A generic framework for forecasting short-term traffic conditions on urban highways

Seif-Eddine Attoui, Maroua Meddeb

A generic framework for forecasting short-term traffic conditions on urban highways

Seif-Eddine Attoui
IRT-SystemX
Email: seifeddine.attoui@irt-systemx.fr

Maroua Meddeb
IRT-SystemX
Email: maroua.meddeb@irt-systemx.f

***Abstract*—With the emergence of Connected and Smart Cities, the need to predict traffic conditions has led to the development of a large variety of forecasting algorithms. In spite of various research efforts, the choice of models and techniques strongly depends on the use case, the highway infrastructure as well as the provided dataset. This study is launched as part of a project which aims to design an Intelligent Transport System (ITS) dedicated to highway supervisors to regulate traffic. This system needs to be supplied by continuous, real-time forecasting of short-term traffic congestions in order to make decisions accordingly. In this paper, we propose a general framework that, first, performs different data preprocessing techniques to improve data quality, and second, provides real-time multiple horizons predictions. Our framework uses different models combining Machine learning and Deep learning algorithms. Experiments results confirmed the necessity of the data preprocessing step, especially with highly dynamic data and heterogeneous mobility contexts. In addition, our methodology is tested in a real case study and shows very encouraging results.**

***Index Terms*—Intelligent Transportation Systems, Traffic forecasting, Short-term, Data preprocessing, Data balancing**

I. Introduction

Since the 1950s, with the increase of motorists and the concentration of residents in city areas, the congestion of urban transportation infrastructure has increased significantly [1]; causing economic, ecological and social problems related mainly to wasted fuel consumption, pollutant emissions and travel time cost. For instance, when a traffic congestion occurs and is not timely handled, it may cause traffic paralysis. Therefore, effective forecasting models for traffic congestion should be designed in order to prevent the formation of congestions and hence increase the efficiency of the road network. Under the tremendous pressure on the decision-makers to settle these problems, great efforts have been made in recent decades, especially in traffic flow management. The latter has been continuously improved with the expansion of data provided by sensors throughout the transport infrastructure and with the development of data analysis methods. These advances have led to a new concept of traffic flow management system called Intelligent Transportation System (ITS) [2].

The ITSs supervise the traffic network by assisting in route planning, guiding vehicle dispatch, and mitigating traffic congestion. This system has flourished with the application of statistical and machine learning algorithms to anticipate traffic conditions, especially short-term ones. Since then, short-term traffic forecasting has become an important part of most Intelligent Transportation Systems. Indeed, short-term traffic forecasting enables ITS operators to have a global visibility on what the traffic network will be in the short term. Besides, it allows them to take real-time actions depending on the forecasted network; for example, updating traffic lights according to network forecasts to avoid traffic jams [3].

Short-term traffic forecasting is one of the most dynamic research areas with huge published literature. In the latter, many approaches have been widely exploited, ranging from statistical models to machine and deep learning techniques, where promising results have been noticed. However, most of the research work has often concentrated on the models themselves and neglected other important aspects like preprocessing data to improve models' performances.

In this paper, different from existing work, we have designed a global framework to tackle short-term congestion forecasting with different time horizons. Our proposal is made up of two components: A training module and a real time application module. The former aims to carefully preprocess the input data with various techniques, and then train various forecast models, which take advantage of the processed data to produce precise estimates. The latter module, which is the real time application module, is used to guarantee the availability of prediction at all times.

Our contributions can be summarized as follows :

- We propose a global framework for short-term traffic state forecasting and we investigate several predictive models ranging from machine and deep learning approaches. In particular, we enhance the performance of our designed models with careful data preprocessing and feature engineering.
- We propose a data balancing approach. To the best of our knowledge, in the context of traffic forecasting, this is the first study that deals with unbalanced data in the preprocessing phase by using a down-sampling approach.
- We validate and evaluate our proposed approach through extensive experiments in a real case study, using highly dynamic and heterogeneous data. The performance of the designed models have been investigated on different time scales. In addition, the proposed framework is integrated into a real-time ITS and used to regulate traffic by the Lyon Metropolis.
- Our results highlight different engineering insights. Among them, the use of density data as an input for

long horizon forecasts enhance significantly the quality of the estimates, which can be very useful as generally the performance of models decreases as forecast horizon increases.

The remainder of this paper is organized as following; in Section 2, we start by presenting the existing literature in terms of the different techniques investigated. Afterwards, in Section 3, we formulate the congestion state forecasting problem properly. Then, in Section 4, the proposed global methodology is explained with details of its different parts. Section 5, discusses the different experimental evaluations. Finally, we conclude the paper and draw some insights in Section 6.

II. Related Work

A large body of the literature studies in the transportation research lies in traffic congestion forecasting [4], which is a crucial application in intelligent transportation systems [5]. Several studies have reviewed the different techniques used for traffic forecasting [3] [2] [6]. One essential line of work touches upon short-term prediction, with focus on road features such as speed [7], flow [8] and travel time [9]. In what follows, we divide the literature according to the investigated approaches into two categories, namely parametric methods and non parametric techniques [10].

Techniques that simplify the learned function to a known form are called parametric methods. Among these approaches auto regressive integrated moving average (ARIMA) [11] is one of the pioneering techniques applied to process traffic sequential data. Since then, variants of ARIMA such as seasonal ARIMA (SARIMA) models [12], kalman filter models [13] and other models [14] were adopted to perform traffic forecasting. Despite the fact that these techniques can perform sometimes relatively good, they require traffic data to meet some theoretical or physical assumptions, which simplify the learning process in one hand but limit what it can be learned in another hand. Furthermore, some studies point out that the performances of these models dropped significantly with the increase of time horizons [15].

