Abstract—Recently, efforts have been made to standardize signal phase and timing (SPaT) messages. These messages contain signal phase timings of all signalized intersection approaches. This information can thus be used for efficient motion planning, resulting in more homogeneous traffic flows and uniform speed profiles. Despite efforts to provide robust predictions for semi-actuated signal control systems, predicting signal phase timings for fully-actuated controls remains challenging. This paper proposes a time series prediction framework using aggregated traffic signal and loop detector data. We utilize state-of-the-art machine learning models to predict future signal phases’ duration. The performance of a Linear Regression (LR), Random Forest (RF), a light gradient-boosting machine (LightGBM), a bidirectional Long-Short-Term-Memory neural network (BiLSTM) and a Temporal Convolutional Network (TCOV) are assessed against a naive baseline model. Results based on an empirical data set from a fully-actuated signal control system in Zurich, Switzerland, show that state of the art machine learning models outperform conventional prediction methods.

Index Terms—Signal phase and timing (SPaT), time series forecasting, supervised learning, actuated traffic signal control.

I. INTRODUCTION

DIGITIZATION has substantially transformed the transportation sector over the past decade. The availability of several new data sources e. g., sensor and in-vehicle technologies enables data-driven methods to be integrated into established traffic management systems. In addition, new developments such as vehicle-to-infrastructure (V2I) communications open the possibility for new methodologies that utilize infrastructure data for motion planning, speed advisory systems, or route choice [1]. Recent developments in traffic signal control at urban intersections e. g., fully-actuated signal control [2] or self-steering algorithms [3], affect signal phasing and result in different green, red, and cycle times. Therefore, it would benefit speed advisory systems if the duration of a future signal phase is known. Ideally, fewer vehicles have to stop when crossing an intersection and uncertainty for other transportation modes is reduced. Signal phasing and timing (SpaT) messages provide the necessary information. Unfortunately, determining the future phase duration of fully-actuated signal control systems is not trivial reverse engineering as such systems typically involve complex optimization and signal phases are dependent on Loop Detector (LD) detections.

In this paper, we propose a methodology to forecast the duration of the following red phase i.e., when a traffic stream is not allowed to cross the intersection. By providing an accurate prediction for the next red-phase in this work denoted as the Time-to-Green (T2G) with Machine Learning (ML) models, we can enhance SpaT messages. We utilize empirical traffic signal and LD data based on previous work [4], and compute domain-specific features for time series forecasting. We first introduce a simple reference model no-change, a dummy model that utilizes the duration of the last occurring red phase to justify the use of ML. Then, a selection of complex models found to be strong candidates for various ML problems are compared against the dummy case. Finally, we implement a Random Forest (RF) regressor, a light gradient-boosting machine (LightGBM), and a Temporal Convolutional Network (TCOV). The proposed framework allows for a) the usage of empirical traffic signal and LD data, b) extensive feature engineering, and c) the assessment of supervised machine learning models for phase predictions. A numerical experiment in Zurich, Switzerland, is conducted to prove the concept.

An accurate prediction of the T2G can not only help the improvement of speed-advisory systems but consequently also have an impact on the homogeneity of traffic flow in multi-modal urban transportation networks. We address the opening challenges for T2G predictions by providing the following contributions:

1) The framework design allows for predicting the next signal phase of multi-modal fully-actuated signal control systems. The work is based on an empirical data set allowing for real-time applications.
2) The prediction of the next red phase, modeled as a supervised learning problem, captures the complex and non-linear relation between a traffic signal and LD detection data.

3) The framework requires no prior knowledge about the implemented traffic signal control system. Hence, the method also provides accurate predictions where the signal control algorithm is proprietary.

4) A feature engineering process that incorporates the concepts of traffic flow theory. The approach enhances the quality of a given prediction model and can be used for multi-model systems with transit priority.

The remainder of this paper is organized as follows: Section II provides an overview of recent research on the prediction of signal phasing and timing. Besides, due to the limited contributions in this area utilizing ML techniques, we provide an overview of publications applying such techniques to similar transportation problems. In Section III the time series problem is defined. Section IV describes the utilized LD and traffic signal data and the feature engineering. Furthermore, the framework definition, the selected models, and the performance metrics are introduced. Section V shows the applicability of all models based on a case study with a detailed presentation of prediction results and a final performance evaluation. Finally, a discussion, conclusion, and proposal of future work are given in Section VI and Section VII.

II. RELATED WORK

Recently, efforts were made to standardize SPaT messages [5]. Such messages contain the current phase with a prediction for the corresponding phase duration for all signalized intersection approaches. Hence, SPaT information allows for a more efficient and environmentally friendly motion planning of human-driven and/or autonomously operated individual or public transport vehicles. Especially in urban areas, this would lead to more homogeneous traffic flow, a smoother speed profile (i.e., the absence of speeding and heavy breaking between traffic lights) or an improvement in ride comfort [6]. In this section, we first present methods for obtaining SPaT estimation/predictions (Section II-A) and continue with related work that specifically focus on transportation problems with ML applications (Section II-B).

A. Prediction Methods for SPaT Information

Most of the existing methods to obtain SPaT information for semi-actuated traffic signal control systems are based on aggregated trajectory data. In such methods, signal timings are unknown and can either be fixed or change slowly in time. They use estimation approaches based on floating car data [7], [8], [9], [10] or travel time measurements collected with wireless traffic sensors [11]. For example, [7] and [8] employed a queue discharging model to estimate the start of green signals based on aggregated low-frequency bus and probe data. Yu et al. [10] formulated the SPaT estimation problem into a general approximate greatest common divisor problem, aiming to obtain the cycle lengths, green times, and the phase schemes based on historical sparse taxi trajectories. Protschky et al. [12] used a Bayesian learning approach to reconstruct the cycle length from historical trajectory data for traffic signals where the cycle length is fixed within a certain period. These methods typically rely on the underlying assumption that the cycle length is fixed, although some of them e.g., [8] and [12] are able to identify the occasional changes in the traffic signal timing plan.

Other studies propose probabilistic methods [13], [14], [15] and ML techniques [4], [16], [17] that can be used to predict the SPaT information for actuated and adaptive traffic signals. Compared to the estimation problems above, here, historical traffic signal timings are available. Protschky et al. [16] employed a Kalman Filter (KF) to estimate the probability of phase switches at each time step using historical traffic signal data. This work was further enhanced to consider implementation factors such as latency and data losses [13]. Based on historical signal data, [14] estimated the conditional distribution of each signal phase given the real-time signal phase measurements and predicted the phase duration as the conditional expectation and the confidence interval. These methods treat the SPaT information as a time series and are expected to yield satisfactory prediction accuracy if the variance of the signal phase duration is small. However, in cases where the signal phase duration changes drastically i.e., with high variance, these methods may not yield the best results, as they cannot incorporate relevant vehicle detection information. [15] took an initial step to establish the relationship between real-time vehicle information and traffic signal timings. Based on historical floating car and bus trajectory data, first, the approach calculates the joint distribution of the driving speed and the distance to the stop line, given a particular signal state green or red. Afterward, the phase duration is predicted using a Maximum A Posteriori (MAP) estimation. This work only links signal state to the information of an individual vehicle at a single time step. However, in reality, many detectors can contribute to the signal timings in a complex signalized intersection with multiple approaches and movements. Finally, previous work from the authors in [4] shows a preliminary ML approach with traffic signal and LD data. Non-aggregated raw data is utilized for predicting the T2G without extensive feature engineering. Also, the set of ML models is tested on a small data set, leading to an overoptimistic result for the LSTM models.

