Advanced Engineering Informatics 50 (2021) 101343

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Improving traffic prediction using congestion propagation patterns in smart cities

Attila M. Nagy *, Vilmos Simon

Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics Budapest, Magyar Tudósok krt 2., Budapest, Hungary

ARTICLE INFO

Keywords:
Traffic forecast
Traffic propagation
Exogenous data source
Smart cities
Intelligent transport systems

ABSTRACT

Accurate traffic forecast is a key task for planning transport infrastructure and real-time optimisation of traffic in large cities. The models used in professional literature usually provide accurate forecasts, but in case of congestion, forecasts can be highly inaccurate. At the heart of these situations are complex processes taking place on the road network of the city, which the prediction models are rarely prepared for. The congestion phenomena propagating on the road network of large cities have a major impact on the development of traffic patterns.

In this article, we present a new traffic prediction model, the Congestion-based Traffic Prediction Model (CTPM), which refines previous forecasts based on congestion propagation patterns. Our aim is to show that using congestion data can greatly improve our forecasts. The developed model can be used in conjunction with any previous model, so there is no need to replace well-functioning methods. To the best of our knowledge, no method has yet been developed that takes traffic information into account for forecasting in such a way. Our performance studies have shown that by using CTPM we were able to refine traffic forecasts by an average of 9.76%.

1. Introduction

Nowadays, significantly increased vehicle traffic is a major challenge for city management organizations. The world currently has 1 billion vehicles on the road, but estimations say this could rise to 2.5 billion by 2050 [1], so efficient vehicle management is a key task for a large city.

Dating all the way back to the 1970s, so called Intelligent Transportation System (ITS) [2] solutions have been created to address this problem. The ITS systems are the cornerstone of traffic management and urban planning in the smart cities of today. One of the critical points of these systems is traffic forecasting, which predicts traffic trends of the future on the basis of current and historical data. These forecasts can be useful inputs for traffic light controls or route planning, which can prevent or significantly reduce the negative effects brought on by congestion.

Researchers have been working on vehicle traffic forecasting for a long time. In professional literature there are many published solutions [3,4] that work fairly accurately in normal conditions when there are no unexpected events on the road network. However, in many cases these methods are not prepared for complex congestion propagation patterns. Thus, they can provide very inaccurate forecasts in extreme situations, even though precise forecasts would be most needed during these time periods. These critical situations can be caused by incidents, extreme weather conditions, traffic hazards, etc. Although the uncertainty factor cannot be completely eliminated [4] from traffic prediction, the negative impact of congestion can be reduced by taking into account exogenous data sources [3].

Data from processes with a significant impact on vehicle traffic can be used as an exogenous data source. These external data sources provide additional information to the prediction models so they can refine their forecasts. External data sources may include, for example, data from neighbouring measuring stations, weather data, incident information, or national holiday and event dates. Integrating data like these into models has posed a major challenge for researchers since the way to achieve it is unclear. Another challenge is that in many cases only a small amount data is available, while training a prediction model would require orders of magnitude more.

One of the major problems in the transportation of major cities around the world is the phenomenon of traffic jams and congestion occurring on the road network. Congestion has a serious impact not only on the lives of vehicle drivers, but also on the lives of every inhabitant of the city. Congestion increases fuel consumption [5] and harmful

* Corresponding author.
E-mail addresses: anagy@hit.bme.hu (A.M. Nagy), svilmos@hit.bme.hu (V. Simon).

https://doi.org/10.1016/j.aei.2021.101343
Received 17 December 2020; Received in revised form 13 April 2021; Accepted 14 June 2021
Available online 2 August 2021

emissions, [6] and causes significant economic damage [7]. According to a laboratory study [8], pollution associated with congestion increases the chances of developing allergies and exacerbates the symptoms of people who are sensitive to them. Other studies [9] have shown that congestion increases the risk of heart attacks.

In recent years, several papers [10–14] have been published that deal with understanding and forecasting the propagation of congestion. These may be valuable sources for the prediction models, but so far we have not found any attempt in the professional literature to integrate congestion related data. For this reason, this article presents a new prediction model that can integrate useful information from exogenous, congestion-related data sources into the prediction model. By doing so, we want to demonstrate that congestion data are also an important external data sources for traffic forecasting. For example, we can foresee situations where congestion on a road segment would soon lead to congestion on adjacent road segments. In such a situation the traffic forecast can support traffic control centers in managing the road network and allocating resources systematically, such as opening/closing lanes [15], dynamic parking pricing [16], or adaptive traffic lights [17] with a high level of automation [18].

The remainder of this article contains the following chapters. In Section 2, we review the efforts made in professional literature to integrate exogenous data sources into the prediction models. In Section 3, we present the basic idea behind our method and describe the methods on which we will build later. Then in Section 4, we describe in detail the features used in our prediction model and the architecture created. The quality of the CTPM is proven by performance studies carried out in Section 5. Finally, we conclude our article with a short conclusion in Section 6.

2. Related works

As shown in Section 1, exogenous data sources can play an important role in refining traffic forecasts. There are two difficulties that usually come up when using data sources. First, getting the data isn't easy. In several cases, research is hampered by the lack of a sufficient amount and quality of data to train the prediction models. Second, the way in which these data sources can and should be integrated into existing prediction models is also challenging for researchers. Despite this, there have been considerable efforts in recent professional literature to show that the integration of exogenous data sources is a forward-looking effort.

In Article [19], traditional traffic models are used to demonstrate the serious negative impact of weather changes on traffic parameters, such as speed, road network capacity or critical density. Having assessed the effects, they created a modified model of the METANET model [20] by adding snowfall data, which then achieved on average a 21% better Root Mean Squared Error (RMSE) than the basic model.

The authors of [21] created a Deep Belief Network (DBN) based prediction model which can take weather data into account. Several weather-related features were then tested. With the help of cross-correlation tests, they manually selected those that affect traffic flow. For the forecasting of traffic flow, the following weather parameters were taken into account in addition to the measured traffic flow: weather condition, temperature, visibility and wind gust. The weather condition parameter contains 24 different weather phenomena such as fog, rain, snow, etc. For each element of the set, an integer was assigned from 1 to 24 to enable DBN to interpret it also.

Performance analysis was performed on flow data with 15-min aggregation and a 15-min prediction horizon. Based on the RMSE values presented in the article, a 16% improvement in forecast accuracy was achieved.

The Deep Ensemble Stacked Long Short Term Memory (DE-SLSTM) [22] method also integrates weather data, in addition to short-term and historical speed data, to improve the accuracy of the forecasts. The weather data used includes atmospheric pressure, temperature, precipitation, wind speed, and relative humidity. In contrast to the method described in Article [21], the relationship between this weather data and traffic data has not been investigated. According to the performance analysis presented, in those cases where weather data were taken into account, better results were achieved but the rate of improvement was not significant.

In addition to weather conditions, holidays and other events also affect traffic trends. In the event of a national holiday, while some parts of the city may be deserted, other parts and certain segments of motorways may experience increased loads. These times are an anomaly for a model that has not been prepared for this type of behavior. When training such models, it is challenging that a certain holiday occurs only once a year. Therefore, even if we looked at ten years of data, we would only have a limited amount.

The Segment Prediction Algorithm (SPA) [23] method first divides the data into segments for each day and then classifies the data measured on holidays into clusters using the k-Means algorithm. Clustering can be used to merge holidays that show similar behavior, thus reducing data shortage. SPA will then use binary linear regression to make forecasts during holidays. According to the results of that article, the method achieved a 40% improvement in RMSE values compared to the Long Short Term Memory (LSTM) network.

