Traffic Simulator AI-Based Surrogate for an Urban Road Network

Cantatore, F.; Raimondi, G.; Oneto, L.; Coraddu, A.; Pasquale, C.; Siri, E.; Siri, S.; Sacone, S.; Anguita, D.

DOI
10.1109/ITSC58415.2024.10919938

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the IEEE 27th International Conference on Intelligent Transportation Systems (ITSC 2024)

2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

Traffic Simulator AI-based Surrogate for an Urban Road Network

F. Cantatore, G. Raimondi, L. Oneto, A. Coraddu, C. Pasquale, E. Siri, S. Siri, S. Sacone, D. Anguita

2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-0592-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITSC58415.2024.10919938

***Abstract*— Relying on a traffic simulator is often necessary when multiple traffic conditions need to be replicated, either to predict specific quantities of particular interest or to assess more general network properties, such as efficiency or resilience, under different scenarios. While potentially delivering a high level of fidelity, producing such a simulation may come at a prohibitive computational cost when it is applied in a real context, making this approach unsuitable for real-time applications in most cases. In this regard, the goal of the present work is twofold. Firstly, we aim to surrogate a traffic simulator with a data-driven approach in order to produce real-time traffic predictions that are also sufficiently accurate and effective. Secondly, we want to determine the minimum amount of data, i.e., the smallest number of sensors deployed on the network, that still allow to obtain predictions within a predefined bound. The effectiveness of the approach is evaluated on the full-scale urban network of Rapallo, Italy, in which we employ the AIMSUN NEXT simulator targeting the morning peak hours, i.e. between 7:00 a.m. and 9:00 a.m. In the paper, multiple state-of-the art ML algorithms are tested to assess their effectiveness as surrogate models under the considered problem.**

I. Introduction

Being able to estimate and predict the evolution, in time and space, of a traffic system provides the foundation for any effective evaluation, management and planning strategy, especially when it comes to the complexity of urban areas. Starting from the ‘50s, both researchers and practitioners have developed numerous approaches in order to model the various traffic phenomena emerging from the interaction between users and the transport infrastructure. Over time, these efforts have yielded the development of multiple traffic flow models, each tailored to specific applications and levels of detail. These models, generally classified in microscopic, mesoscopic, and macroscopic, have been extensively explored and documented for example in [1]–[4].

In particular, microscopic models are often incorporated into traffic simulators, which are generally adopted to create a synthetic environment to reproduce a real system and to obtain the main traffic quantities, such as flow and speed, associated with all the roads of the network. Such simulators are also exploited, more in general, to evaluate the efficiency of transport networks in different scenarios by calculating specific performance indices (e.g., average travel time, fuel consumption, pollutant emissions). These scenarios may relate to fluctuations of mobility demand on the network, considered as variations in its intensity and its spatio-temporal distribution, changes in the network layout due to the introduction/subtraction of network components (e.g., new infrastructure, modification of existing layout, collapse of infrastructure, natural disasters that limit the functionality of transport systems, terrorist attacks, etc.), and scenarios generated by the application of traffic regulation and control policies. Several traffic simulators have been developed over the years. Among those currently available, the most popular are: AIMSUN NEXT [5], [6], SUMO [7], [8], VISSIM [9] and VISUM [10], other simulation tools can be found in survey papers [11], [12]. All these traffic simulators differ in the types of simulation they can perform, the characteristics of the simulations, and the type of operating system supported.

The development of traffic modeling and estimation approaches based on both pure data-driven methods [13], [14] and physics-informed data-driven methods [15], [16] has received great attention recently. Although effective, data-driven methods cannot replace traffic simulators as they are strongly dependent on the data on which they have been trained. On the other hand, the higher reliability of the results produced by a simulation comes at a price in computational cost and thus in processing time, which can potentially be quite significant when dealing with large-scale networks or when particular features requires a sophisticated and fine representation.

The present work proposes an AI-based surrogate model to replicate, with some approximation but with limited computational time, the outputs produced by a traffic simulator to be used for real-time optimization or control purposes. In the literature related to vehicular traffic, it is possible to find a few examples of surrogates models applied either for forecasting purposes [17] or to surrogate traffic simulators specifically oriented to the definition of safety indices [18], [19]. In other works, surrogate applications are proposed for skirting the issue of the computational complexity of some optimization problems by applying suboptimal solutions in real time [20]. Differently from the existing literature, the objective of this work is twofold: on the one hand, we want to propose a general methodology to realize a surrogate model capable of forecasting average speed and flow on an urban road transport network and, on the other hand, we want to propose a method that can be used to define the minimum number and location of traffic measurements necessary to carry out a reliable traffic prediction. In our opinion, this last point is particularly important because a proper definition of the number and location of traffic measures would effectively

F. Cantatore, G. Raimondi, L. Oneto, C. Pasquale, S. Siri, S. Sacone, D. Anguita are with the University of Genova, Via Opera Pia 11a/13, 16145, Genova, Italy

E. Siri is with the Université Côte d’Azur, Inria, CNRS, LJAD 2004 route des Lucioles BP 93, 06902 Sophia Antipolis Cedex, France

A. Coraddu is with Delft University of Technology, Mekelweg 5, 2628 CC, Delft, The Netherlands

support the design of the monitoring network of an urban area.

The present paper is organized as follows. Section II describes the considered problems and the real case study on which the traffic data have been generated. In Section III, the surrogate traffic simulation model is given, while the results obtained by testing it on a real case study are presented in Section IV. Finally, Section V provides concluding remarks on the proposed approach.

II. Problem Description and Data Generation

The traffic simulator surrogate model proposed in this paper allows us to address the following problems:

1) Problem (A): prediction of speed and flow profiles over the entire traffic network under consideration;
2) Problem (B): definition of the minimum number of sensors needed to guarantee both a reliable estimation and prediction of speed and flow profiles on the remaining network.

This study makes use of AIMSUN NEXT as the foundation for the development of the surrogate model. It is worth noting that the methodology proposed here is versatile enough to be applied to other commercially available traffic simulators as well as to any output generated by these simulators without any significant adjustment.