B. Machine Learning Based Methods for Similar Problems

Despite the lack of literature on ML-based SPaT prediction for actuated and adaptive signals, ML-based methods have been widely applied to many transportation research topics. Here, we present a short literature review on ML applications on similar prediction problems. Interested readers can refer to [18] and [19] for comprehensive surveys. Two important attributes characterize the SPaT prediction problem: First, it is a prediction problem aiming to obtain a future traffic signal state using historical data. Due to the uncertainty of the future arrivals, i.e., the actuation of the detectors, there is uncertainty in the future signal state. Second, it aims to
establish the relationship between the traffic signal state and detector information.

Within the same family of problems, we can find the short-term prediction of traffic variables, including traffic states (e.g., flow, speed, occupancy), demand (e.g., origin-destination matrix), and accident rates. Such problems are typically formulated as a time-series prediction problem where future variables are predicted from historical ones. Conventional parametric methods, such as Auto-Regressive Integrated Moving Average (ARIMA) and KF, can achieve good performance when the traffic variations are regular. To handle more general traffic scenarios, many ML models have been adopted, such as k-Nearest Neighbor (k-NN) [20], multivariate regression [21], [22], [23], Support Vector Regression (SVR) [24], [25], [26], RF [27], and deep learning methods [28], [29], [30], [31].

It is non-trivial to compare the performance of the proposed methods as these methods are developed and evaluated based on different data sets with specific features. Nevertheless, results indicate that the deep neural networks can outperform other ML methods with sufficient training data [30], [32].

Another traffic problem that shares some similarities with the SpAt prediction problem is the prediction of driver behaviors. This problem links the behaviors of the drivers (e.g., acceleration rates) with the current traffic scenarios (e.g., vehicle position and speed). In addition to the traditional analytical car following and lane changing models [33], [34], [35], many works attempt to employ data driven models to capture driver behaviors based on methods such as Hidden Markov Models [36], support vector machines (SVM) [37], [38], Bayesian Filter [38], and more. Deep learning methods have also attracted much attention within the context of this research problem. For example, [39] employed a deep Convolutional Neural Network (CNN) to perform lane change prediction based on camera data [40].

This paper focuses on applying ML methods to traffic signals and LD data, the most common traffic data source in cities worldwide. Our work proposes the first framework to provide a robust prediction of the next signal phase in a multi-modal and fully-actuated control system with public transportation priority. Furthermore, our work captures complex non-linear relationships between a traffic signal and detector data by applying feature engineering based on traffic flow theory.

III. PROBLEM DEFINITION

Assume an intersection controlled by fully-actuated signal control with installed traffic lights and LDs. There are A traffic lights and B LDs, and historical states are available from all devices (i.e., a traffic light operated in a red or green phase; an LD occupied/not occupied). We denote every traffic light with the index \( i \in S \), where \( S = \{1, 2, 3, \ldots, A\} \) and every LD with the index \( j \in D \), where \( D = \{1, 2, 3, \ldots, B\} \). As fully-actuated signal controls allow for non-constant red and green times, it is essential to distinguish traffic lights by the index \( i \). Hereafter, we define a signal cycle as \( c_{i,n} \). \( n \) denotes the index of a specific cycle \( c \) of a traffic light \( i \). Every \( c_{i,n} \) starts with a red phase for \( i \) and ends when the following red phase starts. Because we consider a fully-actuated control, our definition implies that different signals \( i \) from the same intersection might operate in different cycles at the same time.

The signal states of all traffic lights and LDs during the corresponding cycle \( c_{i,n} \) are then utilized to compute a feature set \( X_n \). Aiming for the prediction of the T2G, denoted as \( \hat{Y}_i(c_{i,n}) \), the problem is formulated as \( \hat{Y}_i(c_{i,n}) = f(X_n) \). The function \( f(X_n) \) denotes the non-linear relationship between the set of input features \( X_n \) and the target \( \hat{Y}_i(c_{i,n}) \). Note that the T2G prediction corresponds to the red time of the next cycle \( c_{i,n+1} \).

IV. METHODOLOGY

A. T2G Framework and Input Data

In the following, we introduce a T2G prediction framework designed to adapt to any intersection configuration, irrespective of the number of LDs or traffic lights. The framework’s architecture is designed to function as a supervised ML problem, taking traffic signal and LD data as inputs. Its structure is not dependent on the signal control system in use, eliminating the need for any alterations. Note that the design is also flexible to incorporate additional emerging sensor data, provided that the sensor can timestamp unique detection events. The raw traffic signal and LD data functions as an input to the T2G framework. The input signals are transformed into a structured format within this step, and undefined signal states are eliminated [41]. Consequently, the quantities in the data set can be defined as follows: Let \( s_i(k, c_{i,n}) \) and \( d_j(k, c_{i,n}) \) be the signal state of a traffic signal \( i \in S \) and an LD \( j \in D \) at discrete time step \( k \), respectively. Consequently, \( s_i(k, c_{i,n}) \) is defined as follows:

\[
s_i(k, c_{i,n}) = \begin{cases} 0, & \text{if: } i \text{ is red} \\ 1, & \text{else: } i \text{ is green.} \end{cases}
\]

Note that in (1) only the red and green signal phases are considered. Other common signal indications such as the start and end of a green phase (red-yellow and yellow, respectively [42]) are considered as \( s_j(k, c_{i,n}) = 0 \). Analogously, we define the state \( d_j(k, c_{i,n}) \):

\[
d_j(k, c_{i,n}) = \begin{cases} 0, & \text{if: } j \text{ is not occupied} \\ 1, & \text{else: } j \text{ is occupied.} \end{cases}
\]

The final processed time series for all \( i \) and \( j \) are concatenated in the set \( \mathbf{R} = [s_i(k, c_{i,n})]_{i=1}^{A}, [d_j(k, c_{i,n})]_{j=1}^{B} \) which represents the non-aggregated data set. \( \mathbf{R} \) serves as an input to data aggregation and feature engineering procedure.

B. Feature Engineering

We perform data aggregation and feature engineering based on \( \mathbf{R} \). We aggregate the data by signal cycles. This approach is selected as (a) the prediction target T2G is an aggregated quantity by definition (a float value representing the duration of the next red phase) and (b) aggregated quantities are more easily accessible for traffic operators or other authorities compared to data streams with a resolution of, e.g., 1 sec. Note that this approach differs from previous works in [4] and [43], where the non-aggregated data set \( \mathbf{R} \) is utilized without any further feature engineering. In the following we integrate
basic concepts of traffic flow theory: a) signal phase duration, to infer the control actions applied to the system; b) traffic flow dynamics, helping to understand vehicle flow during a given signal phase; c) detector occupancy and last detection timestamp, used as a proxy quantity for traffic density; d) bottleneck/disruption analysis, approximated by computing a queue and congestion indicator based on the detector and signal data. We first utilize the traffic signal data $s_i(k, c_{i,n})$ to compute the red $r_i(c_{i,n})$ and green time $g_i(c_{i,n})$ of a signal $i$ operating in cycle $c_{i,n}$. Consequently, the duration of the individual signal phases can be defined as follows:

$$
r_i(c_{i,n}) = \sum_{k=1}^{K} \left(1 - s_i(k, c_{i,n})\right), \quad (3)
$$

$$
g_i(c_{i,n}) = \sum_{k=1}^{K} s_i(k, c_{i,n}), \quad (4)
$$

Note that $K$ defines the discrete time step of the last sample of cycle $c_{i,n}$. Figure 1 depicts an example of the introduced quantities. The black pulse signals denote the raw traffic signal data $s_i(k, c_{i,n})$. The computation of the red and green time for signal $i$ by utilizing (3) and (4) give $r_i(c_{1,1})$ and $g_1(c_{1,1})$. The summation $r_i(c_{1,1}) + g_1(c_{1,1})$ results in the duration of cycle $c_{1,1}$. The derivation is performed for all traffic signals in $S$ and serves as an input to forecast $Y_i(c_{i,n})$. Note that in Figure 1, no prediction is shown for $s_i(k, c_{i,n})$ as the last signal phase shows a red phase. Since we only predict the T2G, predictions of the following green phases are not considered. Nevertheless, the framework would allow for such an application.