The solution described in Article [24] is a hybrid method that can achieve accurate forecasts for national holidays by combining the Discrete Fourier Transform (DFT) and Support Vector Regression (SVR) methods. The purpose of using DFT is to find a general trend in traffic data. Subtracting this trend from the measured data, the forecast is made using the SVR model on the remaining time series. The solution was tested at a motorway toll gate in Jiangsu province, China, where increased traffic was observed during the national holidays. Their measurements showed that their method could achieve an improvement of up to 17% on, for example, the 2015 Tomb-Sweeping Day. It was also observed that the method performed better on longer holidays of 6–7 days than on shorter holidays of 2–3 days, because of the much smoother time series found in the former.

The forecast may encounter significant inaccuracies should an incident occur. In this case, the number of vehicles on the post-incident road segments (downstream) decrease, while in the pre-incident section (upstream) the speed of vehicles is what may decrease drastically. The rate of reduction varies depending on the severity and type of incident. The use of incident data is complicated by the fact that these events occur rarely, so models with high data demand are difficult to use.

In addition to the weather, the prediction model developed by the authors of Article [25] takes incident data into account as well. The information regarding the incident was added to the model as a binary feature. For example: Has there been a serious incident on the downstream side of the measuring station? Has there been an incident on the upstream side of the measuring station? etc. The feature selection performed in the article using correlation analysis and Least Absolute Shrinkage and Selection Operator (LASSO) regression [26] has shown that the incident data contains important information. Unfortunately, the performance studies carried out in the article do not specify exactly how much the use of incident features has improved the accuracy of the forecast.

Congestion propagation patterns may also be an important exogenous data source. By identifying the congestion propagation paths, it is possible to anticipate hidden processes on the road network that may occur within a specified time (the length of the time depends on the speed of the congestion propagation), for example, the congestion of a road segment can herald the future congestion of an adjacent road segment.

A number of studies [11–13] have already been carried out to try define congestion propagation paths, which then can help explore paths on the directed graph of the road network. Congestion Propagation Modeling Algorithm (CPMA) [14] adds to this its ability to assign propagation probabilities and expected propagation times to explored

propagation paths. These may be valuable sources for the prediction models, but so far we have not found any attempt in the professional literature to integrate congestion related data. Thus, our goal was to develop a new method, which can also use congestion-related data during traffic prediction, such as propagation paths, congestion probabilities, and expected propagation times.

This article contains the following scientific results:

- The creation of Congestion-based Traffic Prediction Model (CTPM), which can integrate congestion propagation data, thus refining the forecast.
- The CTPM can handle the challenge that often only a small amount of congestion propagation data is available, which makes it difficult to train the model.
- The CTPM can refine forecasts of other prediction models already known in professional literature so that the methods that have worked before can still be used. The CTPM supports arbitrary traffic variable such as *total flow* or *speed*.

3. Methodology

In the first part of this section, we present how we solved the challenges of integrating the previously mentioned congestion data in prediction models. Then, the CPMA and Extreme Gradient Boosting (XGB) are introduced since our solution is based on these models.

Let $\mathcal{N}(\mathcal{I},\mathcal{R})$ be a directed graph representing a city's road network, where $\mathcal{I}=\{I_1,I_2,\ldots,I_{|\mathcal{I}|}\}$ is the set of intersections and $\mathcal{R}=\{R_1,R_2,\ldots,R_{|\mathcal{R}|}\}$ the set of road segments. We can obtain information about the current state of traffic with the help of data collected from measurements taken by sensors installed or from vehicles on the examined road segments. The collected traffic data can be easily transformed to congestion data by applying arbitrary congestion definition [27–29] from the literature. As CPMA uses binary congestion data (congested and uncongested states), we can use a simple threshold-based congestion definition. From the obtained data we can extract $\mathcal{PP}=\{PP_j|j=1,\ldots,|\mathcal{PP}|\}$ that is a set of congestion propagation path where $|\mathcal{PP}|$ represents the size of the set and $PP_j=\{R_u\rightarrow\ldots\rightarrow R_v|R_u,R_v\in\mathcal{R}\}$ is a propagation path containing an ordered finite sequence of road segments. Road segment R_v, which is the last road segment of propagation path PP_j, is called the *target road segment*.

These propagation paths PP_j can be integrated into the prediction models, but we face two challenges when using them: there is a small amount of available data and the arbitrary shape of the congestion propagation path can be difficult to integrate into a general model.

Only those congestion phenomena carry information value that propagate to several road segments and thus form a propagation path PP_j. In cases where the phenomenon is caused by a one-time event, we cannot collect enough data about it, so they cannot be used during training. However, propagation paths PP_j that repeat multiple times can provide useful data. The repetition of the propagation path also allows us to determine the expected propagation time and propagation probability.

Since the number of recurrences may vary significantly between two congestion propagation paths, we do not build separate prediction models for each of them, but rather a prediction model that can handle all known congestion propagation paths. Of course, in cases where we have a large amount of data on one of the congestion propagation paths, taking into account the data on other congestion propagation paths may negatively affect the prediction performance, but this approach will produce better results than building models separately in general.

The second challenge arises from the desire to handle all congestion propagation uniformly, even though the lengths of different congestion propagation may vary. The problem is solved by always taking into account the data of the last road segment that is congested or, in other words, where the congestion currently stands.

For example, if we want to predict the traffic of road segment R_3, let $PP=\{R_1\rightarrow R_2\rightarrow R_3\}$ be a propagation path where congestion propagates from road segment R_1 to road segment R_3. Let's assume that the congestion formed on R_1 at time t_1. The phenomenon propagated to R_2 at time t_5 and finally reached R_3 at time t_8. In this case, in time interval $[t_1,t_5)$ the data from R_1 is taken into account and in time interval $[t_5,t_8)$ the data from R_2 is taken into account for the forecast of R_3.

As the CPMA algorithm is capable of producing the necessary propagation paths and statistics, the operation of this algorithm is the first to be described in Section 3.1. Then in Section 3.2 we present the operation of the XGB [30] model, which will be used as a base prediction model. Finally, in Section 4.2 the detailed structure of the CTPM and the set of features used in the forecast are described.

3.1. CPMA algorithm

The Congestion Propagation Modeling Algorithm (CPMA) is responsible for finding the frequently occurring propagation paths on the graph of the city's road network and for calculating the propagation probabilities and expected propagation times associated with the propagation paths, based on the input dataset.

First, CPMA identifies the propagation paths, using the Spatial Congestion Propagation Patterns (SCPP) [13] algorithm. SCPP awaits the data in the form of matrix $\overline{\overline{C}}_{\mathcal{N}}$, sized $|\mathcal{R}|\times T$. Matrix $\overline{\overline{C}}_{\mathcal{N}}$ contains the congestion observations of the road segments $\mathcal{R}$ of the road network $\mathcal{N}(\mathcal{I},\mathcal{R})$ for a T long examined time period measured in the number of time intervals. $\overline{\overline{C}}_{R_r,t}=1$ if the road segment R_r was congested in the time interval number t. $\overline{\overline{C}}_{R_r,t}=0$ if the road segment R_r was not congested in the time interval number t.

To explore the effects, we also need the adjacency information of the road segments. The CPMA expects this as its input in the form of matrix $\overline{\overline{A}}_{\mathcal{N}}$. The matrix $\overline{\overline{A}}_{\mathcal{N}}$ is the edge-adjacency matrix of the directed graph $\mathcal{N}(\mathcal{I},\mathcal{R})$ sized $|\mathcal{R}|\times|\mathcal{R}|$, which can be determined from the adjacency matrix of the line graph of the graph $\mathcal{N}(\mathcal{I},\mathcal{R})$. If two road segments are adjacent to each other $\overline{\overline{A}}_{R_u,R_v}=1$, otherwise $\overline{\overline{A}}_{R_u,R_v}=0$ $(R_u,R_v\in\mathcal{R})$.