The simulation has been calibrated and validated on real data collected from the urban area of Rapallo, Italy. The considered road network is reported in Figure 1, where the letters from A to U represent the considered 19 centroids. The simulated network includes all the relevant roads of the city, that are 332, reporting their physical characteristics (e.g., capacity) in detail and also including all the real-time cycles of the traffic lights. The division of the city into 19 areas has been done considering the main attraction and origin points in the different zones of the city, the arterial roads connected with the city, the parking areas in the city center, and so on. The origin-destination matrix has been estimated based on the mobility requirements of the users living and working in Rapallo city, and then such a matrix has been calibrated by means of AIMSUN NEXT considering real-time measurements from the network obtained both with fixed cameras and with mobile sensors provided by the Municipality.

The simulations used for this paper consider a generic weekday morning, from 7:00 a.m. to 9:00 a.m., which corresponds to a critical period from a mobility point of view since it includes the peak hours in which high densities can be detected on the main roads. Such roads are indeed too narrow to receive the high mobility demand that is concentrated in short time intervals. In particular, all the simulations are used as measurements in the first hour from 7:00 a.m. to 8:00 a.m. and for the predictions from 8:00 a.m. to 9:00 a.m.

The dataset containing the input/output relation explained in Section III has been generated using Python scripts to automatically simulate different scenarios in AIMSUN NEXT and collect the related data. Notably, 2076 different traffic

Fig. 1: The road network of Rapallo city simulated with AIMSUN NEXT

scenarios have been simulated. In each scenario, the origin-destination matrix has been randomized, adding a zero-mean Gaussian noise, with variance between 1 and 3, allowing us to consider the traffic behavior in the morning peak hours on different weekdays. For each scenario, flows and speeds in each of the 332 arcs are saved with a sampling time of 1 minute.

It is worth noting that the simulator outputs were properly filtered before using this dataset as input for the ML algorithms.

III. Surrogate Models for Realistic Operations

The problems previously described, namely to develop a surrogate model for real time prediction or optimization purposes [17], can be mapped into now-classical ML problems [21], [22]. In particular, as mentioned in Section II, two problems must be solved:

(A) Given the speed $v_i(t)$ and flow $q_i(t)$ at each road and each allowed running direction $i \in \mathcal{I}_s = \{1, \cdots, n_s\}$ by the sensors S_i, where n_s is the total number of sensors to measure these quantities, at time $t \in \Delta^- = \{t_0 - \delta^-, t_0 - \delta^- + \delta^t, t_0 - \delta^- + 2\delta^t, \cdots, t_0\}$, where t_0 is the present time, predict $v_i(t)$ and $q_i(t)$ $\forall i \in \mathcal{I}_s$ for $t \in \Delta^+ = \{t_0 + \delta^t, t_0 + 2\delta^t, \cdots, t_0 + \delta^+\}$;

(B) Find the minimum number of sensors S_i with $i \in \mathcal{I}_m \subseteq \mathcal{I}_s$ that, given $v_i(t)$ and $q_i(t)$ with $i \in \mathcal{I}_m$ for $t \in \Delta^-$, predict with an acceptable error for practical applications (in our case $\leq 10\%$ [23])
   (i) $v_i(t)$ and $q_i(t)$ $\forall i \in \mathcal{I}_s \setminus \mathcal{I}_m$ for $t = t_0$. This is a classical nowcasting problem.
   (ii) $v_i(t)$ and $q_i(t)$ $\forall i \in \mathcal{I}_s$ for $t \in \Delta^+$. This is a classical forecasting problem.

Note that each road $r \in \{a, b, c, \cdots\}$ may be two-way, denoted with $\leftrightarrow$ (needing then 2 sensors), or just one-way, i.e. $\leftarrow$ or $\rightarrow$.

Problem (A) can be easily mapped into a now-classical ML multi-output regression problem [21], [22]. In regression, we have an input space $\mathcal{X} \subseteq \mathbb{R}^d$ composed of d inputs (in

our case, $v_i(t)$ and $q_i(t)$ with $t \in \Delta^-$ and $i \in \mathcal{I}_s$), an output space $\mathcal{Y} \subseteq \mathbb{R}^p$ (in our case, $v_i(t)$ and $q_i(t)$ with $t \in \Delta^+$ and $i \in \mathcal{I}_s$), and a series of n examples, a dataset, of the input/output relation $\mathcal{D}_n = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$ where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ $\forall i \in \{1, \cdots, n\}$ (in our case the simulations described in Section II). The scope of the regression problem is to learn the unknown input/output relation $\mu : \mathcal{X} \to \mathcal{Y}$ based just on $\mathcal{D}_n$.

Problems (B).(i) and (B).(ii), as Problem (A), can be again mapped into a now-classical ML multi-output regression problem [21], [22] plus a feature reduction/selection phase [24], [25]. In particular, starting from Problem (A), it is possible to apply a feature reduction phase to estimate $\mathcal{I}_m$, namely the minimum number of sensors that allows to keep the prediction error below 10% in Problem (A). Then Problem (B).(i) becomes an ML multi-output regression problem where $\mathcal{X}$ counts of $v_i(t)$ and $q_i(t)$ with $t \in \Delta^-$ and $i \in \mathcal{I}_m$ and $\mathcal{Y}$ counts of $v_i(t)$ and $q_i(t)$ with $t = t_0$ and $i \in \mathcal{I}_s \setminus \mathcal{I}_m$. Moreover Problem (B).(ii) becomes an ML multi-output regression problem where $\mathcal{X}$ counts of $v_i(t)$ and $q_i(t)$ with $t \in \Delta^-$ and $i \in \mathcal{I}_m$ and $\mathcal{Y}$ counts of $v_i(t)$ and $q_i(t)$ with $t \in \Delta^+$ and $i \in \mathcal{I}_s$.