Per definition, every $c_{i,n}$ starts with a red phase. To also utilize the temporal component of the set $R$, we compute the day, hour, minute, and second of a cycle’s starting point as separate features denoted as $D$, $H$, $M$, and $S$, respectively.

Specified by design, the signal phase timings of a fully-actuated system are directly influenced by vehicles detected at LDs. Instead of a direct, proportional change in signal timings with traffic increases, the system responds with red and green times that are influenced by the traffic state of individual and public transportation demand. Consequently, an enhancement of the next red phase’s prediction requires features based on vehicle detections. Data from LDs are of great importance as the detections transmitted to the signal control system are, in fact, key for determining future red- and green phases. Consequently, we utilize all signals $d_j(k, c_{i,n}) \forall j, k, i, n$ of detectors $j$ and compute a set of features to infer the current traffic state. Figure 2 depicts an example of the utilized signals and visually supports the feature engineering in the following. First, we compute the traffic flow when traffic signal $i$ is red or green, respectively. To determine the traffic flow based on LD data, we assume that one signal peak corresponds to one detected vehicle. This is a reasonable assumption based on the time intervals used. Hence, the arrows in Figure 2 indicate when vehicles pass a given LD, which corresponds to $d_j(k, c_{i,n})$ changing its state from 1 to 0. We denote these traffic flows during a red or green phase as $q_{i,R}(c_{i,n})$ and $q_{i,G}(c_{i,n})$. The quantities represent the summation of signal changes in the corresponding traffic signal phase defined with Iverson brackets (the function takes the value 1 if the statement is true and 0 otherwise) as follows:

$$
q_{i,R}(c_{i,n}) = \sum_{k=R}^{K} \left[d_j(k_R - 1, c_{i,n}) - d_j(k_R, c_{i,n}) = 1\right]. \quad (5)
$$

and

$$
q_{i,G}(c_{i,n}) = \sum_{k=G}^{K+G} \left[d_j(k_G - 1, c_{i,n}) - d_j(k_G, c_{i,n}) = 1\right]. \quad (6)
$$

where $K_R = r_i(c_{i,n})$ and $K_G = g_i(c_{i,n})$, i.e., the duration of the red and green phase, respectively.

Next, we compute the occupancy of a detector $j$ during a cycle $c_{i,n}$. For that, we again utilize the corresponding signal $d_j(k, c_{i,n})$ and compute the summation of time steps the detector was occupied (indicated in Figure 2 by the blue areas). The summation is then normalized by the cycle length of $c_{i,n}$, which is computed by the summation of cycles’ red and green time:

$$
o_j(c_{i,n}) = \sum_{k=1}^{K} \frac{d_j(k, c_{i,n})}{r_i(c_{i,n}) + g_i(c_{i,n})}. \quad (7)
$$

The occupancy is defined within the interval $[0, 1]$; if a detector is not occupied within a signal cycle, the occupancy

---

**Fig. 1.** Feature computation of red and green time based on $s_i(k, c_{i,n})$.

**Fig. 2.** Feature computation based on traffic signal and detector data $s_i(k, c_{i,n})$ and $d_j(k, c_{i,n})$. 
is 0; if a detector is fully occupied throughout a cycle, \( o_f(c_{i,n}) = 1 \).

Further, we want to determine information about the last detection of an LD in a signal cycle. This feature allows for inferring information about the current traffic demand at a signal. Especially for public transportation vehicles, this feature can be utilized as a proxy to provide information about the next arrival. For example, suppose a detector that only gives detection information about a bus or tram has been activated in the last cycle. In that case, the likelihood that no detections occur in the next cycle might increase, and consequently, a longer red phase might be expected. The orange areas highlight the last detection of a detector in Figure 2. We compute the time duration from the last detection until the end of a cycle \( l_j \) as follows:

\[
l_j(c_{i,n}) = \left( r_i(c_{i,n}) + g_i(c_{i,n}) \right) - v(d_j(k, c_{i,n})), \tag{8}
\]

where the function \( v(\cdot) \) computes the time stamp of the last detection in cycle \( c_{i,n} \) of detector \( j \) based on the signal \( d_j(k, c_{i,n}) \).

As last feature inputs, we compute a queue and congestion indicator denoted as \( QI_l(c_{i,n}) \) and \( CI_l(c_{i,n}) \), when traffic light \( i \) is red or green, respectively. The quantities’ definition is based on a time duration threshold of a single detector activation; i.e., if \( d_j(k, c_{i,n}) \) shows an activation lasting longer than a threshold \( p \) during a cycle \( c_{i,n} \), \( QI_l(c_{i,n}) \) or \( CI_l(c_{i,n}) \) is set to 1. Formally, this can be denoted as:

\[
QI_l(c_{i,n}) = \left[ \left| u(d_j(k, c_{i,n})) > p \land s_i(k, c_{i,n}) = 0 \right| \right], \tag{9}
\]

and

\[
CI_l(c_{i,n}) = \left[ \left| u(d_j(k, c_{i,n})) > p \land s_i(k, c_{i,n}) = 1 \right| \right], \tag{10}
\]

where function \( u \) determines the longest detection during a cycle and computes the corresponding time duration in seconds. The returned set of values from \( u \) is then thresholded with \( p \) and conditional on the state of \( s_i(k, c_{i,n}) \), the long occupation represents a queue during red light or congestion during a green light. We utilize the queue and congestion indicators to determine if a single vehicle or multiple vehicles (with small headway) occupy a detector longer than \( p \). The latter does represent a traffic state where queues/congestion is likely. However, theoretically, the occupation larger than \( p \) caused by a single vehicle can also be caused by a random phenomenon (e.g., a taxi loading/unloading passengers). Therefore, this does not necessarily represent the same traffic state as the occupation by multiple vehicles. Nevertheless, this behavior can lead to queues/congestion, so we treat these two cases identically.

Finally, we can derive the target variable \( Y_i(c_{i,n}) \). We predict the next T2G based on an input sample from the current cycle. As the T2G target value in the data constitutes the red time of the next cycle \( r_i(c_{i,n+1}) \), the target feature is simply denoted as:

\[
Y_i(c_{i,n}) = r_i(c_{i,n+1}) \tag{11}
\]

The data set combined in \( X_n \) contains all the presented features for all traffic lights \( i \), respectively. Finally, the T2G values \( Y_i(c_{i,n}) \), serving as targets for the regression problem, are added to \( X_n \) and utilized to implement the supervised learning problem with a set of machine learning models for tackling \( Y_i(c_{i,n}) = f(X_n) \).

To address the challenge of handling long-tailed distributions in \( X_n \), we employ a multi-step transformation approach. Firstly, we apply the Yeo-Johnson transformation [44], to alleviate the skewness and normalize the feature distributions. The Yeo-Johnson transformation is advantageous as it can handle both positive and negative data and effectively improves the distributional properties of the features toward conformity with Gaussian assumptions. Subsequently, to further enhance robustness to outliers, we employ a robust scaler that centers the data by subtracting the median and scales it by dividing with the interquartile range (IQR). Finally, we employ a MinMax scaler to bring all features within the range of 0 and 1. This scaling technique is valuable for ensuring feature comparability and reducing the dominance of features with larger values. By combining the scaling techniques, we achieve a comprehensive pre-processing pipeline that addresses the challenges posed by long-tailed distributions.