Later, to cope with the limitations of the parametric methods, researchers start to consider non parametric methods such as support vector machine (SVM), k nearest neighbours (KNN) and artificial neural network (ANN). Many studies have investigated SVM and its different variants as support vector regression (SVR) and online SVM to perform traffic prediction and have claimed their ability to process dynamic traffic data [16]. Moreover, both univariate and multivariate KNN have been widely applied to traffic flows and speeds forecasting [17]. Some other work showed good performances when combining different machine learning technique as in [18], where authors proposed a hybrid short-term traffic flow forecasting model that combines KNN to SVM. The developed model performed relatively better in comparison to KNN, SVR and back-propagation neural network (BPNN). Nevertheless, these mentioned techniques require prior knowledge and careful feature engineering to achieve better scores. ANN models on their sides were widely applied in the traffic forecasting field due to their ability to work with multidimensional data with few or no feature engineering as well as their capacity to model the non linear relationship between inputs and outputs to provide generalized solutions [11]. For instance, in [19], authors removed unimportant fluctuations from input flow signal using a wavelet transformation and then trained an ANN on historical data to forecast traffic flow in multiple highways at different locations. Despite the undeniable advantages of ANNs, their results were not completely satisfactory due to their shallow depth architecture. Therefore, researchers shift their research axe toward deep learning architectures [20]. In particular, recurrent neural networks (RNNs) especially long short term memory (LSTMs) and its different variants, have gained popularity in these recent years to process temporal time series sequence and extract relevant temporal features from big amounts of data, in different fields. In traffic field, they have been widely used for speed, flow and congestion prediction and have shown remarkable results [21] [22]. In order to capture the spatial dependencies from the traffic input data, some work have adopted convolution neural networks (CNNs) by exploiting traffic images [23] [24]. Furthermore, efforts have been put on hybrid model to make use of both temporal and spatial features [25]. In [26], Yu et al. designed a deep learning model that combines CNNs with LSTMs to extract spatial and temporal dependencies of different roads to predict traffic speed. Results have demonstrated the superiority of the proposed model for both long and short term forecasting over other deep leaning based algorithms.

However, most previous studies focused on predicting traffic variables such as speed, flow and time travel on one or multiple road segments. To the best of our knowledge, few works considered the exploitation of the different traffic properties to forecast the presence or the absence of a congestion. One potential reason of the lack of such studies of traffic congestion prediction is the difficulty to find large data sets with high spatial and temporal granularity [20] [27] [28].

Another aspect that is worth mentioning, regarding the previous literature, is that studies mostly focused on model development and fine tuning, with few attention to data preparation. Careful data pre-processing can significantly improve their quality, thus enhancing model performance. In this paper we attempt to address the different literature gaps that were identified.

III. Problem formulation

Our objective is to predict future traffic conditions in real-time. More precisely, congested states or smooth states given previously observed data from loop sensors on a highway network. We formally express the observed dataset as a spatio-temporal sequence of data points $M = \{M_1, M_2, \ldots, M_T\}$, where M_t is a snapshot of the sensor measurements at time t in a part of the highway. i.e.

$$M_t = \begin{bmatrix} m_t^{(1,1)} & \dots & m_t^{(1,j)} \\ \vdots & \vdots & \vdots \\ m_t^{(i,1)} & \dots & m_t^{(i,j)} \end{bmatrix} \quad (1)$$

Where $m_t^{(i,j)}$ is the traffic measurement of the sensor (i, j) at time t. We aim to predict the most likely K-step of traffic states, given previous M observations. This means solving :

$$\hat{Y}_{t+1}, \hat{Y}_{t+2}, ..., \hat{Y}_{t+K} = \underset{Y_{t+1}, Y_{t+2}, ..., Y_{t+K}}{\arg\max} p(Y_{t+1}, Y_{t+2}, ..., Y_{t+K} | M_{t-j+1}, M_{t-j+2}, ..., M_t) \quad (2)$$

Where $\hat{Y}_{t+1} \in \{0,\ 1\}$ is the predicted state with its two possible values: congested state and smooth state, $Y_{t+1} \in \{0,\ 1\}$ is the real state (ground truth) and j is the time lag of previous observations.

IV. Methodology

In this section, we present our general framework and explain each of its components. The goal of this framework is to improve data quality and produce more accurate real-time forecasts, by performing different data preprocessing techniques, such as outlier handling, data smoothing, data balancing, etc. and forecasting traffic congestion over multiple horizons using various data-driven models: low horizons to predict traffic conditions as early as possible and act accordingly, and high horizons to have a full visibility on traffic conditions evolution.

Our framework consists of two main components, namely the training component and the real-time application component. The training component aims to train our models with a high-quality data, by applying several data preprocessing techniques, like data balancing, which plays an essential role to enhance the performance of our framework compared to state of the art work. Within the same component, we define a prediction strategy based on Machine learning and Deep learning algorithms. This strategy aims to design a prediction model for each sensor and each time horizon. Recent empirical works have proved the effectiveness of this strategy compared to multi-output prediction [29] [30]. The second component's goal is to apply this strategy to predict the highway traffic states in real-time, by running prediction of all trained models at once every 1 minute, then visualize the results in a highway map as presented in Fig. 1. We also apply a data imputation technique within this component to ensure the availability of predictions in real-time.

A. Data preprocessing

Data preprocessing aims to improve data quality, by cleaning up incomplete, noisy and inconsistent data, thereby improving the quality of predictions. To that end, we apply several data preprocessing techniques to our raw data, such as handling outliers, temporal aggregation, data smoothing, and data balancing in order to choose a suitable dataset in the learning phase.

1) Handling outliers: An outlier is an observation that deviates so much from other observations, causing unusual values and events in the dataset, thus leading to test failures or distortion of the actual results. In the field of traffic transportation, outliers can be caused by the malfunction of the loop sensors or by the excessive speed of certain individuals. In our approach, outliers are considered as values that exceed a certain threshold. Concretely, we define an algorithm that compares each measured value to a set of configurable thresholds. For example, for the speed measurement V we define a threshold V_{max} where $V < V_{max}$ & $V > 0$. We do the same for other measurement variables (i.e. flow Q, time travel TT, etc.). After that, outliers are replaced with neighboring sensor values. In case neighboring sensors are missing or detected as outliers, a Null value is set.

2) Data temporal aggregation: Temporal aggregation eliminates the variance heterogeneity of the traffic dataset, reduces the sensitivity of traffic changes, and leads to a smoothed traffic variation. Therefore, it improves the results of the prediction models. However, there is no solid approach for selecting the appropriate level of aggregation. For this reason, we perform several experiments with different datasets and different prediction models to determine an optimal aggregation interval. The results show that aggregation levels from 3 min to 6 min gave the best prediction results in the context of machine learning and deep Learning models. The obtained results are compatible with several experimentation studies, such as in [31].

