles," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 87–92, 2021.
[36] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
[37] J. Huang and S. Agarwal, "Physics informed deep learning for traffic state estimation," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, 2020, pp. 1–6.
[38] R. Shi, Z. Mo, K. Huang, X. Di, and Q. Du, "Physics-informed deep learning for traffic state estimation," 2021, *arXiv.2101.06580*.
[39] F. Rempe, A. Loder, and K. Bogenberger, "Estimating motorway traffic states with data fusion and physics-informed deep learning," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2208–2214.
[40] H. Drucker, "Improving regressors using boosting techniques," in *Proc. ICML*, vol. 97, Jul. 1997, pp. 107–115.
[41] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[42] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[43] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jan. 2001.
[44] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
[45] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
[46] G. E. Hinton, "Connectionist learning procedures," in *Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1990, pp. 555–610.
[47] Sii-Mobility. *Italian National Mobility and Transport Action for Sustainable Mobility*, Founded by MIUR. Accessed: Nov. 4, 2023. [Online]. Available: http://www.sii-mobility.org
[48] Universities and Companies. *National Center on Sustainable Mobility (MOST) Involving MIUR*. Accessed: Nov. 4, 2023. [Online]. Available: https://www.centronazionalemost.it/
[49] *Km4City Ontology, Knowledge Base 4 the City*. [Online]. Available: https://www.km4city.org and https://www.snap4city.org
[50] *Snap4City*. [Online]. Available: https://www.snap4city.org
[51] C. Badii et al., "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.

Stefano Bilotta is a Researcher at the University of Florence, DIMAI, and with DISIT Laboratory. He has been involved in international projects as: Sii-Mobility, Snap4City and Trafair, and in the CN Most. His research interests include simulation, artificial intelligence, parallel solution, dynamic systems, machine learning, languages and coding theory, applied in the domains of mobility and transport, traffic flow reconstruction algorithms.

Valerio Bonsignori is a Ph.D. student on the national Ph.D.-AI course. He has been involved in the CN Most. His interested are on artificial intelligence.

Paolo Nesi (Member, IEEE) is a Full Professor at the University of Florence, DINFO Department, Chief of DISIT Laboratory and of Snap4City action. He is and has been the coordinator of several Research and Development multipartner international and national projects. He has published more than 400 papers on international journals and conferences. His research interests include artificial intelligence, massive parallel and distributed systems, physical models, the IoT, mobility, big data analytic, AI/XAI, neuro-symbolic computing, formal model, machine learning, and data privacy. He has chaired of a number of international conferences.

Open Access funding provided by 'Università degli Studi di Firenze' within the CRUI CARE Agreement