In Figure 2, a scheme of the previously discussed concept is reported. Figure 2a reports a small area of Rapallo city where, for simplicity, we consider 4 intersections connected by 3 two-way roads and 1 one-way roads. We translated this piece into a graph, see Figure 2b, and we associate the sensors to each road depending on whether it is one- or two-way. Then we depicted in Figure 2d a graphical representation of Problem (A). As for Problems (B).(i) and (B).(ii), we assumed the discovery of $\mathcal{I}_m$ reducing the graph of Figure 2b into the one of Figure 2c and then we depicted Problems (B).(i) and (B).(ii) in Figure 2e respectively.

Based on what described in Section II, we have that $t_0 =$ 8:00 a.m., $\delta^- = 1$ hour, $\delta^+ = 1$ hour, $\delta^t = 1$ minute, and $n_s = 332$ (note that from n_s one can retrieve d and p, i.e., the dimension of the inputs and the outputs in regression). Instead, the number of samples is $n = 2076$, namely, the number of simulations.

An ML regression algorithm $\mathscr{A}$, characterized by its hyperparameters $\mathcal{H}$, selects a model h based on the available data $h = \mathscr{A}_{\mathcal{H}}(\mathcal{D}_n)$. Many different ML algorithms exist in the literature [21], [24] but, as the no-free-lunch theorem states [26], there is no way to determine a-priori the best ML algorithms to use for a specific application.

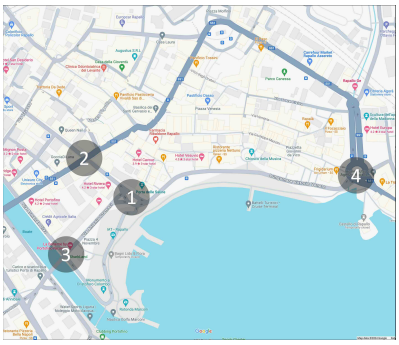
For this reason, in this work we will consider a series of different state-of-the-art ML algorithms: Ridge Regression (RR) [27] and Kernel Ridge Regression (KRR) [28], coming from the family of kernel methods [29], Random Forest (RF) [30] and XGBoost [31], from the family of ensemble methods [32], and a Single Layer Neural Network (SNN) [33] and a Graph Convolutional Network (GCN) [34], from the family of Neural Networks [22]. In this way, we have reasonably covered most of the state-of-the-art algorithms that can deal with the regression problems we face in this paper.

In RR [27], a linear model, the regularization hyperparameter λ needs to be tuned. In KRR [28], we chose to rely on the Gaussian kernel for the reason described in [35], and then the regularization hyperparameter λ and the kernel coefficient γ need to be tuned. In RF [30] we need to tune the number of features to randomly sample from the whole features during each node of each tree creation n_f and the maximum number of elements in each leaf of each tree n_l. As RF performance improves by increasing the number of trees n_t, we set it to 1000 as a reasonably large number yet computationally tractable. In XGBoost [31], we need to tune the learning rate of the gradient l_r, the max depth of each tree n_d, the minimum loss reduction m_l, the number of points to randomly sample from the whole training set for each tree creation n_b, and the number of features to randomly sample from the whole training set during the creation of each node for each tree n_f. In SNN [33], we use the sigmoid activation function in the hidden layer and the linear activation in the output layer. We use the ADAM [36] optimizer to train the model. Then we need to tune the number of hidden neurons n_h, the regularization hyperparameter λ on the weights of the last layer, and the learning rate of ADMAM l_r. In GCN [34] (and more specifically, the node labeling problem) we need to tune the number of layers n_l, the convolutions depth (number of neighborhoods) nn_i at layers i, and the number of channels ch_i at layer i. The convolutional layer has all ReLU [22] activation, the last layer is linear with linear activation. Also, in this case, we use the ADMAM optimizer, and then we need to tune its learning rate l_r. The summary of these hyperparameters with the associated search space is reported in Table I.

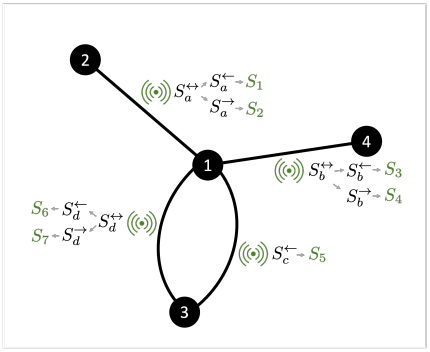
TABLE I: Hyperparameters and hyperparameters search space for all algorithms tested in this work, $d = 13$ denotes the number of features in the dataset.

Algorithm	**Hyperparameters**
RR	$\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$
KRR	$\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$, $\gamma : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$
RF	$n_f : \{d^{1/3}, d^{1/2}, d^{3/4}\}$, $n_l : \{1, 3, 5, 10\}$, $n_t : \{1000\}$
XGBoost	$l_r : \{0.01, 0.02, 0.03, 0.04, 0.05\}$, $n_d : \{3, 5, 10\}$, $m_l : \{0, 0.1, 0.2\}$, $n_b : \{0.6n, 0.8n, 1n\}$, $n_f : \{0.5d, 0.8d, 1d\}$
SNN	$h_l : \{2^5, 2^6, \cdots, 2^{16}\}$, $\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$, $l_r : \{0.1, 0.05, 0.01, 0.005, 0.001\}$
GCN	$n_l : \{1, 2, 4, 8\}$, $nn_i : \{1, 2, 3\}$, $ch_i : \{10, 50, 100, 500, 1000\}$, $l_r : \{0.1, 0.05, 0.01, 0.005, 0.001\}$

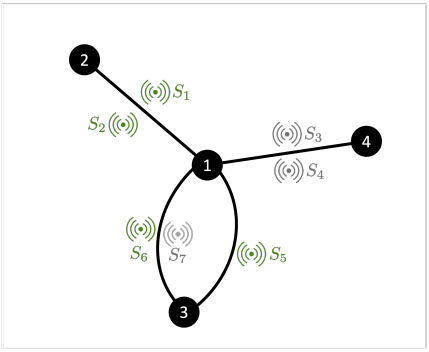
To optimize the hyperparameters and evaluate the performance of the developed model, it is essential to undertake both the Model Selection (MS) and Error Estimation (EE) phases [37]. Resampling techniques, which have gained widespread acceptance among researchers and practitioners for their efficacy in a broad array of scenarios, will be employed in this study [37]. The underlying principle of