3) Data smoothing: Data smoothing is used to reduce the irregularities and the singularities of the dataset, by approximating a function that attempts to capture important patterns in the data, while omitting noise and other fine-scale structures, thus, reducing what can be considered as disturbance or measurement noise. In our approach, we propose to use the Savitzky-Golay filter, which is a non-linear data smoother with widespread application in various scientific fields. The algorithm uses the linear least squares method to fit a successive subset of adjacent data points with a low degree polynomial. Two parameters must be considered when applying this algorithm: window size m and polynomial degree n. In order to avoid any distortion of the data signal trend, we recommend using a window size smaller than 8 ie. $n < m < 8$.

4) Data balancing: Unbalanced data is a common phenomenon in the traffic field, where data has an unequal distribution of observations' number corresponding to each class. This imbalance in the data can lead to a significant bias in the prediction results in favor of the majority class. Therefore, it can cause a poor classification rates in the minor classes and an extreme bias towards the majority class. In fact, minority classes tend to be overlooked by statistical, machine learning or even deep learning models. Indeed, data-driven models attempts to reduce the general errors using a loss function. The latter is usually based on all observations, and since the majority class represents the majority of observations, the loss function tries to reduce the majority class errors, thereby neglecting the minority classes.

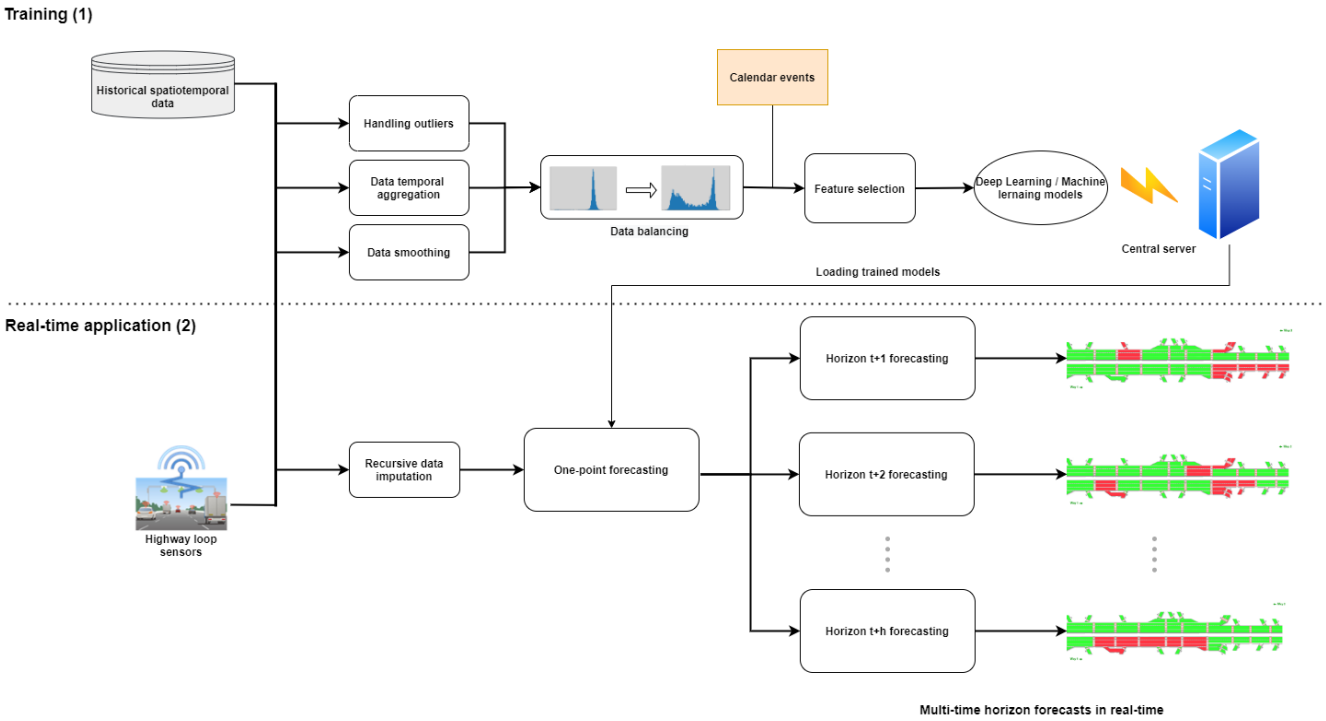


Fig. 1: Our multi-horizon framework for congestion forecasting

Two main approaches have been proposed in the literature for data balancing: minority class up-sampling and majority class down-sampling. The minority class up-sampling consists of adding simulated or duplicated observations to the minority class, while the majority class down-sampling consists of removing irrelevant observation from the majority class until the balance of all classes. Due to the complexity of simulating a good observation, we use, in our framework, the majority class down-sampling approach, in which we develop a technique to extract relevant information from the dataset, and considering the time-series characteristics of data. Therefore, we get at the end a balanced dataset. To our knowledge, in the context of traffic forecasting, this is the first study to deal with unbalanced data in the preprocessing phase by using a down-sampling approach.

Basically, the proposed technique aims to select only a subset of data with interesting traffic phenomena. That said, the appearance of congestion state. To that end, we define a window to retrieve this subset. This window slides over the dataset using three main parameters: α, β and θ . The parameter α (respectively, β) represents the time before (respectively, after) the appearance (respectively, disappear) of the congestion. In fact, the congestion phenomenon generally appears 10 min to 30 min before the actual transition of states (i.e. from the smooth state to the congested state), and disappears up to 10 min later. Using the α and β parameters, the sliding window defines what data is to be considered before and after the congestion phenomena. Finally, the θ parameter represents the congestion state threshold. Fig. 2 gives an example of the data subset considered by the sliding window (red rectangles).

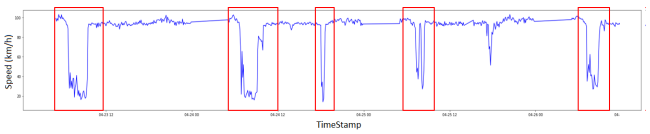


Fig. 2: An example of applying the sliding window technique to a 7-day speed dataset. The red rectangle represents the data of the congestion phenomenon under consideration.