C. Model Selection: Naive Baseline Model

The naive model is introduced as a first baseline model, where the prediction of the next T2G is simply set to the last observed red time. Formally, this can be denoted as follows:

\[
\hat{Y}_i(c_{i,n}) = r_i(c_{i,n}). \tag{12}
\]

Such a simple forecasting approach is utilized in various research domains and also transportation [45], [46] for performance comparison of more robust forecasting models. As stated by [47], naive models should not be treated as forecasting models but rather as a benchmark to disqualify proposed prediction models that perform worse on a problem than a naive model.

D. Model Selection: Linear Regression

A simple but effective statistical technique to model the relationship between a an input and desired response is LR [48]. An LR model predicts the response variable, \( \hat{Y}_i(c_{i,n}) \), based on \( p \) predictors. It is formulated as

\[
\hat{Y}_i(c_{i,n}) = \beta_{i,0} + \sum_{j=1}^{p} \beta_{i,j}x_j(c_{i,n}) + E_i(c_{i,n}) \tag{13}
\]

where \( \beta_{i,0} \) is the intercept, \( \beta_{i,1} \) to \( \beta_{i,p} \) are the regression coefficients, and \( E_i(c_{i,n}) \) is the error term. The model parameters are estimated using the Ordinary Least Square (OLS) method with the implementation from scikit-learn [49]. Research that similarly introduces LR models within this context can be found in e.g., [50].

E. Model Selection: Ensemble Models

In this section, we first introduce the RF, a supervised ensemble learning technique based on decision trees [51]. RFs are applied in various research domains for classification and regression tasks. In this work, we utilize the RF implementation from scikit-learn [49] to predict the T2G by solving a supervised regression problem. In the implementation, samples are randomly selected with replacements from the training data set to create a bootstrap sample. Next, for all bootstrap samples, a decision tree is fit. This procedure results in a
collection of decision trees that are denoted as an RF. For a detailed mathematical background on RF and corresponding theorems and proofs, the interested reader is referred to [51].

While individual decision trees of a random forest model are trained independently, gradient boosting adds new trees sequentially during the training process, correcting the errors of previous. Gradient boosting methods have been applied to numerous research problems [52]. LightGBM is a gradient boosting framework based on decision tree algorithms and is frequently used because of improved computational complexity. Unlike traditional boosting methods, LightGBM introduces two concepts that allow for lower computational costs: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [53]. In our study, we deploy LightGBM for the T2G prediction task. Implementation details and optimizations for LightGBM are available in [53].

F. Model Selection: Deep Learning Models

The last type of models that are used in this work are deep learning models. First, we incorporated a BiLSTM neural network into the T2G framework. An LSTM model address the drawbacks of recurrent neural networks which are vanishing/exploding gradient and extending the architecture with a memory block [54]. Along with other neural network designs, LSTMs are constructed with an input, hidden, and output layer. Unique to the LSTM, the hidden layer contains memory blocks with memory cells. These cells have states that can be influenced by a forget gate, which determines which information is discarded or ‘forgotten’ over time [55]. While LSTMs are designed to maintain information from previous time steps, BiLSTMs are built to capture information from both past and future time steps, enhancing the model’s context awareness. In detail mathematical definitions of LSTMs are given in [54]. Reference [55] shows an application to a transportation problem.

The Temporal Convolutional Neural Network (TCOV) is a neural network variant specifically designed for sequence modeling. Unlike traditional CNNs that often use spatial convolution, TCOVs deploy dilated causal convolutions, ensuring that the model respects the temporal order of the data. These convolutions allow the network to achieve a larger receptive field without increasing the number of parameters or the computational burden. In our T2G framework, we incorporate TCOVs due to their capability to capture long-term dependencies in sequence data effectively. The foundational principles and architectural details of TCOVs can be found in e.g., [56]. The implementation of the BiLSTM and the TCOV in our framework is performed with TensorFlow [57].

G. Hyperparameter Tuning

Hyperparameter tuning is an essential step in a machine learning pipeline to improve the model accuracy. Therefore, we utilize the open-source library Hyperopt [58]. The framework allows defining a hyperparameter search space (listed for all the models except the LR in Table I). Then the adaptive Tree-structured Parzen Estimator (TPE) optimization algorithm [58] is utilized to sample values from the pre-specified distributions and evaluate the model for several trail runs. Every trail model is then evaluated with the specified loss function during k-fold cross-validation. Note that in both tuning procedures the MAE and Mean Squared Error (MSE) are utilized as a loss function, and the models with the best results are selected. The final chosen architectures are listed in Table I.

H. Performance Metrics

We evaluate the models on the test data set \(X_{t, test}\) with performance metrics. First, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are utilized:

\[
\text{MAE} = \frac{1}{K_{test}} \sum_{k=1}^{K_{test}} \left| \hat{Y}_i(c_{i,n+k}) - Y_i(c_{i,n+k}) \right|, \tag{12}
\]

\[
\text{RMSE} = \sqrt{\frac{1}{K_{test}} \sum_{k=1}^{K_{test}} \left( \hat{Y}_i(c_{i,n+k}) - Y_i(c_{i,n+k}) \right)^2}. \tag{13}
\]

\(\hat{Y}_i(c_{i,n+k})\) again represents the predicted T2G for a future cycle \(c_{i,n+k}\) of signal \(i\). \(Y_i(c_{i,n+k})\) is the T2G from the test data set. \(k\) is here utilized to sum the errors over all samples from the test data set, i.e., \(K_{test}\). Besides the evaluation concerning the MAE and RMSE, we introduce two additional and strict error metrics. As we want to evaluate if the prediction meets the requirements of practical applications (e.g., speed-advisory systems), we introduce the Exact Hit (EH) and the Near-Misses (NM) ratio as follows:

\[
\text{EH} = \left( \frac{\sum_{k=1}^{K_{test}} \left| \hat{Y}_i(c_{i,n+k}) - Y_i(c_{i,n+k}) \right| = 0}{K_{test}} \right) \cdot 100. \tag{14}
\]

\[
\text{NM} = \left( \frac{\sum_{k=1}^{K_{test}} \left| \hat{Y}_i(c_{i,n+k}) - Y_i(c_{i,n+k}) \right| > 0}{K_{test}} \right) \cdot 100. \tag{15}
\]
A raw data set (resolution of 1 second) of 2 months of consecutive telegram data from January and February 2019 is available. In the following analysis, we take data from all weekdays from 7:00 to 22:00 hours into account as public transportation operates on a regular schedule within this time frame.

V. Numerical Experiments

The numerical experiment is based on a historical data set from an intersection in the city center of Zurich, Switzerland [41]. The intersection depicted in Figure 3 is regulated by a fully-actuated signal control system with 12 traffic signals (indicated with circled numbers). Signals 1, 2, 4, 5, and 6 control vehicular traffic streams. Traffic signal 3 is implemented for cyclists who can travel straight ahead and is co-regulated with signal 2. Signals 7 – 10 regulate pedestrian flows. From north to south and vice versa, tram lines (indicated with dashed lines) are potentially prioritized by the signal control (signal 11 and 12). These signals only operate in a green phase when a public transportation vehicle arrives at the intersection. The case study does not focus on predicting the T2G for these signals, i.e., only results for signal 1 – 10 are presented. Section VI provides a discussion with numerical insights for signals 11 and 12. Figure 3 also depicts the five associated LDs (indicated as rectangles with the corresponding numbering at the intersection approaches).