The last input parameter is the value ϵ, which is the threshold value that determines the minimum frequency of a propagation path. If a propagation path is recorded more than ϵ times, it is considered as a frequent propagation path. Its value is worth determining based on the length of the examined time period T, but it also depends on what is common for the environment.

With the help of the matrix $\overline{\overline{C}}_{\mathcal{N}}$, matrix $\overline{\overline{A}}_{\mathcal{N}}$ and threshold value ϵ SCPP find congestion propagation paths with complexity $\boldsymbol{O}(T|\mathcal{R}|+|\mathcal{L}|)$, where T is the length of the examined period in time intervals, $|\mathcal{R}|$ the number of road segments, which is an upper estimate of the number of occurring congestions, and $|\mathcal{L}|$ is the number of edges of the directed graph describing the possible propagation paths. This is significantly faster and more scalable than other solutions found in professional literature that have quadratic or exponential complexity.

The output of the SCPP algorithm is the set of congestion propagation paths $\mathcal{PP}=\{PP_j|j=1,\ldots,|\mathcal{PP}|\}$. CPMA determines the propagation probability and the expected propagation time in time intervals for each PP_j propagation path.

CPMA takes advantage of the fact that congestion propagation has a Markov property. The congestion propagation process for a propagation path PP_j is described by a Markov process, which is represented by a state sequence $\mathcal{X}=\{X_1,X_2,\ldots,X_T\}$. $X_t=R_r$, if the state of the process at the number t time interval is the same as R_r road segment, $R_r\in PP_j$. In the event that the propagation stops before it reaches the last road segment from PP_j, it will go into zero state, which represents the absence of propagation.

The state transition matrix $\overline{\overline{\boldsymbol{P}}}$ can be determined using the state sequence. Certain values of the state transition matrix contain the

probability of the congestion propagating between two road segments that are included in propagation path PP_j. Matrix $\overline{\overline{\boldsymbol{P}}}$ can be used to determine the probability of a full propagation path PP_j:

$$Pr\left(R_u \rightarrow R_v\right) = \prod_{r=1}^{|PP_j|-1} \frac{p_r}{1-\widehat{p}_r}, \tag{1}$$

where R_u is the source road segment of propagation path PP_j, R_v is the destination road segment of propagation path $PP_j, \widehat{p}_r$ is the probability that the congestion does not propagate from road segment number r to road segment number $r+1$ and p_r is the probability that the congestion does propagate from road segment number r to road segment number $r+1$. The probability of the congestion has come to the end (terminated) is $1-(p_r+\widehat{p}_r)$.

Using the propagation path probability, the expected propagation time can also be determined between R_1 and $R_{|PP_j|-1}$:

$$E\left(t, R_1 \rightarrow R_{|PP_j|-1}\right) = \overline{e}_1 \left(\overline{\overline{\boldsymbol{I}}} - \overline{\overline{\boldsymbol{P}}}_{1\times|PP_j|-1}\right)^{-2} \overline{e}^T_{|PP_j|-1} \frac{p_{|PP_j|-1}}{Pr(R_u \rightarrow R_v)}, \tag{2}$$

where the value of vector $\overline{e}_1$ is $[1,0,\ldots,0]$, the value of vector $\overline{e}_{|PP_j|-1}$ is $[0,\ldots,0,1]$, $\overline{\overline{\boldsymbol{P}}}_{1\times|PP_j|-1}$ is the sub-matrix of matrix $\overline{\overline{\boldsymbol{P}}}$ formed from state $\{R_1,\ldots,R_{|PP_j|-1}\}$ and $p_{|PP_j|-1}$ is the probability of congestion propagating from $R_{|PP_j|-1}$ to R_v. Both $\overline{e}_1$ and $\overline{e}_{|PP_j|-1}$ are $|PP_j|-1$ long.

3.2. Extreme Gradient Boosting (XGB) regression model

In our solution, we used the XGB [30] model, which is an enhanced version of the Gradient Boosting Decision Tree (GBDT) [31] model, as a base prediction model. GBDT is a tree-based ensemble learning algorithm that is also widely used for regression tasks, thanks to its efficiency and accuracy, and the interpretability of its results.

Let the input dataset be $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$, which consists of n pieces of record and m pieces of features, so that $|\mathcal{D}| = n, \boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}$. Let f_k mark a simple decision tree, where $k = 1,\ldots,K$. In the case of tree-based ensemble methods, we increase the performance of the prediction by using more weak decision tree models at the same time. To do this, all weak decision tree models must be at least as accurate as if the forecasts were carried out at random. These weak decision trees are created sequentially using an additive strategy, so that the newly trained decision tree corrects the output of the previous decision trees. The output of these weak decision trees is then summed:

$$\widehat{y}_i = \sum_{k=1}^{K} f_k\left(\boldsymbol{x}_i\right), f_k \in \mathcal{F}, \tag{3}$$

where $\widehat{y}_i$ is the predicted value, K is the number of trees and f_k is a function from the set of functions $\mathcal{F}$, which represents the set of actually useful regression trees. Thus, the task while training is to optimise objective function $\mathcal{L}$:

$$\mathcal{L}\left(\theta\right) = \sum_{i}^{n} l\left(y_i, \widehat{y}_i\right), \tag{4}$$

in which θ denotes the parameters to be optimised and l is the loss function, which is convex, differentiable, and measures the distance between the real value y_i and the predicted value $\widehat{y}_i$.

XGB, like GBDT, uses an additive strategy while training, but adds a regulating term to the $\mathcal{L}$ objective function, which helps punish complex models in order to avoid overfitting. This modifies (4) as follows:

$$\mathcal{L}\left(\theta\right) = \sum_{i}^{n} l\left(y_i, \widehat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right), \tag{5}$$

ahol $$\Omega\left(f\right) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{6}$$

where f_k is a decision tree, $\Omega(f_k)$ is the regulating term, T represents the number of leaves of a decision tree and w_j represents the value of leaf number j. The parameter γ controls the number of leaves in the trees, while parameter λ controls the values of the leaves. Regulating parameters are really important when there are only small amounts of data available, so it is very important to use them in this domain.

To solve (5), we use the predicted value $\widehat{y}_i$, which, according to (3), is the sum of the predictions from outputs of decisions trees f_k ($k = 1,\ldots,K$). So, in the case of using k number of decision trees:

$$\mathcal{L}^{(k)}\left(\theta\right) = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(k)}\right) + \sum_{t=1}^{k} \Omega\left(f_t\right) \tag{7}$$

$$= \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(k-1)} + f_k\left(x_i\right)\right) + \Omega\left(f_k\right), \tag{8}$$

where the term (k) in the upper index indicates that the value was calculated by taking into account the results of k number of decision trees so far, and n is the number of records in dataset $\mathcal{D}$.