5) Features selection: We propose to use the recursive feature elimination algorithm, which recursively eliminates the least important features at each prediction step until a small set of the most important ones is obtained. Therefore, at each entry of the algorithm, all traffic variables (i.e. traffic flow Q, speed V and density K) with a J_{max} previous observations are used, and by the end of the processing, the obtained set is ranked to select the important variables and the optimal time lag $j_{optimal}$.

B. Multi-horizon traffic forecasting

Having an accurate traffic congestion prediction is a challenging problem, especially for real-time applications where the uncertainty of predictions cannot be verified. Recent researches have focused on model's development and tuning their parameters, while few works have investigated an accurate strategy for real-time forecasting. Within this framework, we propose a real-time forecasting strategy based on multi-horizon forecasting to verify the predictions' uncertainty. The

proposed strategy aims to use for each loop sensor a forecasting model to enhance the prediction results. In addition, higher horizons are used to have a global visibility of the traffic state. Thus, confirming prediction results. Take as an example forecasting the traffic state at horizon t_{+6} : at each loop sensor, we develop 3 predictions models, one for the t_{+6} horizon, and two for higher horizons t_{+9} and t_{+12}. Then, we predict at each minute the traffic state. By visualizing the three results side by side, as shown in Fig. 1, and with the assumption that congestion lasts at least 10 minutes, we can clearly remark a potential inconsistency in the results.

In this part, various forecasting models including machine and deep learning approaches were designed and tunned to perform the multi-horizon prediction. The latter models will be further detailed in the experiment part.

C. Data imputation

Loop sensors are not very reliable. Sometimes they fail or give erroneous measurements, leading to datasets with missing values. It is known that these missing values negatively affect the accuracy of forecasts. Therefore, an online data imputation strategy is essential to ensure the availability of the real-time forecasting system. To that end, we propose to replace the missing value with the nearest neighboring sensor value. For larger gaps of missing values, we apply the iterative imputer algorithm based on the KNN model. The algorithm estimates all other features in an unsupervised manner by modeling each feature with missing values as a function of other features in a cyclic manner. Note that we use this strategy only on the real-time application phase. For the training phase, we drop all missing values to maintain data quality.

V. Validation

In this section, we present the results of the experimental evaluation of our framework. Before diving deeper into this latter, we first introduce the real-world use-case to which our framework is applied, as part of the development of an ITS system for managing reserved carpooling lanes. We then present the dataset used. Finally, we present the results of the preprocessing performed and the predictions models.

A. Lyon reserved-lanes case study

This work is part of the redevelopment of the Lyon metropolitan section of the M6/M7 freeway into an urban boulevard. With an average of 120 000 vehicles per day, the highway is considered as a major axis in Lyon, because it serves as a gateway to the city and is mainly used for home-work trips (commuting). In addition, the highway connects Paris to the south of France, which also makes it an important route throughout France. Consequently, major traffic jams are always observed during holidays. We present, in Fig. 3, the transformation of the M6/M7 freeway.

Facing major traffic jam problems, especially on both sides of the tunnel, which is a recurrent, active bottleneck, the Metropole de Lyon have decided to reserve the left lane for carpooling (vehicle with at least 2 passengers) as well as

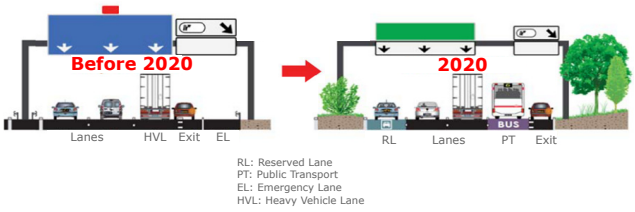


Fig. 3: M6/M7 lanes transformation

environmentally friendly vehicles and taxis. This lane is called Reserved Lane (RL). This initiative aims to encourage drivers to share their vehicles and so take advantage of a shorter travel time. This will reduce the number of vehicles and therefore reduce traffic congestion.

In order to guarantee an efficient and regular travel time for RL users, while limiting excessive travel time degradation on other lanes and avoiding congestion transfers to secondary networks, a dynamic activation of the RL is considered, depending on traffic conditions. In other words, if it is activated, the RL is used only for carpooling and in case of excessive traffic degradation, the RL must be deactivated and then used as a usual lane.

The main objective of this study is to anticipate a potential traffic degradation that could lead to lane deactivation. An ITS system is then designed within this project to provide recommendations to lane supervisors on the activation/deactivation of the RL. This system is based on short-term traffic congestion forecasting to decide on lane state. It is represented as a website application that displays the current traffic state of the highway M6/M7, the predicted state as well as the recommended action about the RL. Fig. 4 shows one of the highway sections. In this work, we apply the prediction methodology detailed in previous section to provide results displayed in this system.

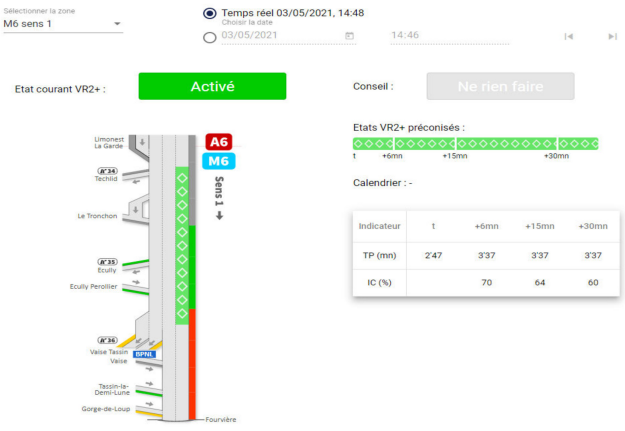


Fig. 4: Decision support website

B. Data description

We used in this work a real-world dataset collected from sensors placed along the M6/M7 urban highway in the heart

of Lyon city. As depicted in Fig. 5, the urban highway is made up of two sections: section M6 from point A to point B with a length of 11km, and section M7 from point B to point C with a length of 6km. In each section, a maximum speed of 90 km/h is authorized.