In Figure 4, feature distributions for the red and green times of all traffic signals are presented. The violin plots present the red and green time feature distributions for all considered traffic signals. The data distributions highlight that the signal control system is fully-actuated. For example, the red time of signal 4 (shown separately in Figure 4) operates with an average red time of 38.46 seconds. Nevertheless, the minimum and maximum values in the data set show red times of 29 sec and 104 sec, respectively. Note that signal 4’s red time distribution shows a long tail due to red times higher than 70 seconds. Nevertheless, these samples are not outliers, as the red time distribution of signal 5. The determined occupancy and last detection is long-tailed, similar to the red time distribution of signal 4. This traffic light is again highlighted as the traffic light and loop detector data is influenced by public transportation. The distribution of the occupancy and last detection is long-tailed, similar to the red time distribution of signal 5. The determined occupancy usually ranges from 0.0 to 0.2 (corresponding to 0% to 20%) per cycle. Nevertheless, also high occupancy records occur that indicate congestion or other traffic incidents. Similar, last detections range from 0 to 100 seconds but again a long tail to

\[
NM = \left( \frac{1}{K_{\text{test}}} \sum_{k=1}^{K_{\text{test}}} \left[ \left| \hat{Y}_i(c_{i,n+k}) - Y_i(c_{i,n+k}) \right| \leq 2 \right] \right) \cdot 100. \tag{15}
\]

EHs are produced when the prediction model forecasts the T2G with an error of 0 seconds and NMs when the error is lower or equal to two seconds. The threshold value is chosen based on studies such as e.g., [59] and [60], that find the response time for action of Connected Automated Vehicles (CAV) is ranging from one to two seconds (and higher under certain circumstances).

A. Data Aggregation and Descriptive Analysis

First, we take the processed data set from [41] and compute the set \( R \) containing all \( x_i(k, c_{i,n}) \) and \( d_j(k, c_{i,n}) \). A detailed explanation of the procedure and the developed algorithm can be found in [41]. Afterward, we proceed to aggregate the data set to cycles. As cycles diverge for every traffic signal \( i \), we create a separate data set for every traffic signal. Every feature in such a data set is grouped by the corresponding cycles of \( c_{i,n} \). This approach allows determining the features of all remaining traffic signals and LDs during the cycles of \( i \). Consequently, we determine 10 data sets with the proposed feature set: red time, green time, traffic flow during red and green, occupancy, last detection of an LD, and the queue and congestion indicator. These features are concatenated to the vector \( X_n \) and serve as an input to the model training/testing procedure. Last, we determine the T2G \( Y_i(c_{i,n}) \) which serve as the target value for the regression problem.

Figure 5 shows the distributions of all computed features for traffic signal 5: red/green time, occupancy, last detection, flow during red/green, and congestion/queue indicator. This traffic light is again highlighted as the traffic light and loop detector data is influenced by public transportation. The distribution of the occupancy and last detection is long-tailed, similar to the red time distribution of signal 5. The determined occupancy usually ranges from 0.0 to 0.2 (corresponding to 0% to 20%) per cycle. Nevertheless, also high occupancy records occur that indicate congestion or other traffic incidents. Similar, last detections range from 0 to 100 seconds but again a long tail to
an upper limit of 500 seconds can be depicted. Note that there were around 100 samples larger than 500 seconds, which have been removed in this visualization but not in the analysis.

Finally, the flows during red/green and the queue/congestion indicators are shown as bar plots in Figure 5. Again, it can be noted that high flows occur less frequent than lower flows during a red and also green phase of signal 5. The indicators for a queue and congestion show that during a high number of cycles both indicators are 0 (i.e., no queue or congestion). As shown later in this experiment, these features do not have a significant impact on the T2G prediction.

B. T2G Prediction Results

This section applies our set of models to the training and testing procedure. We assess the model qualities by utilizing the processed data set and split it into 70% train and 30% test data, respectively. First, the naive baseline model is applied to the train data sets of traffic lights $i = [1, 10]$. As discussed, the naive baseline model is a benchmark to assess the ML models applied in the following. As the model utilizes the last red time of a signal cycle, no training or hyperparameter training procedure is performed.

For traffic lights 1 – 10, the results of the naive baseline show MAE and RMSE errors from $1.54/2.78$ sec ($i = 9$) to $8.42/13.48$ sec ($i = 10$). Also, the EH ratio is below 37.16% for all traffic lights except signals 5 and 9 (ratio of 46.42% and 54.68%, respectively). The NM ratios are below 60%, except for traffic signals 5 and 9, where 67.41% and 74.26% are computed, respectively. See Table II for all results. The highest NM ratio is computed for traffic light 9 which also shows the highest EH ratio of 54.68%. The difference in performance can be explained by the variation of the T2G values: If the signal control assigns the identical red phase multiple times throughout a certain time frame (such a system behavior can correspond to a standard program; i.e., no high traffic demand detected or arriving public transportation), the naive model predicts an exact hit with an error of 0.00 sec. Contrary, the absence of a standard control program or high variations in the T2G lead to an obvious worse performance of the naive model.

We now present the results of all the supervised learning models that are a) trained on the training data set and b) assessed on the test data set with the introduced performance metrics. Before all models are trained, the Recursive Feature Elimination (RFE) method is applied to the traffic signal data
TABLE II

| Model | MAE [s] | RMSE [s] | EH [%] | NM [%] | MAE [s] | RMSE [s] | EH [%] | NM [%] | MAE [s] | RMSE [s] | EH [%] | NM [%] |
|-------|---------|----------|--------|--------|---------|----------|--------|--------|---------|----------|--------|--------|
| Naive | 5.22    | 9.2      | 35.49  | 58.43  | 3.20    | 5.42     | 8.20   | 68.56  | 2.23    | 5.57     | 62.77  | 80.81  |
| LR    | 4.59    | 7.89     | 36.53  | 55.93  | 2.56    | 4.68     | 32.94  | 72.42  | 2.18    | 4.72     | 54.68  | 76.35  |
| RF    | 4.59    | 7.93     | 37.16  | 56.07  | 2.50    | 4.70     | 35.91  | 72.86  | 2.08    | 4.71     | 58.56  | 77.17  |
| LSTM  | 4.58    | 9.29     | 33.37  | 56.3   | 3.21    | 5.42     | 8.69   | 68.24  | 2.27    | 5.5      | 59.71  | 79.47  |
| TCOV  | 4.3     | 8.37     | 46.42  | 67.41  | 2.75    | 4.95     | 9.87   | 76.17  | 1.81    | 5.13     | 69.71  | 84.65  |
| RF    | 4.87    | 8.24     | 33.73  | 54.92  | 2.63    | 4.77     | 33.87  | 70.55  | 2.22    | 4.79     | 52.69  | 76.40  |
| LSTM  | 4.21    | 7.36     | 36.59  | 55.59  | 2.59    | 4.60     | 24.96  | 70.93  | 2.1    | 4.42     | 51.21  | 75.69  |
| TCOV  | 7.32    | 11.75    | 33.33  | 48.51  | 4.60    | 7.06     | 3.76   | 40.02  | 3.16    | 7.18     | 54.39  | 73.99  |
| RF    | 1.54    | 2.78     | 54.68  | 74.26  | 1.24    | 1.86     | 35.16  | 86.16  | 0.99    | 2.02     | 68.16  | 86.25  |
| LSTM  | 8.42    | 13.48    | 36.81  | 49.35  | 4.94    | 7.68     | 4.31   | 66.81  | 3.27    | 7.81     | 59.57  | 74.82  |

| Average | 5.05 | 8.61 | 38.36 | 57.68 | 3.02 | 5.11 | 19.70 | 66.27 | 2.22 | 5.19 | 59.15 | 74.60 |

The method assigns a weight to an input feature which functions as a proxy for feature importance. The features with minor importance are then eliminated from the data set. In our case, the feature space is reduced from 70 features to 35 features. For example, two traffic signals that regulate non-conflicting pedestrian flows within the same signal stage show a high correlation, and RFE will eliminate one of the features.