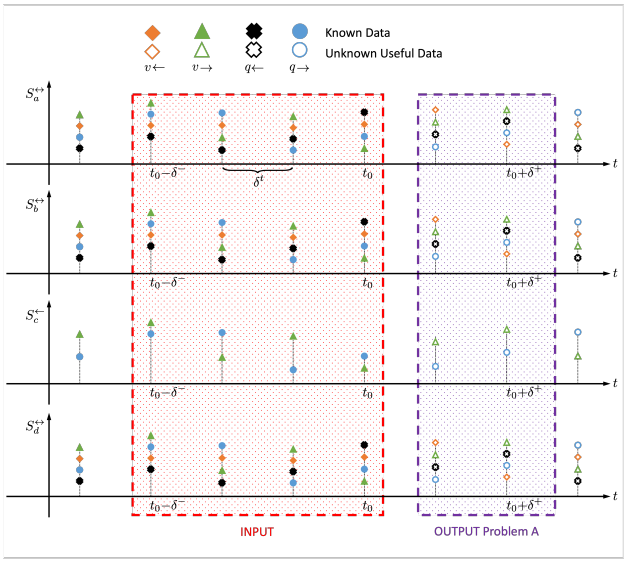
(a) Example of a small area of Rapallo city with 4 intersections, highlighted with the dark circle, connected by 3 two-way roads and 1 one-way roads.



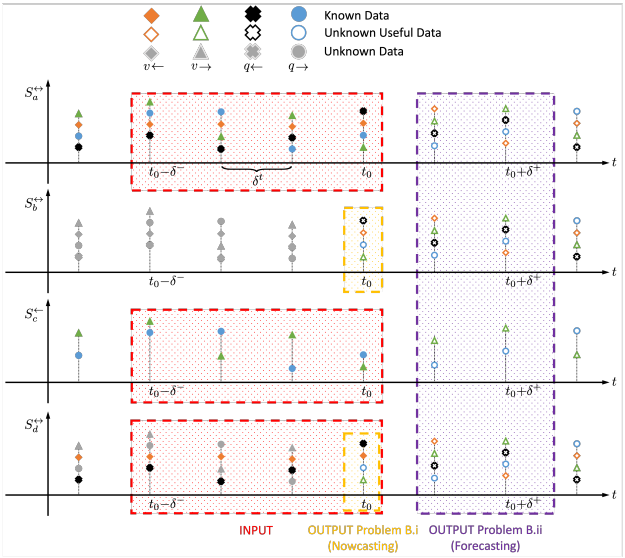
(b) Corresponding graph of Figure 2a. Green symbols are all possible active sensors needed to measure all speeds and flows on all roads.



(c) Equivalent version of the graph of Figure 2b where we assume the knowledge of the minimum number of sensors required (the green ones).



(d) Graphical representation of Problem (A) for the area of Rapallo city of Figure 2a.



(e) Graphical representation of Problems (B).(i) and Problems (B).(ii) for the area of Rapallo city of Figure 2a.

Fig. 2: Problems (A), (B).(i), and (B).(ii) in a small example case in Rapallo city.

resampling techniques involves the manipulation of the original dataset $\mathcal{D}_n$. This dataset is resampled multiple times (n_r), either with or without replacement, to generate three distinct datasets for learning, validation, and testing purposes, denoted as $\mathcal{L}_l^r$, $\mathcal{V}_v^r$, and $\mathcal{T}_t^r$, respectively, for each resample $r \in 1, \cdots, n_r$. These datasets are constructed to ensure mutual exclusivity, with $\mathcal{L}_l^r \cap \mathcal{V}_v^r = \oslash$, $\mathcal{L}_l^r \cap \mathcal{T}_t^r = \oslash$, and $\mathcal{V}_v^r \cap \mathcal{T}_t^r = \oslash$, thereby guaranteeing that each set is independent of the others. Furthermore, the union of these datasets for any given resample r encompasses the entirety of the original dataset $\mathcal{D}_n$, ensuring comprehensive coverage and utilization of the available data for model training, validation, and testing.

In order to select the best combination of hyperparameters $\mathcal{H}$ in a set of possible ones $\mathfrak{H} = \{\mathcal{H}_1, \mathcal{H}_2, \cdots\}$ for the algorithm $\mathscr{A}_\mathcal{H}$ or, in other words, to perform the MS phase, the following procedure has to be applied

$$\mathcal{H}^* : \quad \arg\min_{\mathcal{H}\in\mathfrak{H}} \sum_{r=1}^{n_r} \mathtt{M}(\mathscr{A}_\mathcal{H}(\mathcal{L}_l^r), \mathcal{V}_v^r), \tag{1}$$

where $\mathscr{A}_\mathcal{H}(\mathcal{L}_l^r)$ is a model h built with the algorithm $\mathscr{A}$ with its set of hyperparameters $\mathcal{H}$ and with the data $\mathcal{L}_l^r$, and where $\mathtt{M}(h, \mathcal{V}_v^r)$ is the desired metric for h computed using the data in $\mathcal{V}_v^r$. Since the data in $\mathcal{L}_l^r$ are independent of the data in $\mathcal{V}_v^r$, $\mathcal{H}^*$ should be the set of hyperparameters which allows achieving a small error on a data set that is independent of the training set.

Then, to evaluate the performance of the optimal model which is $h^* = \mathscr{A}_{\mathcal{H}^*}(\mathcal{D}_n)$ or, in other words, to perform the EE phase, the following procedure has to be applied:

$$\mathtt{M}(h^*) = \frac{1}{n_r} \sum_{r=1}^{n_r} \mathtt{M}(\mathscr{A}_{\mathcal{H}^*}(\mathcal{L}_l^r \cup \mathcal{V}_v^r), \mathcal{T}_t^r). \tag{2}$$

Since the data in $\mathcal{L}_l^r \cup \mathcal{V}_v^r$ are independent from the ones in $\mathcal{T}_t^r$, $\mathtt{M}(h^*)$ is an unbiased estimator of the true performance, measured with the metric $\mathtt{M}$, of the final model [37].