Its worthwhile to approximate the loss function within the objective function with its Taylor expansion up to the second order. This will not only make the use of any loss function easier, but will make the model converge faster to the global optimum during training:

$$\mathcal{L}^{(k)}\left(\theta\right) = \sum_{i=1}^{n}\left[l\left(y_i, \widehat{y}_i^{(k-1)}\right) + g_i f_k\left(x_i\right) + \frac{1}{2}h_i f_k^2\left(x_i\right)\right] + \Omega\left(f_k\right) + const, \tag{9}$$

where $g_i = \partial_{\widehat{y}_i^{(k-1)}} l(y_i, \widehat{y}_i^{(k-1)})$ is the first derivative, while $h_i = \partial^2_{\widehat{y}_i^{(k-1)}} l(y_i, \widehat{y}_i^{(k-1)})$ the second derivative. If the constant values are omitted, the term can be further simplified:

$$\mathcal{L}^{(k)}\left(\theta\right) = \sum_{i=1}^{n}\left[g_i f_k\left(x_i\right) + \frac{1}{2}h_i f_k^2\left(x_i\right)\right] + \Omega\left(f_k\right). \tag{10}$$

This objective function can also be used to find the best structure of a decision tree. Since it is not technically executable to try all possible tree structures, we determine only one level of the tree at a time. At the starting position of the greedy algorithm, all data records are located in a single leaf of the tree and then it iteratively gives new branches to the tree step by step. In each step, the set of data records I in the leaf is split into a set of left (I_L) and right (I_R) elements. This can be used to determine how much the loss decreased after a split:

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\left(\sum_{i\in I_L} h_i\right)+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\left(\sum_{i\in I_R} h_i\right)+\lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\left(\sum_{i\in I} h_i\right)+\lambda}\right] - \gamma. \tag{11}$$

The greater the reduction, the better the split was. This means the structure of the tree is always modified by the best possible split in each step.

4. The Congestion-based Traffic Prediction Model (CTPM)

In this section we describe the elements of CTPM in detail. When training the model, we assume that we have already executed the CPMA algorithm for a selected time period so the set of propagation paths $\mathcal{PP}$ is known. In addition, a prediction model is available for each *target road segment* R_v, which makes forecasts without taking traffic information into account. The *target road segment* R_v is the road segment for which we want to forecast traffic data for, e.g. flow, speed, occupancy, at time t and time distance *horizon*, where both t and *horizon* are measured in time

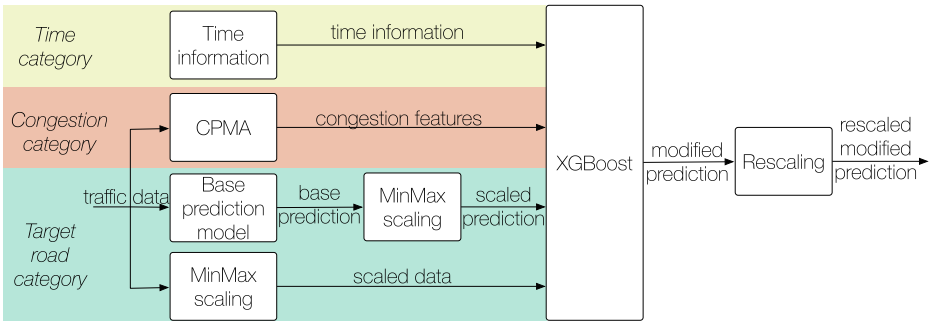


Fig. 1. The features used in the model.

Table 1
Table of features used for the forecast.

Category	Name	Data type	Value	Description
Time	*minute*	int	[0, 60)	Minute of time t
	hour	int	[0, 24)	Hour of time t
	dayofweek	int	[0, 7)	Number of days in week of time t
	month	int	[1,12]	Month of time t
Target road	$prediction_{[t,t-2]}$	list of floats	[0, ∞)	Forecasts of road segment R_v
	$bestlags_{[0,10]}$	list of floats	[0, ∞)	Best 10 lags
Congestion	*distance*	float	[0, ∞)	Distance between road segments R_r and R_v
	expected probability	float	[0,1]	Propagation probability between road segments R_r and R_v
	expected time	int	[0, ∞)	Expected propagation time between road segments R_r and R_v
	expected arrival time from state	int	[-∞, ∞)	Expected arrival time of congestion between road segments R_r and R_v
	elapsed time in state	int	[0, ∞)	Elapsed time on road segment R_r
	$cval_{[t,t-6)}$	list of floats	[0, ∞)	Traffic data measured on road segment R_r

intervals.

The remainder of this section contains the following sections. First, we define the set of required features in Section 4.1, which the XGB model uses while training and forecasting. For the sake of transparency, we have organized the features into a table and have also defined the data transformation steps where it is not clear. Then, in Section 4.2, we describe what steps CTPM are taken during the forecast.

4.1. Feature set

In this section we describe the defined features that CTPM uses at time t to forecast traffic data at time $t + horizon$. We collect the features separately for each propagation path PP_j, and then combine them to form the dataset for the XGB model. For each propagation path PP_j, data is collected only from time periods where active congestion propagation was observed. In other cases, the forecasts of the base prediction model are taken into account (see 4.2). Let the state X_t at time t of active congestion propagation path PP_j be $X_t = R_r$, where road segment R_r is the road segment the congestion propagation currently stands.

The defined features are divided into three categories depending on their source: time-related features, congestion-related features and features related to the *target road segment* R_v. These categories are marked with separate colours in Fig. 1 and with separate columns in Table 1.

4.1.1. Time features

Vehicle traffic is a periodically changing, time-dependent process. This becomes apparent when you plot traffic flow as a function of time. An example for this can be seen on Fig. 2, on which we displayed the measurements of a traffic measuring station from the second week of January 2018. It can be clearly seen that due to the periodic behavior of traffic, each day is a separate spike, and weekdays and the weekend are also well distinguished. On the figure, the congested time intervals are marked with red color. You can see that congestion occurs in morning and afternoon rush-hours during the weekdays, while the weekend is free from congestion. The congested time intervals were determined by the Volume to Capacity (V/C) method [32].

It is important for the CTPM to have information about the date related to time t, as these may contain important correlations. For

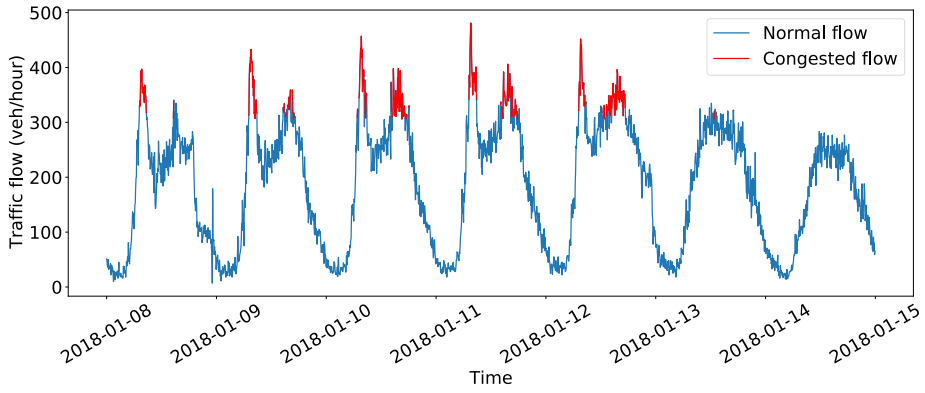


Fig. 2. Example for the periodicity of vehicle traffic.

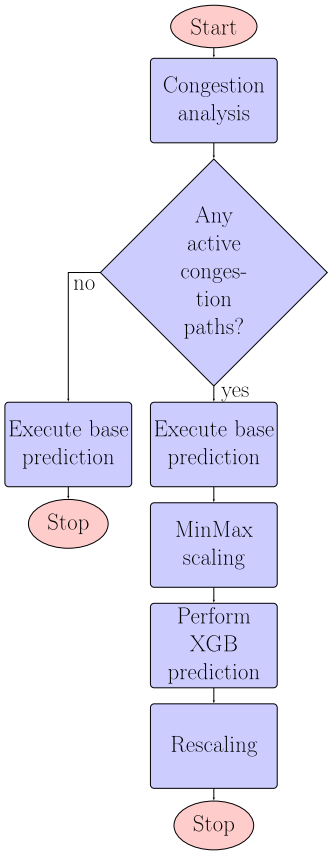


Fig. 3. The prediction process of the CTPM.

example, it is possible to distinguish between the morning and afternoon spike, or whether it is a weekend or not. The features related to time defined and used by us are described in Table 1 under the Time category.