The LR models allow for predictions with MAEs for traffic lights, 1 – 10, in the range between 1.24 and 3.2 sec. The RMSE errors range from 1.86 sec up to 7.1 sec. Comparing the EH- and NM-ratios of the LR models to the naive baseline, we find that EH-ratios decrease significantly for all traffic signals, except traffic signal 6. However, for that signal no substantial improvements can be observed (an increase in EH-ratio of 0.14%). For the NM-ratios, the ratios improve between 8.76% and 16.79%. However, for traffic signal 8 and 10, the performance in NMs decreases by 8.49% and 12.54%, respectively. Note that only evaluating the LR models by the MAE would lead to an acceptable performance improvement compared to the naive model. Nevertheless, the EH- and NM ratios a) show a good performance but b) are lower compared to the RF models. On average LightGBM shows a similar performance for the MAE/RMSE with 2.28 sec/4.98 sec and a lower performance for EHs/NMs with 45.68% and 71.9%. Lastly, the results of the utilized deep learning models biLSTM and TCOV are presented. Both models are applied after the hyperparameter tuning procedure explained in Section IV-G. The activation function for the models is fixed to ReLU. For the training procedure, 200 epochs are computed with an early stopping criteria by monitoring the validation loss with a patience \( p = 5 \). The batch size is set to 64 and the MAE is utilized as a loss function.

For the RF models, the MAEs for traffic lights 1 – 10 range from 0.9 to 3.27 sec and RMSEs from 2.02 to 7.81 sec. Note that for the MAE the RF model of traffic light 3 shows the best performance in this study. Most interestingly, the RF models show for several traffic lights the highest EH and NM ratios. Only the biLSTMs (introduced below) can beat the RF models for traffic light 7, 8, and 9.

The bold values indicate the best performances in Table II.
of 2.96 sec, RMSE of 6.07 sec, and a EH and NM ratio of 51.53% and 71.52%, respectively.

TCOV performs on average best for the T2G prediction when investigating the MAE and RMSE values of 2.13 sec and 4.86 sec, respectively. For several traffic lights the lowest MAE and/or RMSE is shown. Nevertheless, as already shown for the biLSTM models, the RF models outperform TCOV when investigating the EHs and NMs. Nevertheless, the EH and NM ratios outperform LightGBM.

A final comparison of all models is shown in Figure 6. The subplots show the MAE, RMSE, EH, and NM metrics for signals 1 – 10. The TCOV models show the lowest error values with an MAE of 2.13 sec (standard deviation: 0.66 sec) and an RMSE of 4.86 sec (standard deviation: 1.57 sec). The EH-ratio of 59.14% (standard deviation: 6.27%), and an NM-ratio of 78.56% (standard deviation: 4.17%) is the lowest for the RF models.

C. Feature Importance of Random Forest Models

We implemented an analysis that highlights the relationship between an ML approach and traffic flow principles. The addition highlights the importance of using concepts from traffic flow theory (based on fluid dynamics) to infer features that contain substantially more information than just a time series of 0 and 1. Also, insights about the relationship between theoretical traffic flow concepts and practical machine learning implementations are provided. One decision tree in a random forest splits input values based on the condition of impurity. When solving a regression problem, impurity is defined as the variance and should be minimized during training. Each feature’s contribution allows for the calculation of the feature importance to solve the initial problem definition \( \hat{Y}_{i}(c_{i,n}) = f(X) \); i.e., the approximation of a function that maps the input feature vector to the T2G target values.

We compute the feature importance for the 10 most relevant features for the models of traffic signals 4 and 6, respectively. Figure 7 shows the 10 most important features for the prediction of the T2G of traffic signals 4 and 6. In the case of signal 4 \( o_{1} \) is the most important, and \( r_{10} \) is the least important feature of the presented subset (Figure 7 (a)). Figure 7 also highlights the feature importance on the intersection. The traffic stream in blue shows the signal for which the T2G prediction is computed. In red, the relevant devices (LDs or traffic signals) of conflicting traffic streams are shown, and green highlights the devices of the compatible traffic streams.

For the both signals, the most important feature appears to be the occupancy of LD 1, i.e., \( o_{1} \) detecting arriving trams from the north of the intersection area. This is expected as the T2G is highly dependent on the priority of public transportation. Additionally, for both models, the occupancy \( o_{5} \) for arriving vehicles and trams from the south is listed in
the 10 most relevant features. For the T2G prediction of traffic signal 4, the second most important feature is the green phase’s duration of signal 1, $g_1$ (non-conflicting traffic stream); for the T2G of traffic signal 6, it is the red phase duration, $r_2$. Finally, note that in both cases, the feature representing the hour of the day $H$ is important and highlights that both models find T2G patterns that depend on the time of the day.

Interestingly, all computed features introduced in Section IV appear at least once in one or the other feature importance subsets of the two presented models. On the other hand, the congestion and queue indicator $QI(c_{i,n})$ and $CI(c_{i,n})$ do not appear, and an analysis shows that the RFE procedure already eliminated these features.

VI. DISCUSSION

A. Metrics for Model Evaluation

A model for T2G predictions has to meet strong accuracy requirements. Hence, a low accuracy prediction of the following green can cause safety issues that are not acceptable in practice. Consequently, a judgment based on standard metrics such as the MAE or RMSE can lead to a good performance on average, but individual predictions might still not meet the initial requirements. Therefore, in this study, we introduced the EH and NM to evaluate models based on the forecast being identical to the target, or an error smaller or equal to two seconds, respectively.

For example, the performance concerning the MAE for the LR models (Table II) shows errors that are close to the ones for the RF models. However, the EH and NM ratio in Table II significantly improved compared to the naive baseline. On the contrary, the performance ratios of the LR models even decrease compared to the naive model. Also, an analysis of the TCOV models shows the best performance for MAE/RMSE but lower ratios for EH and NM compared to RF. This highlights the importance of EH and NM for this problem.

Although the hyperparameter tuning was carried out by assessing different loss functions, the MAE function showed the best performance concerning all presented performance metrics. For example, utilizing the mean squared error as a loss function did not improve the results. A more extensive data set might help improve generalization and performance. However, for the RF models, the loss function allows for the most accurate results. In addition, RF models are easier to fit and allow for interpretation of the model parameters (the feature importance analysis described in Section V-C).

B. Vehicle Detection After T2G Prediction

As shown in Section V-C the LD data representing trams’ detection is of great importance for the RF models. However, results also show that models sometimes fail to predict a T2G peak. One potential explanation for this behavior is detections of vehicles that occur after the prediction of the T2G. In other words, we predict the duration of the next red phase, and afterward, the corresponding phase starts. If a tram arrives at an intersection approach within this phase, the signal control system might react according to predefined conditions. As a result, the red phase can be shortened or extended (dependent on the traffic relation), and the T2G duration also changes. However, this information is only available in the next cycle. Therefore, the presented prediction models potentially miss high peaks of the T2G.