In this study, the metric $\mathtt{M}$ employed to evaluate the model performance is the Mean Average Percentage Error (MAPE) [38]. This quantitative measure of accuracy provides a standardized way to assess the error magnitude across different datasets. Additionally, to complement this quantitative assessment, a qualitative metric will also be utilized: the scatter plot of actual versus predicted values [39]. This graphical representation offers an intuitive means to visually inspect the model predictive accuracy and the relationship between actual and predicted outcomes.

In addition to evaluating the model quality, this study will also assess the time required to generate predictions using both the original simulator and its surrogate, as developed in the aforementioned section. It is imperative to clarify that within the scope of this research, the primary concern is not the duration necessary for the construction of the model. Rather, the focus is on the prediction latency, given that these predictions are integral to real-time traffic optimization scenarios.

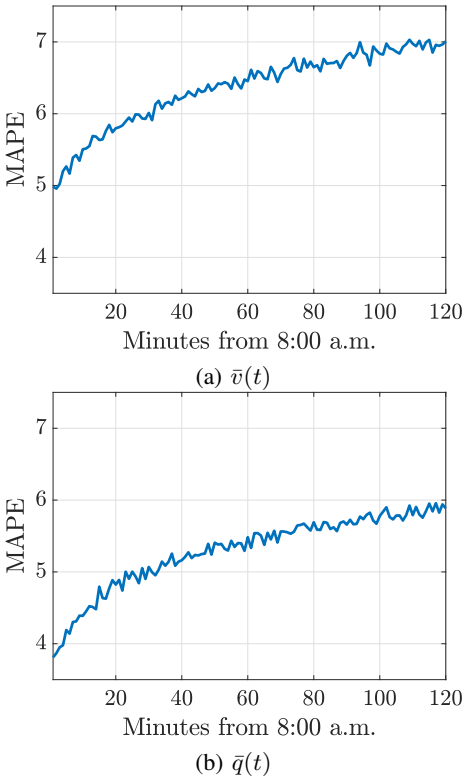
The final challenge addressed in this study pertains to the issue identified in Problem (B), specifically, reducing the requisite number of sensors for accurate prediction. The literature presents various methodologies for addressing this challenge [24], [25]. In this work, we employ the backward elimination technique [24], [25]. The backward elimination procedure initiates with the full set of sensor data, systematically removing one sensor at a time. This step is predicated on evaluating each sensor's impact on the overall performance metrics. The sensor whose absence least affects performance is permanently excluded from the dataset. Subsequently, the procedure is reapplied iteratively to the reduced set of sensors. This process continues until the remaining sensors achieve a performance metric that meets or exceeds the predefined threshold, in this instance, a MAPE of 10% or less. Although this approach is recognized for its computational load, it remains a viable solution within the context of our study due to the manageable scale of computational resources required.

IV. Experimental Results

In this section, we present the results obtained by applying the methods proposed in Section III to solve the problem described in Section II using the data referred to the full-scale urban network of Rapallo (332 arcs, 2076 scenarios, 1-minute discretisation). We will start with the results on Problem (A) in Section IV-A and proceed with the results on Problem (B) in Section IV-B.

A. Problem (A)

Regarding Problem (A), we report in Table II the MAPE (mean and standard deviation) for each algorithm (RR, KRR, RF, XGBoost, SNN, and GCN) in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for $t = t+\delta^t$ (shorter forecasting horizon), for $t = t+\delta^+$ (longer forecasting horizon), and averaged over $t \in \Delta^+$ (i.e., $\bar{v}(\bar{t})$ and $\bar{q}(\bar{t})$). Table III reports the time needed to make the forecasting ($v_i(t)$ and $q_i(t)$ for all $i \in \mathcal{I}_s$ and $t \in \Delta^+$) using the simulator AIMSUN NEXT (see Section II) or its surrogate based on the different algorithms (RR, KRR, RF, XGBoost, SNN, and GCN). Figure 3 reports the MAPE for the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for the different $t \in \Delta^+$. Figure 4 reports the Scatter Plot for the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s$ for $t = t+\delta^t$ (shorter forecasting horizon), for $t = t+\delta^+$ (longer forecasting horizon), and for $t \in \Delta^+$.



(a) $\bar{v}(t)$

(b) $\bar{q}(t)$

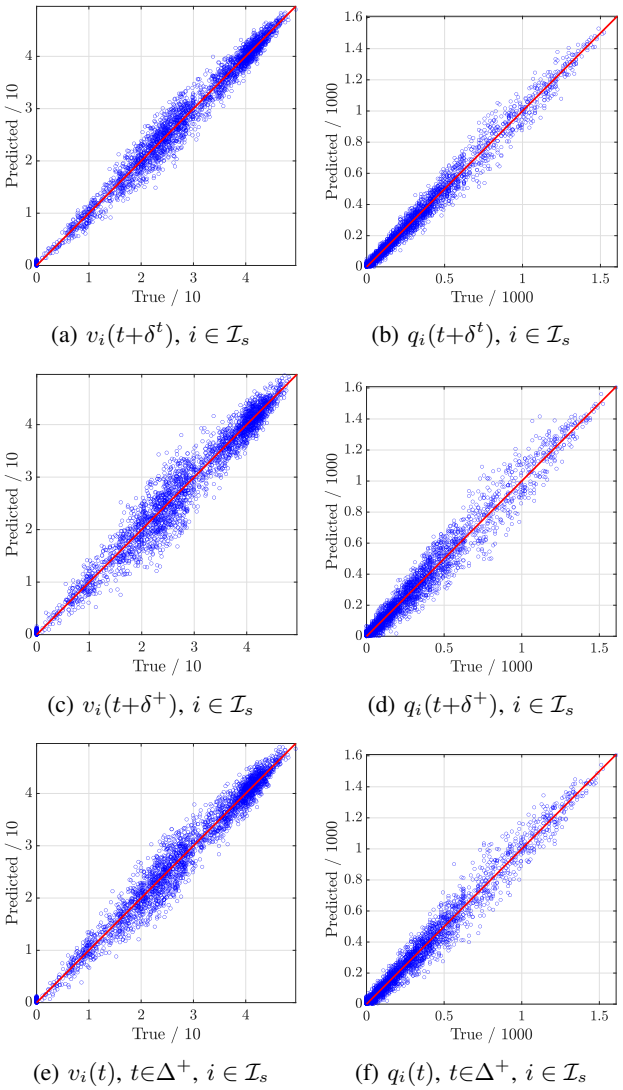
Fig. 3: Problem (A) - MAPE for the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for the different $t \in \Delta^+$.