4.1.2. Target road segment features

We collect two important pieces of information about the *target road segment* R_v. One is the forecast itself, carried out by a base prediction model at the same time t and refers to time $t + horizon$. This forecast needs to be scaled as shown in Fig. 1. This is because there is data from multiple propagation paths in the same dataset at the same time, with varying amounts of traffic on each road segment. We compared several scaling methods, and finally decided to use the Min-Max scaling [33] on interval $[0, 1]$, because it gave us the best results based on the metrics used during the evaluation.

During our study we also found that for the XGB model the predictions of earlier times ($t-2$) are also useful information if used as a feature. Taking into account forecasts from earlier than this showed to lower the accuracy of the forecast. In Table 1 the forecasts for the times $t, t-1$ and $t-2$ are indicated by the $prediction_{[t,t-2]}$ variable.

The other important information that can be collected from the *target road segment* R_v is the historical data itself relating to road segment R_v. Using autocorrelation, we examined which lags have the highest correlation with the current time t. As data measured in the previous week or two weeks earlier may also show a high correlation with time t, these were also included in the examination. Running autocorrelation studies is a resource intensive task, so we only ran it in interesting time periods. Based on the findings of Article [21], we decided to look at time windows with length 10. And so time periods $[t-1,\ t-10]$, $[t-1week,\ t-1week-9], \ldots$become a manageable amount of calculation.

Based on the correlation coefficients calculated like this, the best 10 lags were chosen and used as 10 different features. In Table 1. the current measurement for time t and the best 10 lags are recorded as the $bestlags_{[0,10]}$ variable.

4.1.3. Congestion features

The congestion features were always defined between the *target road segment* R_v of the propagation path PP_j and the currently congested road segment $X_t = R_r$. These are listed in Table 1 under the Congestion category. Congestion features can be divided into two groups according to their calculation method: static and dynamic.

Static features include data that appear immediately after the CPMA algorithm is run. The CPMA algorithm has already been run, as this is why the set of propagation paths $\mathcal{PP}$ is known. Static features include the propagation probability, the expected time of propagation, and the distance between road segments R_v and R_r.

The dynamic group of congestion features include data that is generated depending on time t. In *elapsed time in state*, we recorded how long the propagation path PP_j has been in the state of road segment R_r. *expected arrival time from state* determines the time it takes for the congestion to propagate from the road segment R_r to the road segment R_v from the current time t. This value may be negative if more time has passed than the expected propagation time calculated by the CPMA algorithm. The traffic data measured on the road segment R_r is also important data, as it may contain valuable information about congestion behaviour. Comparing different intervals, we found that taking into account the interval $[t, t-5)$ provides the best results as any time period that is longer will distort the output. Traffic information for road segment R_r is denoted by the $cval_{[t,t-6)}$ variable in Table 1.

4.2. The process of the forecast

The forecast follows the steps shown in Fig. 3. First, we perform a congestion analysis in each time t. During the congestion analysis, we go through the previously identified set of the propagation paths $|\mathcal{PP}|$ and see whether there exists a propagation path PP_j with the *target road segment* equal to R_v that is also currently in an active state. If there are more than one active propagation paths PP_j at the same time, we examine which of the road segments R_r in the state X_t of the propagations is closest to the target road segment R_v based on their distance. The distance is determined based on the measured length of the road on the road network.

The congestion features that are required to run the CTPM are also produced during the congestion analysis. Some of the propagation path-related features are static, such as the propagation probability or expected propagation time between R_r and R_v, while others are dynamic, such as how long a propagation path has been in a given state. These features are described in detail in Section 4.1.3.

The last step of the congestion analysis is filtering. Our studies have shown that propagation probability under 20% significantly degrades the output. The low probability means that there was little chance for the given propagation to occur, so the statistics collected rather distort the forecast than refine it.

If no active propagation path PP_j is found, we will simply use the base prediction model, because we assume that it can provide an accurate forecast for the road segment R_v if there is no active propagation. Otherwise, we found an active propagation, so we continue to carry out the steps described.

In case of an active propagation path, we also run the base prediction model first, the output of which is denoted by the $prediction_{[t,t-1]}$ variable in Table 1. The $prediction_t$ value will be the base prediction we want to improve.

Since we train only one XGB model for all *target road segments*, it is necessary to scale the traffic data of the *target road segment* and to also scale the related forecasts, even if the decision tree-based models are not sensitive to scaling. We compared several scaling methods, and finally decided to use the Min-Max scaling [33] on interval [0.1], because it gave us the best results based on the metrics used during the evaluation.

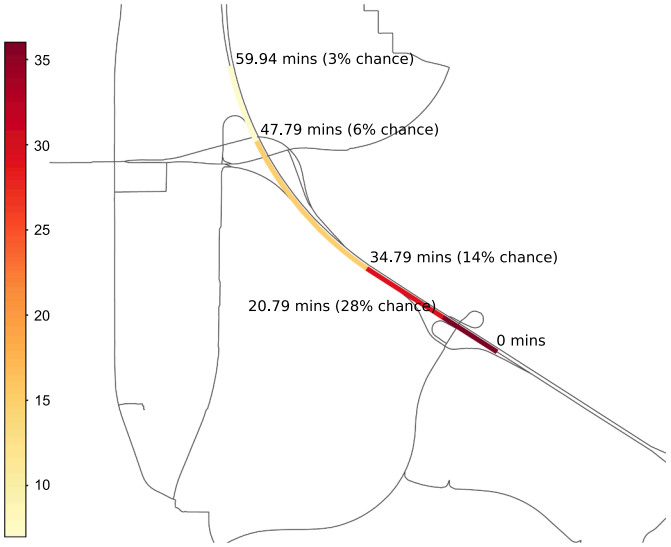


Fig. 4. Visualizing a traffic congestion propagation.

Scalers are produced during training, which is then stored for later use.

After scaling is performed, the XGB model can be run, which uses the features described in Table 1. These features are produced throughout the entire process. As the modified forecast returns a scaled value, it is necessary to scale the modified forecast back to the original range in the last step.

5. Evaluation

During the evaluation, our objective was to demonstrate that by integrating congestion data the CTPM can improve the performance of any prediction model. We implemented our method and the investigations in Python programming language. In the implementation, we used well-known scientific python packages, such as Pandas [34], Scikit-learn [35], Keras [36], and XGBoost [37]. We consider it important to highlight that we do not want to compare base prediction techniques, but rather to validate the usage of congestion data.

The following sections of this Section first describe the dataset used (Section 5.1) then briefly describe the base prediction models used (Section 5.2). Then, in Section 5.3 the results of the examinations are evaluated.

5.1. Dataset

During the evaluation, the Caltrans Performance Measurement System (PeMS) dataset [38] was used. The total dataset includes measurements of approximately 39,000 measuring stations located along main routes starting from 1999. Measuring stations have been installed across the state of California. The total size of the dataset is approximately 12 terabytes currently, which is publicly available and can be downloaded free of charge by anyone. The analysis of the entirety of the data would have taken too long, so we only examined a fraction of it. The evaluation was performed in District 3 (Sacramento area) for a one-year period from January 1, 2018 to December 31, 2018.

The studies included a total of 614 measuring stations, from which data were collected every 5 min (time interval $= 5$ min), a total of 105,120 times in one year. During the evaluation we used the first 80 percent of the studied time period as a training dataset, with the remaining 20 percent serving as a test dataset.