A more extensive data set or additional feature engineering could help eliminate this limitation. Additionally, works such as [14] capture this system behavior as the predictions are updated consistently during the red phase. Nevertheless, this leads to fluctuations in the T2G that are problematic for control systems (e.g., motion planning of automated vehicles). Ibrahim et al. [14] also requires data aggregation per cycle length; meaning knowledge of the cycle length must be present a priori which is only possible for semi-actuated signal control systems.

C. Prediction of T2G for Public Transportation Signals

As presented in Figure 3, traffic signals 11 and 12 are dedicated to public transportation vehicles. These traffic lights only operate in a green phase when a tram is detected and needs to pass the intersection. Hence, the average red and green times differ significantly from those of signals 1–10: For signals 11 and 12, the average red/green time are 203.12 sec/13 sec, and 214.50 sec/17.65 sec, respectively. Also, the minimum and maximum values range from 5 to 500 sec, 12 to 500 sec for the red times, and 6 to 353 sec, 4 to 320 sec for the green times. Note that the maximum allowed duration of 180 sec does not apply to these signals. Consequently, a significantly higher variance of the quantities is given, which needs to be captured by the applied prediction model. Also, the described limitation from Section VI-B that vehicle detections after the T2G prediction, i.e., occurring within the red phase we predicted the duration for, has a substantial influence on model performance. As signals 11 and 12 only operate in a green phase when a vehicle is detected, it is evident that many detections occur during a red phase.

In Table III we shortly present the prediction results of the RF model traffic signal 11. The prediction errors are high compared to the values shown for signals 1–10. Table III compiles the MAE, RMSE, EH, and NM ratio for the RF models.

As expected, the performance metrics show a significantly higher magnitude with an MAE of 85.88 and 64.02 sec, respectively. This is because the RF models can not capture the high T2G peaks that frequently occur for signals 11 and 12. Additionally, the EH and NM ratios underline the modest performance with values below 2% for all metrics in

| Signal | MAE [s] | RMSE [s] | EH [%] | NM [%] |
|--------|---------|----------|--------|--------|
| 11     | 61.56   | 92.40    | 3.21   | 8.12   |
| 12     | 54.08   | 78.61    | 3.87   | 8.33   |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Table III. The other models show similar results for traffic signal 11 and 12.

A model design for designated traffic lights that satisfies accuracy requirements, additional information such as GPS signals that provide the location of trams, or additional LDs that allow inferring location or speed is needed.

D. Utilization of T2G Predictions for Speed-Advisory Systems

T2G predictions can serve as an input to motion planning algorithms leading to a smoother speed profile and more homogeneous traffic flow as vehicles do not need to stop at an intersection. Our proposed methodology considers the complex relationship between traffic signals and LD detections to determine the T2G. Nevertheless, when motion planning algorithms consider the predictions of the T2G for determining, e.g., the speed profile to cross an intersection, the timestamp of an LD detection will also change. Consequently, the proposed models in this work that learned this temporal relationship via a historical data set (offline learning) cannot dynamically adapt to the new system behavior. As a solution, the authors suggest researching the directions of (a) online learning, which allows for learning new data patterns as they are available, or (b) meta-learning allowing ML models to learn from T2G prediction outputs and adapt to the new system behavior.

VII. CONCLUSION

This paper proposes a framework for Time-to-Green (T2G) predictions at an urban intersection to enhance the quality of Signal Phase and Timing (SPaT) messages. The problem was constructed as a time series forecast to predict the next signal phase of a fully-actuated signal control system. The framework implementation is generic and can be applied to any intersection that provides Loop Detectors (LD) and signals data. An extensive feature engineering methodology is proposed to enhance the model quality by utilizing concepts from traffic flow theory. To assess the performance of supervised learning algorithms, a Linear Regression (LR), a Random Forest (RF), a light gradient-boosting machine (LightGBM), a bidirectional Long-Short-Term-Memory (biLSTM) neural network, and a Temporal Convolutional Network (TCOV) are implemented and assessed with a set of performance metrics.

In the presented numerical experiment, the methodology was tested on an intersection in Zurich operated by a fully-actuated signal control and public transport priority. A consecutive data set of two months (traffic signal and LD data) was processed, and prediction models are assessed on the accuracy when predicting the T2G. Results show that supervised learning is a promising tools for predicting the next red phase. The lowest Mean Absolute Errors (MAE) and Root Mean Square Errors (RMSE) with 2.13/4.86 sec are achieved by applying the TCOV model. Focusing on the metrics Exact Hit (EH) and Near Miss (NM) ratio, the RF models outperform the other candidates with 59.15%/74.60%. Nevertheless, the models show limitations in predicting the T2G of traffic lights designated for public transportation due to the high variance of the target values and vehicle detections after prediction. Future work will extend the present research with the possibility of updating T2G predictions throughout the next signal phase that can serve as an input to various control systems. We will also look at the parameter tuning of the models concerning computational time. This paves the way for real-time applications. Another promising direction is to develop an algorithm that can be quickly adapted to new environments (e.g., other intersections or scenarios with different transit operations) within a few shots via meta-learning (e.g., [61]).

ACKNOWLEDGMENT

The authors would like to thank Felix Denzler from the city of Zürich for collaborating in a research project and to provide the data.

REFERENCES

[1] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, “A review of motion planning for highway autonomous driving,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 1826–1848, May 2020.
[2] X. Zheng and W. Recker, “An adaptive control algorithm for traffic-actuated signals,” Transp. Res. C, Emerg. Technol., vol. 30, pp. 93–115, 2013.
[3] S. Lämmer and D. Helbing, “Self-control of traffic lights and vehicle flows in urban road networks,” J. Stat. Mech., Theory Exp., vol. 2008, no. 4, Apr. 2008, Art. no. P04019.
[4] A. Genser, L. Ambühl, K. Yang, M. Menendez, and A. Kouvelas, “Time-to-green predictions: A framework to enhance SPaT messages using machine learning,” in Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC), Sep. 2020, pp. 1–6.
[5] CAR 2 CAR Communication Consortium. Automotive Requirements for Spot and Map—Car 2 Car. [Online]. Available: https://www.car-2-car.org/fileadmin/documents/Basic_System_Profile/Release_1.5.1/C2CC\_RS\_2077\_SPATMAP\_AutomotiveRequirements.pdf
[6] A. Genser, R. Spiehler, P. Nitsche, and A. Kouvelas, “Ride comfort assessment for automated vehicles utilizing a road surface model and Monte Carlo simulations,” Comput.-Aided Civil Infrastruct. Eng., vol. 37, no. 10, pp. 1316–1334, Aug. 2022.
[7] S. A. Fayazi, A. Vahidi, G. Mahler, and A. Winckler, “Traffic signal phase and timing estimation from low-frequency transit bus data,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 1, pp. 19–28, Feb. 2015.
[8] S. A. Fayazi and A. Vahidi, “Crowdsourcing phase and timing of pre-timed traffic signals in the presence of queues: Algorithms and back-end system architecture,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 3, pp. 870–881, Mar. 2016.
[9] C. Wang and S. Jiang, “Traffic signal phases’ estimation by floating car data,” in Proc. 12th Int. Conf. ITS Telecommun., Nov. 2012, pp. 568–573.
[10] J. Yu and P. Lu, “Learning traffic signal phase and timing information from low-sampling rate taxi GPS trajectories,” Knowl.-Based Syst., vol. 110, pp. 275–292, Oct. 2016.
[11] X. Ban, R. Herring, P. Hao, and A. M. Bayen, “Delay pattern estimation for signalized intersections using sampled travel times,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2130, no. 1, pp. 109–119, Jan. 2009.
[12] V. Protschky, C. Ruhhammer, and S. Feit, “Learning traffic light parameters with floating car data,” in Proc. IEEE 18th Int. Conf. Intell. Transp. Syst., Sep. 2015, pp. 2438–2443.
[13] V. Protschky, S. Feit, and C. Linnhoff-Popien, “Extensive traffic light prediction under real-world conditions,” in Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall), Sep. 2014, pp. 1–5.
[14] S. Ibrahim, D. Kalathil, R. O. Sanchez, and P. Varaiya, “Estimating phase duration for SPaT messages,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 7, pp. 2668–2676, Jul. 2019.
[15] Y. Zhu, X. Liu, M. Li, and Q. Zhang, “POVA: Traffic light sensing with probe vehicles,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 7, pp. 1390–1400, Jul. 2013.