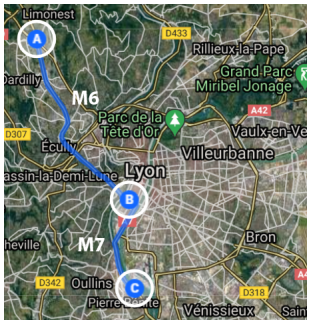


Fig. 5: Location of the M6-M7 highway in Lyon

The collected data is provided by regularly shifted loop sensors, with a distance of 500m to 1000m between two loops, as shown in Fig. 6. These sensors provide the average speed (V), the average flow (Q) and the occupancy rate (TT) per lane every 1 minute. Based on these measurements, we calculate other variables such as the average density (K). Then we aggregate all the variables into a 3-minute timestamp to reduce data noise. We have a one year of data available from 82 sensors placed along the M6-M7 highway, making a total of 175,200 observations.

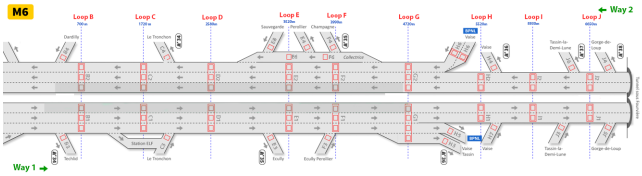


Fig. 6: Part of the M6 highway near Lyon city. Way-1 is the path leading to the city entrance, and way-2 is the path to leaving the city. Sensors are represented with small red squares. We call the group of sensors arranged vertically a loop. For example, loop B contains sensor B1, B2, B3 and B4. Sensors on the same loop in the same path and on different lanes are grouped together to reduce data noise. For example, sensor B1 and B2. Therefore, in this part of M6 highway, we have a total of 38 sensors.

We use the traffic Fundamental Diagram (FD) to analyze the capacity of the road network, in particular the traffic congestion thresholds. We therefore define the congested state either by the speed below 50 km/h, or by the traffic flow (Q) above 1200 vehicles/h, or by the travel time (TT) above 10min, or by density (K) above 50 vehicle/km.

Fig. 7 shows a visual distribution of our dataset in sensor J1. We plot the speed (V), travel time (TT), traffic flow (Q) and density (K) over a period of one day. We can notice from this figure:

- Data is very noisy, especially for density (k) and travel time (TT). Numerous empirical studies have shown that a noisy dataset can dramatically decrease accuracy and lead to poor prediction results. Thus, the application of a data smoothing method is essential in our dataset.
- Compared to other variables, speed (V) can be quickly adapted to state transitions. This means that the speed varies rapidly from a congested state to a smooth state, or from a smooth state to a congested state. For example, the transition from smooth to congested state between 6 a.m. and 7 a.m.
- We also notice that speed is more stable compared to other variables, especially in congested and smooth states. Take as an example the congested state between 9 a.m. and 10 a.m.: unlike other variables, speed varies slightly, but it is always lower than the congestion state threshold.

Based on these observations and what has been said previously, we only define the congested state with a speed below 50 km / h.

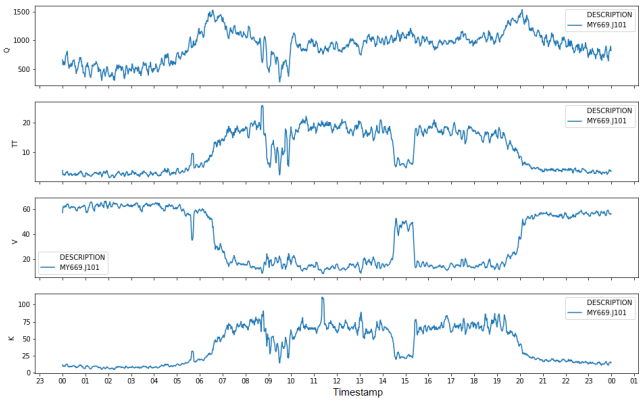
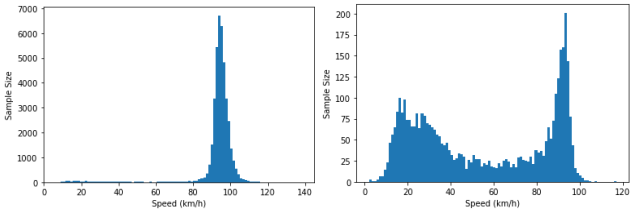


Fig. 7: The relationship between Q, V, TT and K

C. Performance evaluation

In this section, we conduct several experiments to evaluate our framework. As mentioned earlier, our framework has been integrated into an ITS system. Based on the traffic forecasts given by the framework, the ITS system provides recommendation on the activation/deactivation of the reserved lanes. To that end, we set up three forecasting horizons: a t_{+6min} horizon to forecast traffic conditions as soon as possible and take corresponding actions (activation/deactivation), and horizons t_{+15min} and t_{+30min} to have full visibility on the evolution of traffic congestion, and this should also be considered in the activation/deactivation decision. It is also worth to mention that we used regression models for the t_{+6min} forecasts, and classification models for the t_{+15min} and t_{+30min} forecasts. As mentioned in the related work section, the reason is that as the time horizon increases, the performance of the model decreases significantly, especially regression models. Therefore, we apply on the t_{+6min} regression values a threshold to classify the traffic conditions in the two states mentioned above.

1) Data preprocessing Results: In these experiments, we considered about 80% of the data for the training and 20% for the tests. As indicated in the methodology section, we deal with outliers by replacing them with neighboring sensors. We fix the maximum of closest sensors to 2. Therefore, if the two closest sensors are missing, we replace the outlier with Null. Then, we drop the missing values for the training set and impute them for the test set, as previously discussed in the imputation subsection. After that, we aggregate all data into 3-minute timestamp. As a result, we get a training set of 98,323 observations. Finally, we feed these latter into data balancing algorithm. We thus obtain a balanced training set of 4,599 observations. The results of this latter step is shown in Fig. 8.