From Tables II and III and Figures 3 and 4 it is possible to derive a series of observations.

All algorithms show strong performance on the task, with error rates around $5-6\%$. The RF algorithm outperforms the others, while the GCN has the lowest performance. This outcome aligns with two key observations. Firstly, the considered graph topology, representing the same city, remains constant, diminishing the need for a model like GCN that primarily enhances network topology representation. Secondly, the dataset limited size may not be sufficient for effectively training a deep learning model like GCN.

TABLE II: Problem (A) - MAPE (mean and standard deviation) for each algorithm in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for $t = t+\delta^t$, for $t = t+\delta^+$, and averaged over $t \in \Delta^+$ (i.e., $\bar{v}(\bar{t})$ and $\bar{q}(\bar{t})$).

Alg.	$\bar{v}(t+\delta^t)$	$\bar{v}(t+\delta^+)$	$\bar{v}(\bar{t})$	$\bar{q}(t+\delta^t)$	$\bar{q}(t+\delta^+)$	$\bar{q}(\bar{t})$
RR	5.1 ± 0.7	7.0 ± 1.0	6.0 ± 0.9	4.1 ± 0.6	6.2 ± 0.9	5.2 ± 0.7
KRR	5.4 ± 1.3	7.2 ± 1.7	6.3 ± 1.5	4.3 ± 1.0	6.3 ± 1.5	5.3 ± 1.2
RF	5.0 ± 1.0	7.0 ± 1.5	6.0 ± 1.2	3.8 ± 0.8	5.9 ± 1.2	4.8 ± 1.0
XGBoost	5.0 ± 1.2	7.3 ± 1.7	6.1 ± 1.5	4.1 ± 1.0	6.2 ± 1.5	5.2 ± 1.2
SNN	5.7 ± 2.2	7.7 ± 3.0	6.7 ± 2.6	4.6 ± 1.8	6.7 ± 2.6	5.7 ± 2.2
GCN	8.8 ± 3.3	10.4 ± 3.9	9.6 ± 3.6	7.5 ± 2.8	9.8 ± 3.6	8.6 ± 3.1



(a) $v_i(t+\delta^t)$, $i \in \mathcal{I}_s$ (b) $q_i(t+\delta^t)$, $i \in \mathcal{I}_s$

(c) $v_i(t+\delta^+)$, $i \in \mathcal{I}_s$ (d) $q_i(t+\delta^+)$, $i \in \mathcal{I}_s$

(e) $v_i(t)$, $t \in \Delta^+$, $i \in \mathcal{I}_s$ (f) $q_i(t)$, $t \in \Delta^+$, $i \in \mathcal{I}_s$

Fig. 4: Problem (A) - Scatter Plot for the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s$ for $t = t+\delta^t$, for $t = t+\delta^+$, and for $t \in \Delta^+$.

TABLE III: Problem (A) - Time needed to make the forecasting ($v_i(t)$ and $q_i(t)$ for all $i \in \mathcal{I}_s$ and $t \in \Delta^+$) using the simulator AIMSUN NEXT (see Section II) or its surrogate based on the different algorithms.

Method		Prediction Time
Simulator AIMSUN NEXT		23 ± 2 [s]
Surrogate	RR	7 ± 1 [ms]
	KRR	35 ± 7 [ms]
	RF	11 ± 5 [ms]
	XGBoost	12 ± 6 [ms]
	SNN	27 ± 9 [ms]
	GCN	19 ± 6 [ms]

Additionally, it is worth noting that the performance across algorithms is similar for both speed and flow prediction tasks.

The RR is the fastest predictor, but the other models have manageable speeds with no particular issues for optimization. Note that the speedup of the surrogate with respect to the simulator is huge (greater than 3 order of magnitude).

As expected, surrogate performance decreases as the forecast horizon increases. Nevertheless, note that, in the largest horizon, errors are still below $6-7\%$ (see Figures 3).

B. Problem (B)

For Problem (B) we first report in Figure 5 the result of the backward elimination namely, on the x-axis the number of removed sensors ($|\mathcal{I}_s \setminus \mathcal{I}_m|$) and in the y-axis the MAPE of the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over v and q, the roads and each allowed running direction $i \in \mathcal{I}_s$ and over $t \in \Delta^+$.

From Figure 5 it is possible to note that, as expected, removing sensors decreased our ability to make predictions but not remarkably. In fact, we can reduce the number of sensors to $|\mathcal{I}_m| = 7$ and still have an error below 10%. This is a quite important result since, in practical applications, there is no possibility to actually sensorize each road and each allowed running direction. Instead, the number of sensors that the model requires to achieve error below 10%, i.e., $|\mathcal{I}_m| = 7$, is totally manageable in practical applications.