Fig. 4 shows a congestion propagation example from the dataset determined by CPMA. On the left side of the figure, a color scale is added to indicate the frequency of the propagation with color gradients. The propagation of the congestion can be clearly traced on the figure. The source of the congestion is indicated by "0 min" where the formation of the congestion was detected. It can be clearly seen that the farther we get from the source, the probability of propagation decreases, while the expected propagation time increases.

5.2. Base prediction models

To demonstrate performance we selected three base prediction models of different types. In the course of the examinations, the forecasts of these base prediction models are improved by the prediction model we have prepared.

The simplest of the three models is the Historical Average (HA) model. This naive method essentially determines the forecast on the basis of the arithmetic average of historical data [39]. Despite its simplicity, in the case of longer prediction horizons it can achieve better results than the more complex models. Another significance of the HA model is that it is a widespread base model used in actual navigation systems [40] in hybrid solutions [41]. In the case of the HA model, the data from the past 4 weeks was taken into account when executing forecasts.

The Long Short Term Memory (LSTM) [42] model is an extension of the Recurrent Neural Network (RNN) model that is capable of identifying short and long-term dependencies. The LSTM model was chosen because it is able to identify nonlinear relationships and its use is extremely widespread in current professional literature [43,22,44,45]. The disadvantage of training complex LSTM networks is that it is an extremely costly operation, so the training of a separate model for each road segment is very time-consuming. We built the LSTM models using the Keras [36] package. We used 50 LSTM neurons with the Rectified Linear Unit (ReLU) activation function. When training the Long Short Term Memory (LSTM) model, we used the best 50 lags, which we determined as described in Section 4.1.2. We have found that at 50 lags LSTM gives good results while training time remains manageable.

The XGB model has been selected as a base prediction models because over the past few years several researchers have successfully applied it to forecast traffic in professional literature [46–48], while it has been able to compete with more complex models in accuracy. The operation of the XGB model has already been described in detail in Section 3.2. Since the training of simple decision trees is a significantly less costly task, it learns more quickly by orders of magnitude than in the LSTM networks. To be fair, we also used the best 50 lags to train the XGB model.

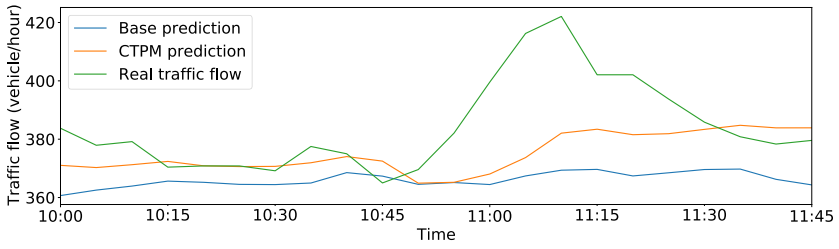
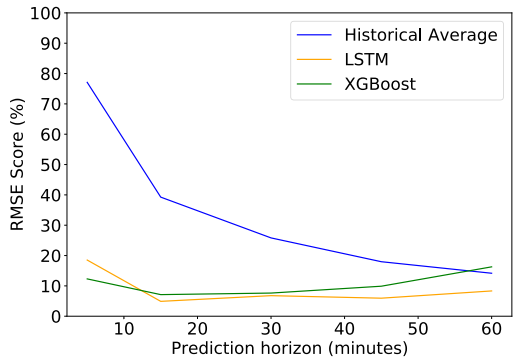
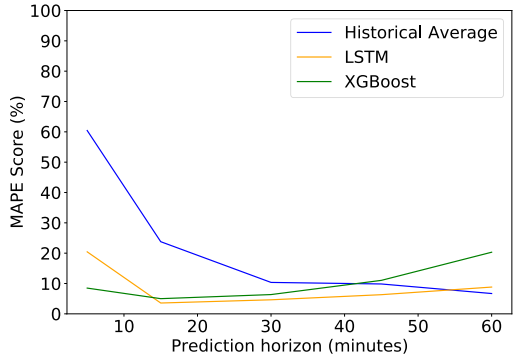


Fig. 5. An example of the prediction results.



(a) RMSE Scores



(b) MAPE Scores

Fig. 6. Results achieved by CTPM as a function of the prediction horizon.

5.3. Results

To train the XGB model inside the CTPM, the squared error objective function was used, while the hyperparameter optimization was achieved by the GridSearchCV [49] with 5-fold cross validation.

In Fig. 5, the output of the trained CTPM is compared with the real traffic flow and the output of the trained LSTM model. In the selected time interval, congestion was detected on the adjacent traffic measuring station. The collected congestion information from this sensor was used to refine the base predictions. It can be seen that the forecasts of CTPM are more accurate than the base prediction model thanks to the congestion information.

To evaluate the CTPM, the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) error metrics were used. The RMSE and the MAPE error metrics are the most widely used metrics to evaluate the performance of regression models. Let $\hat{y}$ be a T long series of forecast values and y be a series of real measurements from the same time period. In this case, the RMSE is calculated as follows:

$$RMSE\left(\hat{y}, y\right) = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}. \tag{12}$$

The MAPE metric is determined by the following equation:

$$MAPE\left(\hat{y}, y\right) = \frac{1}{T}\sum_{i=1}^{T}\left|\frac{y_i - \hat{y}_i}{y_i}\right| 100. \tag{13}$$

Since we are also curious about how much the CTPM improved the base prediction, we also created a metric called *Score*. The *Score* metric determines the percentage of how much more accurate the result of the modified forecast was compared to the original forecast based on the selected metric's values (RMSE or MAPE).

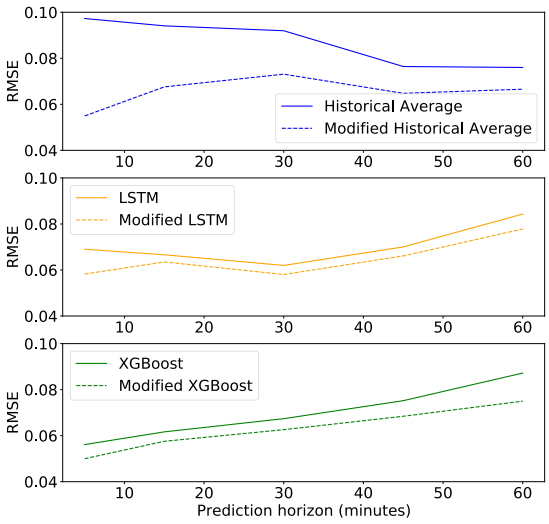
Let $METRIC_{base}$ denote the selected metric's value measured for the base prediction model and $METRIC_{CTPM}$ the selected metric's value measured for the CTPM. Using them, *Score* is equal to:

$$Score\left(METRIC_{base}, METRIC_{CTPM}\right) = \left(\frac{METRIC_{base}}{METRIC_{CTPM}} - 1\right)\cdot 100. \tag{14}$$

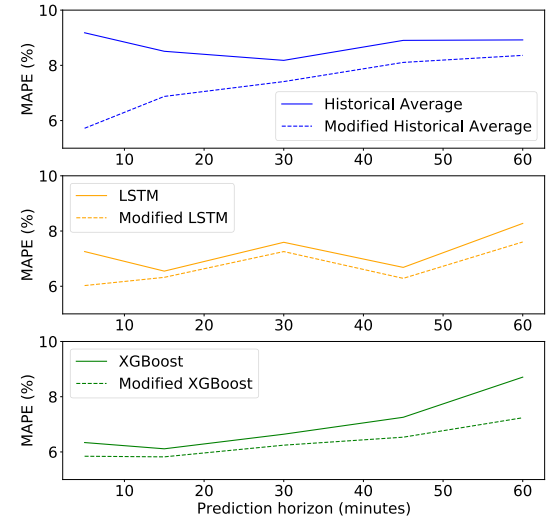
The value of the *Score* may also be negative if our prediction model degrades the original forecast.