(a) (b)

Fig. 8: Data distribution before and after the application of the data balancing algorithm. The distribution graph (a) shows the data before balancing. Congested states (speed below 50 km / h) represent approximately 4% of the total data (approximately 3,709 observations out of 98,323 total observations). The distribution graph (b) shows the data after balancing, where congested states now represent 45% of the total data (approximately 2,097 observations out of 4,599 total observations)

2) Compared prediction models: Several predictive models are considered in this part to validate our proposed framework. These models include Ada Boost, Random Forest, Extra Trees, Gradient Boosting, eXtreme Gradient Boosting (XGBoost), artificial neural network (ANN) and long short term memory (LSTM) network. In addition, these models are compared with a naive forecasting method in which previous observations are directly used as forecasts without any changes. On this study, we mainly focus on the M6-way1 section, more precisely on the 6 loop sensors located in the RL part. In fact, the activation/deactivation of the RLs is highly dependent on its traffic state. i.e. the traffic state from loop B1 to loop G1 as depicted in Fig. 6. We evaluate these models based on the recall and the f1-score metrics. Our goal here is to anticipate the first appearance of the congested state in order to activate the RL section as soon as possible. Since, the recall and the f1-score metrics cannot provide us with an accurate evaluation of the first appearance of the congested state, we use in addition to these two metrics another one in order to measure the prediction delay. We call this metric the prediction delay KPI, it measures the delay between the first real appearance of the congested state and the first prediction of this latter.

Loop	Recall (%)	F1-score (%)	Average prediction delay (min)	Max prediction delay (min)
B1	89 %	87 %	0,64 min	3 min
C1	95 %	95 %	1,16 min	3 min
D1	97 %	94 %	1,23 min	3 min
E1	97 %	97 %	1,63 min	3 min
F1	97 %	97 %	2,6 min	6 min
G1	97 %	97 %	3,26 min	6 min

TABLE I: Predictions performance at t_{+6min} using the XGBoost model at each loop sensor.

For what follows, we study the impact of the data preprocessing techniques on the framework performance. More precisely, we focus on the impact of the data balancing technique on the performance of the predictions quality of the forecasting models mentioned earlier, considering 3 time horizons: 6 min, 15 min and 30 min.

3) 6min-horizon congestion forecasting: In this experiment, we apply the regression paradigm to forecast traffic state at t_{+6min}. In fact, results of both classification and regression have been compared and it seems that the best performances have been obtained through regression. Indeed, due to the high adaptability of speed in state transitions (as shown in Fig. 7), speed prediction at lower horizons (usually lower than t_{+9min}) has better results than traffic classification. After computing regression, the predicted values are then classified based on the traffic condition threshold (i.e. congested state if the speed is less than 50 km / h and smooth state otherwise). For all the forecasting models the time lag was set to 12 min, and the speed (V) and the traffic flow (Q) were used as inputs.

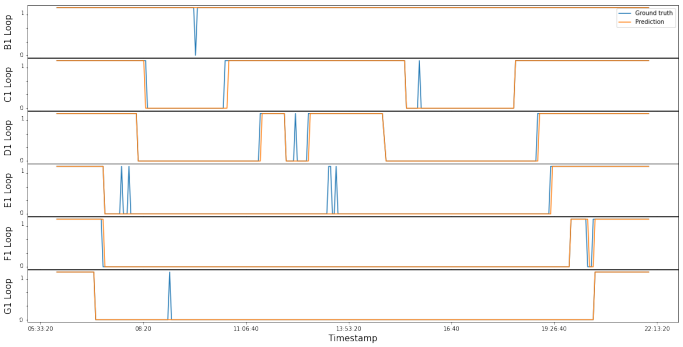


Fig. 9: Forecasting performance at t_{+6min} on 6 loop sensor using XGBoost. Blue lines represent real values while orange ones represent forecast values. Horizontal axis shows timestamps and vertical axis shows traffic state.

In Fig. 10, we show the averaged performance of the considered prediction models in terms of F1-score, Recall and our KPI to calculate the prediction delay. Results demonstrates the superiority of the LSTM model compared to the other models. Indeed, it obtained the lower prediction delay and the highest F1-score and recall. These scores are followed closely by Extra trees and gradient boosting methods. In Table 1, we investigate the performance of a chosen model e.g. XGBoost on the six sensors of the M6-way1 section. From the results

Data Balancing	Loop	Recall (%)	F1-score (%)	Average prediction delay (min)	Max prediction delay (min)
Before	B1	88%	87%	4.5 min	21 min
	C1	90%	88%	5.18 min	18 min
	D1	92%	91%	4.2 min	12 min
	E1	93%	93%	4.83 min	15 min
	F1	95%	93%	8.82 min	24 min
	G1	91%	92%	8.04min	18 min
After	B1	88%	88%	1 min	3 min
	C1	96%	91%	2.18 min	6 min
	D1	95%	93%	3 min	9 min
	E1	96%	92%	3.33 min	9 min
	F1	97%	92%	3.47 min	9 min
	G1	95%	92%	4.02 min	9 min

TABLE II: Predictions performance at t_{+15min} before and after applying data balancing technique.

Data Balancing	Loop	Recall (%)	F1-score (%)	Average prediction delay (min)	Max prediction delay (min)
Before	B1	80%	85%	3.5 min	9 min
	C1	78%	80%	10.36 min	36min
	D1	82%	84%	7.12 min	24min
	E1	92%	92%	8.06 min	30 min
	F1	93%	92%	9.42 min	48min
	G1	88%	89%	9.8min	51 min
After	B1	87%	83%	2 min	9 min
	C1	92%	84%	3.54 min	15 min
	D1	94%	86%	3.6 min	15min
	E1	95%	85%	4.15 min	15 min
	F1	96%	86%	5.57 min	18 min
	G1	93%	87%	4.36min	15 min

TABLE III: Predictions performance at t_{+30min} before and after applying data balancing technique.

obtained, it can be seen that XGBoost performs well in most loop sensors, with an average recall rate of 95% and an average f1-score of 94.5%. We also observe that the average prediction delay increases from loop B1 to loop G1, this is due to the recurrent and the active traffic congestion at the entrance of the tunnel. Also, in Fig. 9, one day prediction results of XGBoost for the 6 sensors are presented. From this figure, we can see observe that the model can anticipate the traffic congestion with a maximum prediction delay of 3 minutes.