Then, with a fixed $\mathcal{I}_m$, we report in Figure 6 (i.e., the counterpart Figure 3) the MAPE for the best algorithm

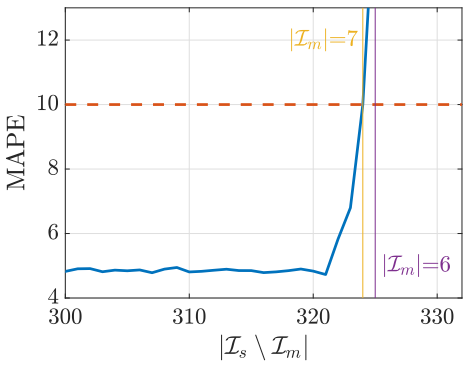


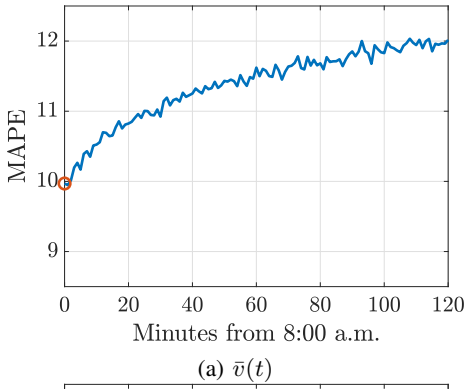
Fig. 5: Problem (B) - Result of the backward elimination, on the x-axis the number of removed sensors ($|\mathcal{I}_s \setminus \mathcal{I}_m|$) and in the y-axis the MAPE of the best algorithm according to Table II (i.e., RF) in predicting speed $v_i(t)$ and flow $q_i(t)$ averaged over v and q, the roads and each allowed running direction $i \in \mathcal{I}_s$ and over $t \in \Delta^+$.

according to Table II (i.e., RF) in predicting (Problem (B).(i) - nowcasting) speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s \setminus \mathcal{I}_m$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for $t = t_0$ and (Problem (B).(ii) - forecasting) speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for the different $t \in \Delta^+$. Figure 7 reports the Scatter Plot for the best algorithm according to Table II (i.e., RF) in predicting (Problem (B).(i) - nowcasting) speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s \setminus \mathcal{I}_m$ for $t = t_0$ and (Problem (B).(ii) - forecasting) speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s$ for $t = t+\delta^t$ (shorter forecasting horizon) and for $t = t+\delta^+$ (longer forecasting horizon).

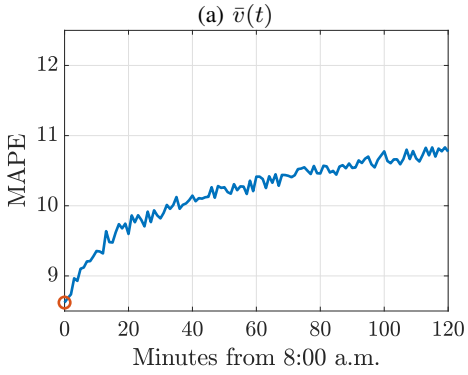
From Figures 6 and 7 it is possible to note that reducing the number of sensors to $\mathcal{I}_m$ do not actually compromises our ability to make effective predictions for traffic systems, i.e. with an acceptable error for practical applications [23].

V. Conclusions

In this paper a surrogate model of the traffic simulator AIMSUN NEXT has been investigated considering the simulation results of the city of Rapallo, during the morning peak. In particular, the surrogate has been used for solving two main problems, the prediction of speeds and flows over the network based on the same measurements in a previous time period, and the definition of the minimum number of sensors and their location to guarantee speed and flow predictions with an acceptable error, i.e. lower than 10%. Different state-of-the-art ML algorithms (RR, KRR, RF, XGBoost, SNN, and GCN) have been tested and compared, and the one showing the best performance has been RF, characterized by the lowest MAPE values and the shortest computational times. Such promising results encourage us to develop future works in this direction for surrogating traffic simulators to predict other estimated quantities, such as travel times and traffic emissions, that are provided by simulators and hard to measure in practice.



(a) $\bar{v}(t)$



(b) $\bar{q}(t)$

Fig. 6: Problem (B) - MAPE for the best algorithm according to Table II (i.e., RF) in predicting (Problem (B).(i) - nowcasting) speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s \setminus \mathcal{I}_m$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for $t = t_0$ and (Problem (B).(ii) - forecasting) speed $v_i(t)$ and flow $q_i(t)$ averaged over the roads and each allowed running direction $i \in \mathcal{I}_s$ (i.e., $\bar{v}(t)$ and $\bar{q}(t)$), for the different $t \in \Delta^+$.

Acknowledgments

This work is partially supported by Spoke 10 "Logistics and Freight" within the Italian PNRR National Centre for Sustainable Mobility (MOST), CUP I53C22000720001.

The authors would like to acknowledge the Municipality of Rapallo for sharing traffic data.

References

[1] A. Kesting and M. Treiber, *Traffic flow dynamics: data, models and simulation*. Springer-Verlag, Berlin Heidelberg, 2013.

[2] L. Elefteriadou, *An Introduction to Traffic Flow Theory*. Springer, 2014.

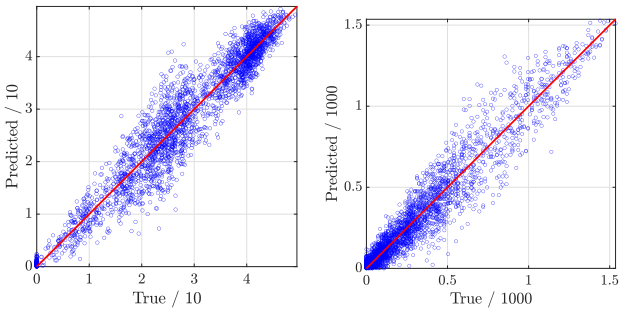
[3] A. Ferrara, S. Sacone, and S. Siri, "Microscopic and mesoscopic traffic models," *Freeway traffic modelling and control*, pp. 113–143, 2018.

[4] S. Siri, C. Pasquale, S. Sacone, and A. Ferrara, "Freeway traffic control: A survey," *Automatica*, vol. 130, p. 109655, 2021.