The CTPM was compared with the three base prediction models along the 5, 15, 30, 45 and 60-min prediction horizons. During the evaluations, the RMSE and MAPE metrics were calculated. The results are presented in Fig. 6, where the *Score* was shown as a function of the prediction horizon.

It can be seen that the use of CTPM has significantly improved the original forecasts for all three base prediction models. Fig. 6a and b clearly show that the change in the *Score* functions is quite similar for the

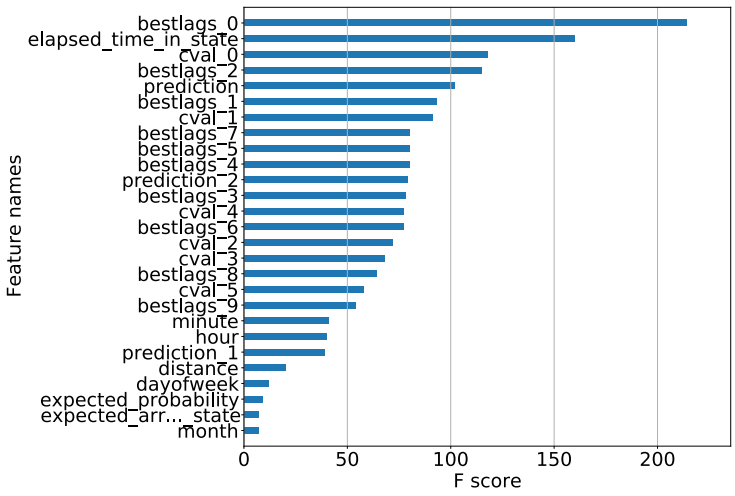


(a) RMSE values achieved by the prediction model as a function of the prediction horizon.

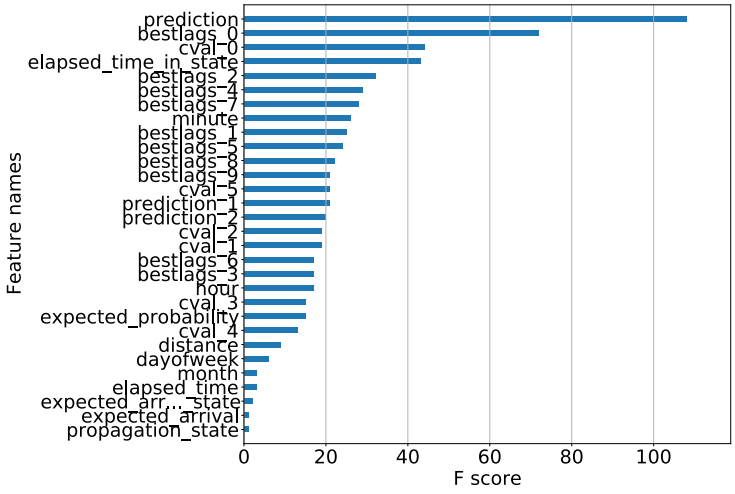


(b) MAPE values achieved by the prediction model as a function of the prediction horizon.

Fig. 7. Results for the two error metrics as a function of the prediction horizon.



(a) Feature importance of HA-based 5-minute forecast



(b) Feature importance of XGB-based 5-minute forecast

Fig. 8. Visualizing feature importance.

two metrics. In no case has there been a negative *Score*, and the lowest improvement globally was 4.9% for RMSE and 3.6% for MAPE.

In case of the *RMSEScore*, we achieved an average improvement of 18.13% using our model, which may seem outstanding. In reality, the results of the HA model are the ones raising the average. The improvement for the HA model was 34.86%, while in the case of the LSTM and XGB models it was 8.9% and 10.65%. Nevertheless, these values remain a significant improvement in the accuracy of the forecast.

In case of the *MAPEScore*, we achieved similarly good results. The average improvement was 13.74% because the improvement for the HA model was only 22.22%. When compared to the *RMSEScore*, the results for LSTM and XGB models did not change significantly. We measured 8.75% and 10.23% *Scores*, respectively.

The best results were achieved along the 5-min horizon. Considering the *RMSE Score*, here we were able to improve the results of the LSTM by 18.52% while improving XGB's output by 12.3%. The greatest improvement was also achieved with the HA model, where a 77.1% *RMSEScore* was measured.

CTPM was also able to improve the output of base models in the longer term. In the case of the 60-min horizon it was able to achieve a 1020% *RMSEScore*, with minor fluctuations for all models. An interesting phenomenon is that at the 60-min horizon there is significantly less improvement in the case of the HA model than at the 5-min horizon. This phenomenon is also observable in the case of the *MAPEScore*.

To better understand the phenomenon in Fig. 7 we plotted the RMSE (Fig. 7)a and MAPE (Fig. 7b) values separately for each base prediction model. On the same figures, the RMSE and MAPE values for the modified results were drawn with a dotted line. In the case of HA, it can be observed that at the original 5-min prediction the base model performs very poorly, but in cases with longer prediction horizons it beats even the more complex prediction models.

Looking at the RMSE and MAPE values in general, it can be said that the RMSE and MAPE values of CTPM (dashed lines) improved the original predictions by an almost constant value, regardless of the model used or the prediction horizon. Comparing to the curves of Fig. 7a, it can be seen that at the 60-min horizon, our prediction model reached an RMSE value of 0.066 using the HA prediction, while LSTM and XGB achieved only 0.078 and 0.075 even with using our prediction model.

The great advantage of decision tree-based models is that the results of the models obtained at the end of training can be easily interpreted thanks to the simple structure. Taking advantage of this, we examined which features the CTPM takes into account with greater weight.

In Fig. 8 we show the different importance of the features at the 5-min prediction horizon for the HA (Fig. 8)a and XGB (Fig. 8b)

prediction models. The importance was determined by the number of times a feature appeared in the trees. The more it appears, the more times it was able to provide the best split during training.

By comparing the two feature importance figures we find a similar ranking. In both cases, the defining features are either *bestlags*, *cvals*, or *prediction* features. It took us by surprise that in both cases one of the most defining features was *elapsed_time_in_state*. After that came the other congestion and time features, which, albeit with a lower weight, still affected the output.

6. Conclusion

In today's smart cities, accurate traffic forecasts play a key role in traffic management and route planning. The performance of the currently used prediction models is satisfactory under general conditions, but it is also necessary to take into account external data sources in the prediction process if we want to reduce the prediction error due to uncertainty.

In this article, we presented the Congestion-based Traffic Prediction Model (CTPM), which can integrate congestion propagation data as an exogenous data source. The structure of the CTPM and the features used have been described in detail. The developed model can be used in conjunction with any previous model, so there is no need to replace well-functioning methods.

The performance studies were carried out along different prediction horizons and base prediction models. Our studies show that the use of congestion data in this way significantly improve the accuracy of the LSTM and XGB models; in the studied cases the average improvement was 9.76%.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Outlook, Seamless transport for greener growth, Organisation for Economic Co-operation and Development 2015 (accessed June 20 (2012)).

[2] L. Zhu, F.R. Yu, Y. Wang, B. Ning, T. Tang, Big data analytics in intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 20 (1) (2018) 383–398.

[3] I. Lana, J. Del Ser, M. Velez, E.I. Vlahogianni, Road traffic forecasting: Recent advances and new challenges, IEEE Intell. Transp. Syst. Mag. 10 (2) (2018) 93–109.

[4] A.M. Nagy, V. Simon, Survey on traffic prediction in smart cities, Pervasive Mob. Comput. 50 (2018) 148–163.