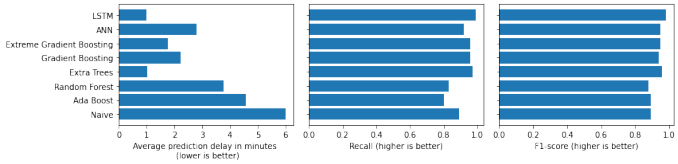


Fig. 10: Forecasting performance of the designed models for time horizon of t_{+6min}

4) 15min-horizon congestion forecasting: In this experiment, we set the time horizon to 15 min and we use the classification paradigm. The reason behind this is that generally models performance decreases significantly with the increase of time horizon, thus classification is preferred over regression to enhance the prediction performances. In addition, we use the density (K) as input instead of other variables, as density (K) provides advanced information about traffic congestions. Indeed, it generally increases at least 15 minutes before the onset of congestion. Such case is depicted in Figure 7, where density begins to increase at 6:30 a.m. before the congestion onset at 7 a.m.

In order to inspect the impact of the data balancing step on the prediction performances, we show in Table 2 the forecasting results of the XGboost technique before and after applying data balancing. Based on these results, one can notice the significant improvement of predictions after this preprocessing step, especially, in terms of prediction delay. The latter is the most important metric for evaluating the performance of the model on congestion anticipation. Indeed, the RL parts are activated according to the first occurrence of congestion. Therefore, the best model is the one with the smallest delay. For instance, we improve the average prediction delay of all sensors by 52% (from 5.92 minutes to 2.83 minutes). Next, in Fig. 12, to confirm the results obtained earlier, we investigate the effect of the data balancing technique on all the forecasting models. From the resulting plot, we can observe that the data balancing improves the performance of almost all the models especially in terms of prediction delays. However the amelioration isn't the same for all the models.

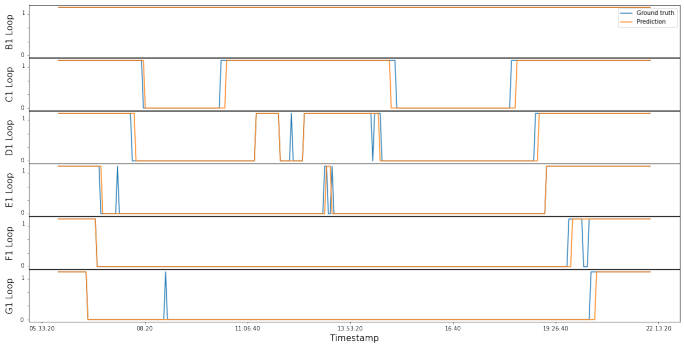


Fig. 11: A one day forecasting results at t_{+15min} on 6 loop sensor using XGBoost.

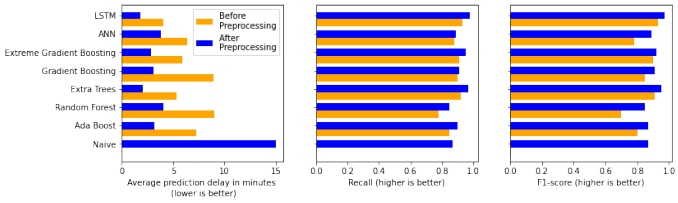
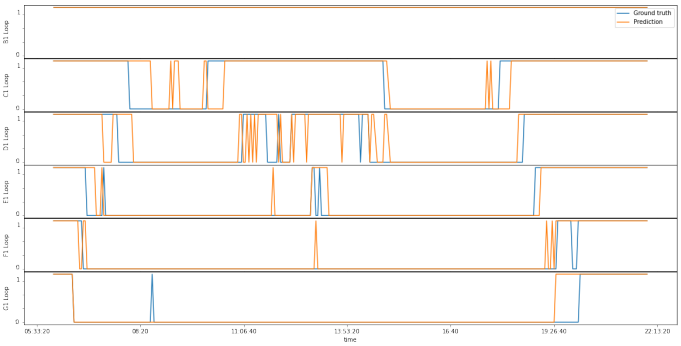


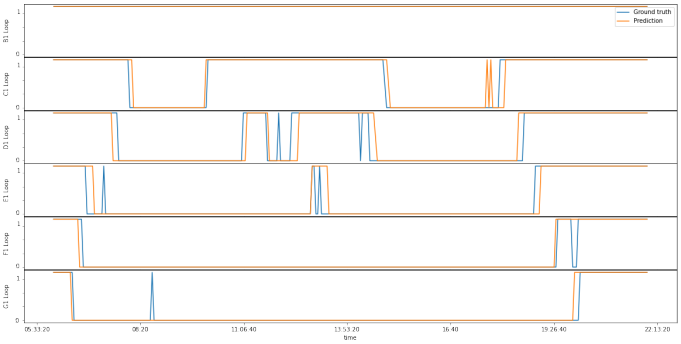
Fig. 12: Forecasting performance at t_{+15min} before and after applying data balancing technique and in function of the implemented methods.

5) 30min-horizon congestion forecasting: In this part, we set the time horizon to 30 min and we use almost the same strategy as in the 15min-horizon. In addition, we add calendar data to the model input to provide it with more information about the seasonality of the time series data. Such calendar data are: weekends, public holidays and school holidays. The objective of the present experiments is to show the behaviour of our framework with large time horizons.

As for the precedent test, We present the prediction performances of the XGBoost model before and after applying the data balancing technique in Table 3. The results demonstrate



(a)



(b)

Fig. 13: A one day forecasting results using XGBoost at horizon t_{+30min}. Graph (a) presents forecasting results before data balancing, while graph (b) presents forecasting results after data balancing

that even with large time horizons as 30 min, the improvement is still significant in terms of predictions delay. Now, to show the nature of the forecasts of the 30 min time horizon, we compute predictions for a random day in both cases before and after data balancing using XGBoost, and we depict the results in Fig. 13. From this figure, we can notice that noise and prediction delay increase significantly in the first graph. Finally, we inspect the quality of the predictions obtained for all the considered forecasting models for the 30 min time horizon. As for the 15 min time horizon, we observe that the data balancing permits to enhance significantly the scores for almost all the models. In addition, the comparative results of the prediction models show that the LSTM as for the other time horizons considered in this study, outperforms the other models whatever the cases (performing or not the data balancing technique).