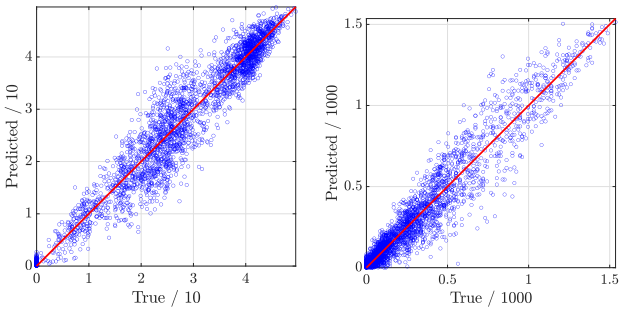
[5] J. Barceló, J. Casas, J. Ferrer, and D. García, "Modelling advanced transport telematic applications with microscopic simulators: The case of aimsun2," in *Traffic and Mobility: Simulation-Economics-Environment*, 1999.

[6] J. Casas, J. L. Ferrer, D. García, J. Perarnau, and A. Torday, "Traffic simulation with aimsun," *Fundamentals of traffic simulation*, pp. 173–232, 2010.

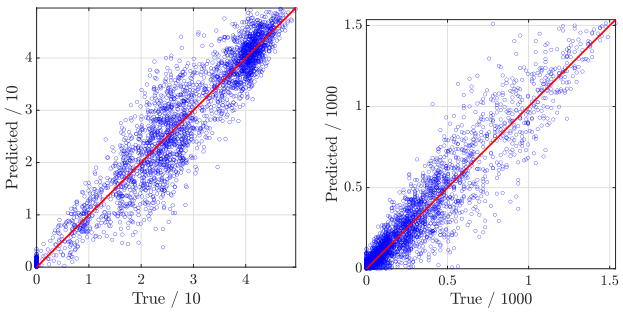
[7] D. Krajzewicz, "Traffic simulation with sumo-simulation of urban mobility," *Fundamentals of traffic simulation*, pp. 269–293, 2010.



(a) $v_i(t_0)$, $i \in \mathcal{I}_s \setminus \mathcal{I}_m$ (b) $q_i(t_0)$, $i \in \mathcal{I}_s \setminus \mathcal{I}_m$



(c) $v_i(t+\delta^t)$, $i \in \mathcal{I}_s$ (d) $q_i(t+\delta^t)$, $i \in \mathcal{I}_s$



(e) $v_i(t+\delta^+)$, $i \in \mathcal{I}_s$ (f) $q_i(t+\delta^+)$, $i \in \mathcal{I}_s$

Fig. 7: Problem (B) - Scatter Plot for the best algorithm according to Table II (i.e., RF) in predicting (Problem (B).(i) - nowcasting) speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s \setminus \mathcal{I}_m$ for $t = t_0$ and (Problem (B).(ii) - forecasting) speed $v_i(t)$ and flow $q_i(t)$ with $i \in \mathcal{I}_s$ for $t = t+\delta^t$ and for $t = t+\delta^+$.

[8] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *IEEE International Conference on Intelligent Transportation Systems*, 2018.

[9] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," *Fundamentals of traffic simulation*, pp. 63–93, 2010.

[10] M. Fellendorf, K. Nökel, and N. Handke, "Visum-online-traffic management for the expo 2000 based on a traffic model," *Traffic Technology International*, 2000.

[11] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on intelligent transportation systems*, vol. 6, no. 3, pp. 314–325, 2005.

[12] Q. Chao, H. Bi, W. Li, T. Mao, Z. Wang, M. C. Lin, and Z. Deng, "A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving," in *Computer Graphics Forum*, vol. 39, no. 1, 2020, pp. 287–308.

[13] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transportation Research Part C*, vol. 34, pp. 89–107, 2013.

[14] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual reviews in control*, vol. 43, pp. 128–151, 2017.

[15] A. J. Huang and S. Agarwal, "Physics-informed deep learning for traffic state estimation: Illustrations with lwr and ctm models," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 503–518, 2022.

[16] K. Binjaku, E. K. Meçe, C. Pasquale, and S. Sacone, "Control oriented freeway traffic modelling by physics-regularized machine learning," in *IEEE International Conference on Intelligent Transportation Systems*, 2023.

[17] E. I. Vlahogianni, "Optimization of traffic forecasting: Intelligent surrogate modeling," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 14–23, 2015.

[18] R. Fan, H. Yu, P. Liu, and W. Wang, "Using vissim simulation model and surrogate safety assessment model for estimating field measured traffic conflicts at freeway merge areas," *IET Intelligent Transport Systems*, vol. 7, no. 1, pp. 68–77, 2013.

[19] C. Katrakazas, M. Quddus, and W. H. Chen, "A simulation study of predicting real-time conflict-prone traffic conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3196–3207, 2017.

[20] Q. Cheng, S. Wang, Z. Liu, and Y. Yuan, "Surrogate-based simulation optimization approach for day-to-day dynamics model calibration with real data," *Transportation Research Part C*, vol. 105, pp. 422–438, 2019.

[21] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[22] C. C. Aggarwal, *Neural networks and deep learning: a textbook*. Spinger, 2023.

[23] A. Spiliopoulou, I. Papamichail, M. Papageorgiou, Y. Tyrinopoulos, and J. Chrysoulakis, "Macroscopic traffic flow model calibration using different optimization algorithms," *Operational Research*, vol. 17, pp. 145–164, 2017.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[25] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys*, vol. 50, no. 6, pp. 1–45, 2017.

[26] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in *Soft computing and industry*, 2002.

[27] G. C. McDonald, "Ridge regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.

[28] V. Vovk, "Kernel ridge regression," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, 2013.

[29] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[30] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[31] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *AMC SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[32] Z. H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[33] C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[34] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," *Neural Networks*, vol. 129, pp. 203–221, 2020.

[35] S. S. Keerthi and C. J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] L. Oneto, *Model Selection and Error Estimation in a Nutshell*. Springer, 2020.

[38] M. Z. Naser and A. H. Alavi, "Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences," *Architecture, Structures and Construction*, pp. 1–19, 2021.

[39] K. L. Sainani, "The value of scatter plots," *PM&R*, vol. 8, no. 12, pp. 1213–1217, 2016.