[5] N. Zhong, J. Cao, Y. Wang, Traffic congestion, ambient air pollution, and health: Evidence from driving restrictions in Beijing, J. Assoc. Environ. Resour. Econ. 4 (3) (2017) 821–856.

[6] M. Rosenlund, F. Forastiere, M. Stafoggia, D. Porta, M. Perucci, A. Ranzi, F. Nussio, C.A. Perucci, Comparison of regression models with land-use and emissions data to predict the spatial distribution of traffic-related air pollution in Rome, J. Exp. Sci. Environ. Epidemiol. 18 (2) (2008) 192–199.

[7] X. Tian, H. Dai, Y. Geng, J. Wilson, R. Wu, Y. Xie, H. Hao, Economic impacts from pm2. 5 pollution-related health effects in china's road transport sector: A provincial-level analysis, Environ. Int. 115 (2018) 220–229.

[8] O.K. Kurt, J. Zhang, K.E. Pinkerton, Pulmonary health effects of air pollution, Curr. Opin. Pulmonary Med. 22 (2) (2016) 138.

[9] K. Chen, A. Schneider, J. Cyrys, K. Wolf, C. Meisinger, M. Heier, W. von Scheidt, B. Kuch, M. Pitz, A. Peters, et al., Hourly exposure to ultrafine particle metrics and the onset of myocardial infarction in Augsburg, Germany, Environ. Health Perspect. 128 (1) (2020) 017003.

[10] H. Xiong, A. Vahedian, X. Zhou, Y. Li, J. Luo, Predicting traffic congestion propagation patterns: A propagation graph approach, in: IWCTS@ SIGSPATIAL, 2018, pp. 60–69.

[11] W. Liu, Y. Zheng, S. Chawla, J. Yuan, X. Xing, Discovering spatio-temporal causal interactions in traffic data streams, in, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 1010–1018.

[12] H. Nguyen, W. Liu, F. Chen, Discovering congestion propagation patterns in spatio-temporal traffic data, IEEE Trans. Big Data 3 (2) (2016) 169–180.

[13] A.M. Nagy, V. Simon, Traffic congestion propagation identification method in smart cities, INFOCOMMUNICATIONS JOURNAL 13 (1) (2021) 45–57.

[14] A.M. Nagy, V. Simon, A novel congestion propagation modeling algorithm for smart cities, Pervasive Mob. Comput. 73 (2021) 101387.

[15] B. Kuhn, K. Balke, N. Wood, J. Colyar, Active traffic management (atm) implementation and operations guide, Tech. rep., United States. Federal Highway Administration (2017).

[16] Z.S. Qian, R. Rajagopal, Optimal dynamic parking pricing for morning commute considering expected cruising time, Transp. Res. Part C: Emerg. Technol. 48 (2014) 468–490.

[17] J. Zeng, J. Hu, Y. Zhang, Adaptive traffic signal control with deep recurrent q-learning, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1215–1220.

[18] A. Haydari, Y. Yilmaz, Deep reinforcement learning for intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. (2020).

[19] Y. Bie, T.Z. Qiu, C. Zhang, C. Zhang, Introducing weather factor modelling into macro traffic state prediction, J. Adv. Transp. 2017 (2017).

[20] M. Papageorgiou, I. Papamichail, A. Messmer, Y. Wang, Traffic simulation with metanet, in: Fundamentals of traffic simulation, Springer, 2010, pp. 399–430.

[21] A. Koesdwiady, R. Soua, F. Karray, Improving traffic flow prediction with weather information in connected cars: A deep learning approach, IEEE Trans. Veh. Technol. 65 (12) (2016) 9508–9517.

[22] C.-H. Chou, Y. Huang, C.-Y. Huang, V.S. Tseng, Long-term traffic time prediction using deep learning with integration of weather effect, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 123–135.

[23] Z. Gao, X. Yang, J. Zhang, H. Lu, R. Xu, W. Diao, Redundancy-reducing and holiday speed prediction based on highway traffic speed data, IEEE Access 7 (2019) 31535–31546.

[24] X. Luo, D. Li, S. Zhang, Traffic flow prediction during the holidays based on dft and svr, J. Sens. 2019 (2019).

[25] S. Yang, S. Qian, Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents, IEEE Intell. Transp. Syst. Mag. 11 (3) (2019) 12–28.

[26] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc.: Ser. B (Methodol.) 58 (1) (1996) 267–288.

[27] S. Wang, X. Zhang, J. Cao, L. He, L. Stenneth, P.S. Yu, Z. Li, Z. Huang, Computing urban traffic congestions by incorporating sparse gps probe data and social media data, ACM Trans. Inform. Syst. (TOIS) 35 (4) (2017) 40.

[28] S. Wang, F. Li, L. Stenneth, S.Y. Philip, Enhancing traffic congestion estimation with social media by coupled hidden markov model, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2016, pp. 247–264.

[29] Y. Yang, Y. Xu, J. Han, E. Wang, W. Chen, L. Yue, Efficient traffic congestion estimation using multiple spatio-temporal properties, Neurocomputing 267 (2017) 344–353.

[30] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[31] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[32] J. Tang, H.R. Heinimann, A resilience-oriented approach for quantitatively assessing recurrent spatial-temporal congestion on urban roads, PLoS One 13 (1) (2018) e0190616.

[33] T. Jayalakshmi, A. Santhakumaran, Statistical normalization and back propagation for classification, Int. J. Comput. Theory Eng. 3 (1) (2011) 1793–8201.

[34] P. developers, Pandas python package (2021). https://pandas.pydata.org/ (Online; accessed 8-April-2021).

[35] S. learn developers, Scikit-learn package (2021). https://scikit-learn.org/stable/ (Online; accessed 8-April-2021).

[36] K. Team, Keras documentation: Keras api reference (2021). https://keras.io/api/ (Online; accessed 8-April-2021).

[37] X. developers, Xgboost python package (2021). https://xgboost.readthedocs.io/en/latest/python (Online; accessed 8-April-2021).

[38] Caltrans pems (2020). http://pems.dot.ca.gov/ (Online; accessed 17-March-2020).

[39] Y. Kamarianakis, P. Prastacos, Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches, Transp. Res. Rec. J. Transp. Res. Board 2003 (1857) 74–84.

[40] Waze, Routing server - waze https://wiki.waze.com/wiki/Routing_server#Routing_requests (accessed: 2017-11-13).

[41] B.L. Smith, B.M. Williams, R.K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, Transp. Res. Part C: Emerg. Technol. 10 (4) (2002) 303–321.

[42] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[43] H. Yu, Z. Wu, S. Wang, Y. Wang, X. Ma, Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks, arXiv preprint arXiv: 1705.02699 (2017).

[44] R. Fu, Z. Zhang, L. Li, Using lstm and gru neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 324–328.

[45] Z. Zhao, W. Chen, X. Wu, P.C. Chen, J. Liu, Lstm network: a deep learning approach for short-term traffic forecast, IET Intel. Transport Syst. 11 (2) (2017) 68–75.

[46] X. Dong, T. Lei, S. Jin, Z. Hou, Short-term traffic flow prediction based on xgboost, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), IEEE, 2018, pp. 854–859.

[47] Z. Mei, F. Xiang, L. Zhen-hui, Short-term traffic flow prediction based on combination model of xgboost-lightgbm, in: 2018 International Conference on Sensor Networks and Signal Processing (SNSP), IEEE, 2018, pp. 322–327.
[48] Y. Qu, Z. Lin, H. Li, X. Zhang, Feature recognition of urban road traffic accidents based on ga-xgboost in the context of big data, IEEE Access 7 (2019) 170106–170115.
[49] Scikit-learn, Scikit-learn, gridsearchcv (2021). https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (Online; accessed 8-April-2021).