VI. Conclusion

Concomitantly with the evolution of intelligent transportation systems and smart cities applications, short-term traffic congestion forecasting is still an actively studied topic in the literature. However, with the increase of motorists, the high dynamic nature of traffic data as well as the lack of complete and large traffic data sets, accurate congestion forecast becomes a challenging task. Our contribution is to provide a complete

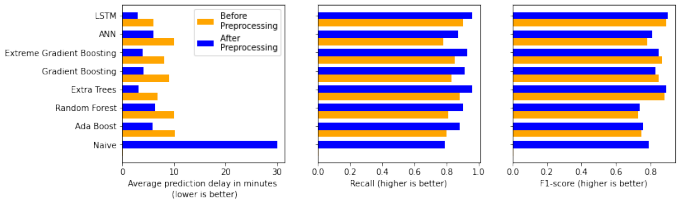


Fig. 14: Forecasting performance at t_{+30min} before and after applying data balancing technique and in function of the implemented methods.

process that aims to both improve data quality and produce more accurate forecasts. The strength of this study lies, on the one hand, in the real-time and continuous forecasting features and on the other hand, in the fact that our proposed methodology was validated by traffic supervisors and currently used by the Metropole de Lyon to regulate congestion. It has been proved that the use of this methodology can provide an accurate congestion forecast with a 95% success rate despite the highly dynamic character of our case study and the lack of some data from the failed sensors along the highway.

Acknowledgement

This research work has been carried out as part of the "Lyon Covoiturage Experimentation – LCE" project at IRT SystemX, Lyon, France, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir". The authors thank the Metropole de Lyon, which provide the data for this study and Spie for the partnership and collaboration in this project.

References

[1] Casmir Onyeneke, Chibuzor Eguzouwa, and Charles Mutabazi. Modeling the effects of traffic congestion on economic activities-accidents, fatalities and casualties. *Biomedical Statistics and Informatics*, 3(2):7–14, 2018.

[2] Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I Vlahogianni. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018.

[3] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[4] Navin Ranjan, Sovit Bhandari, Hong Ping Zhao, Hoon Kim, and Pervez Khan. City-wide traffic congestion prediction based on cnn, lstm and transpose cnn. *IEEE Access*, 8:81606–81620, 2020.

[5] Haitao Yuan and Guoliang Li. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Science and Engineering*, 6(1):63–85, 2021.

[6] Mahmuda Akhtar and Sara Moridpour. A review of traffic congestion prediction using artificial intelligence. *Journal of Advanced Transportation*, 2021, 2021.

[7] Chun Ai, Lijun Jia, Mei Hong, and Chao Zhang. Short-term road speed forecasting based on hybrid rbf neural network with the aid of fuzzy system-based techniques in urban traffic flow. *IEEE Access*, 8:69461–69470, 2020.

[8] Wentian Zhao, Yanyun Gao, Tingxiang Ji, Xili Wan, Feng Ye, and Guangwei Bai. Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access*, 7:114496–114507, 2019.

[9] Usue Mori, Alexander Mendiburu, Maite Álvarez, and Jose A Lozano. A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11(2):119–157, 2015.

[10] Alireza Ermagun and David Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 38(6):786–814, 2018.

[11] S Vasantha Kumar and Lelitha Vanajakshi. Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, 7(3):1–9, 2015.

[12] Quang Thanh Tran, Zhihua Ma, Hengchao Li, Li Hao, and Quang Khai Trinh. A multiplicative seasonal arima/garch model in evn traffic prediction. *International Journal of Communications, Network and System Sciences*, 8(4):43, 2015.

[13] Jianhua Guo, Wei Huang, and Billy M Williams. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, 43:50–64, 2014.

[14] Mascha Van Der Voort, Mark Dougherty, and Susan Watson. Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307–318, 1996.

[15] Howard R Kirby, Susan M Watson, and Mark S Dougherty. Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International Journal of Forecasting*, 13(1):43–50, 1997.

[16] Chuan Luo, Chi Huang, Jinde Cao, Jianquan Lu, Wei Huang, Jianhua Guo, and Yun Wei. Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm. *Neural processing letters*, 50(3):2305–2322, 2019.

[17] Liang Zheng, Chuang Zhu, Ning Zhu, Tian He, Ni Dong, and Helai Huang. Feature selection-based approach for urban short-term travel speed prediction. *IET Intelligent Transport Systems*, 12(6):474–484, 2018.

[18] Zhao Liu, Wei Du, Dong-mei Yan, Gan Chai, and Jian-hua Guo. Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression. *Journal of Highway and Transportation Research and Development (English Edition)*, 12(1):89–96, 2018.

[19] Seyed Omid Mousavizadeh Kashi and Meisam Akbarzadeh. A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks. *Journal of Intelligent Transportation Systems*, 23(1):60–71, 2019.

[20] Robin Kuok Cheong Chan, Joanne Mun-Yee Lim, and Rajendran Parthiban. A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system. *Expert Systems with Applications*, 171:114573, 2021.

[21] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

[22] Zhao Huang, Jizhe Xia, Fan Li, Zhen Li, and Qingquan Li. A peak traffic congestion prediction method based on bus driving time. *Entropy*, 21(7):709, 2019.

[23] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

[24] Weibin Zhang, Yinghao Yu, Yong Qi, Feng Shu, and Yinhai Wang. Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning. *Transportmetrica A: Transport Science*, 15(2):1688–1711, 2019.

[25] Xianglong Luo, Danyang Li, Yu Yang, and Shengrui Zhang. Spatiotemporal traffic flow prediction with knn and lstm. *Journal of Advanced Transportation*, 2019, 2019.

[26] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7):1501, 2017.

[27] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044, 2015.

[28] Giovanni Buroni, Yann-Aël Le Borgne, Gianluca Bontempi, Daniele Raimondi, and Karl Determe. On-board unit big data: Short-term traffic forecasting in urban transportation networks. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 569–578. IEEE, 2020.

[29] Donna Xu, Yaxin Shi, Ivor W Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen. Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*, 31(7):2409–2429, 2019.

[30] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[31] Rivindu Weerasekera, Mohan Sridharan, and Prakash Ranjitkar. Implications of spatiotemporal data aggregation on short-term traffic prediction using machine learning algorithms. *Journal of Advanced Transportation*, 2020, 2020.