[16] V. Protschky, K. Wiesner, and S. Feit, “Adaptive traffic light prediction via Kalman filtering,” in Proc. IEEE Intell. Vehicles Symp., Jun. 2014, pp. 151–157.

[17] Z. Islam, M. Abdel-Aty, and N. Mahmoud, “Using CNN-LSTM to predict signal phasing and timing aided by high-resolution detector data,” Transp. Res. C, Emerg. Technol., vol. 141, Aug. 2022, Art. no. 103742.

[18] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, “Enhancing transportation systems via deep learning: A survey,” Transp. Res. C, Emerg. Technol., vol. 99, pp. 144–163, Feb. 2019.

[19] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[20] H. Chang, Y. Lee, B. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: Systematic approach based on past experiences,” JET Intell. Transp. Syst., vol. 6, no. 3, pp. 292–305, 2012.

[21] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, “Use of local linear regression model for short-term traffic forecasting,” Transp. Res. Rec., J. Transp. Res. Board, vol. 1836, no. 1, pp. 143–150, Jan. 2003.

[22] S. Clark, “Traffic prediction using multivariate nonparametric regression,” J. Transp. Eng., vol. 129, no. 2, pp. 161–168, Mar. 2003.

[23] A. Genser, N. Hautle, M. Makridis, and A. Kouvelas, “An experimental urban case study with various data sources and a model for traffic estimation,” Sensors, vol. 22, no. 1, pp. 144, Dec. 2021.

[24] W.-C. Hong, “Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm,” Neurocomputing, vol. 74, no. 12–13, pp. 2096–2107, Jun. 2011.

[25] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, “Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions,” Expert Syst. Appl., vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[26] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, “Supervised weighting-online learning algorithm for short-term traffic flow prediction,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1700–1707, Dec. 2013.

[27] Y. Zhang and A. Haghani, “A gradient boosting method to improve travel time prediction,” Transp. Res. C, Emerg. Technol., vol. 58, pp. 308–324, Sep. 2015.

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

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

[30] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, “Deep learning: A generic approach for extreme condition traffic forecasting,” in Proc. SIAM Int. Conf. Data Mining (SDM), 2017, pp. 777–785.

[31] F. Qian, G. Hu, and J. Xie, “A recurrent neural network approach to traffic matrix tracking using partial measurements,” in Proc. 3rd IEEE Conf. Ind. Electron. Appl., Jun. 2008, pp. 1640–1643.

[32] A. Koedswiady, R. Soua, and F. Karray, “Improving traffic flow prediction with weather information in connected cars: A deep learning approach,” IEEE Trans. Veh. Technol., vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[33] S. Panwai and H. Dia, “Comparative evaluation of microscopic car-following behavior,” IEEE Trans. Intell. Transp. Syst., vol. 6, no. 3, pp. 342–352, Sep. 2005.

[34] B. Ciufoo, V. Punzo, and M. Montonino, “Thirty years of Gipps’ car-following model: Applications, developments, and new features,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2315, no. 1, pp. 89–99, Jan. 2012.

[35] A. Kesting, M. Treiber, M. Schönhof, and D. Helbing, “Adaptive cruise control systems and their impact on traffic flow and string stability,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2674, no. 4, pp. 471–484, Apr. 2020.

[36] D. Gammelli, K. Yang, J. Harrison, F. Rodrigues, F. Pereira, and M. Pavone, “Graph meta-reinforcement learning for transferable autonomous mobility-on-demand,” in Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining, Aug. 2022, pp. 2913–2923.
Alexander Genser (Student Member, IEEE) received the M.Sc. degree in civil engineering and water management from the University of Natural Resources and Life Sciences (BOKU), Vienna, Austria, in 2018, and the Ph.D. degree from the Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, in 2022. As a Ph.D. student, he was a part of the Traffic Engineering and Control Group, Institute for Transport Planning and Systems (IVT), focusing on traffic optimization and traffic prediction problems on micro- and macroscopic transportation networks. His work has contributed to research projects in mobility data analysis and traffic management funded by Swiss transport authorities and municipalities in Zürich, Aarau, and Lucerne.

Kaiti Yang received the B.Eng. and M.Sc. degrees from Tsinghua University and the Ph.D. degree from the Swiss Federal Institute of Technology (ETH) Zürich. He is currently an Assistant Professor with the Department of Civil and Environmental Engineering, National University of Singapore. Prior to this, he was a Post-Doctoral Researcher with the Autonomous Systems Laboratory, Stanford University. His main research interests include the operation of future mobility systems enabled by connected and automated vehicles (CAVs) and shared mobility.

Lukas Ambühl received the B.Sc. degree in civil engineering and the M.Sc. degree in management, technology, and economics from the Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, in 2012 and 2015, respectively, and the Ph.D. degree from the Research Group of Traffic Engineering, Institute for Transport Planning and Systems, ETH Zürich. He is currently a Pioneer Fellow with ETH Zürich.

Michail A. Makridis (Member, IEEE) received the Ph.D. degree in computer vision from the Democritus University of Thrace, Greece. He is currently a Senior Research Scientist and the Deputy Director of the Traffic Engineering Group, Swiss Federal Institute of Technology (ETH) Zürich. Before, he was the scientific responsible for the Traffic Modeling Group, Sustainable Transport Unit, Joint Research Centre (JRC), European Commission (EC). His research interests include traffic management, simulation, and control for future intelligent transportation systems in the presence of connected and automated vehicles. In 2022, he received the JRC Annual Awards on Excellence in Research from the EC.

Monica Menendez (Member, IEEE) received the dual B.S. degrees (summa cum laude) in civil and architectural engineering from the University of Miami, FL, USA, in 2002, and the M.S. and Ph.D. degrees (focusing on transportation) from the University of California at Berkeley, Berkeley, CA, USA, in 2003 and 2006, respectively. Between 2010 and 2017, she was the Director of the Traffic Engineering Research Group, Swiss Federal Institute of Technology (ETH) Zürich, and prior to that, a Management Consultant with Bain & Company. She is currently the Associate Dean of Engineering for Graduate Affairs and a Professor in civil and urban engineering with New York University Abu Dhabi, Abu Dhabi, and a Global Network Professor of civil and urban engineering with the New York University Tandon School of Engineering. She is also the Director of the Research Center for Interacting Urban Networks (CITIES). She has authored or coauthored over 90 journal articles and over 200 conference contributions and technical reports in the area of transportation. Her research interests include multimodal transportation systems paying special attention to new technologies and information sources. She is an active reviewer for over 20 journals and a member of multiple editorial boards for top journals in transportation.

Anastasios Kouvelas (Senior Member, IEEE) received the Diploma, M.Sc., and Ph.D. degrees from the Department of Production and Management Engineering (Operations Research), Technical University of Crete, Greece, in 2004, 2006, and 2011, respectively, specializing in modeling, control, and optimization of large-scale transport systems. He has been the Director of the Research Group Traffic Engineering and Control, Institute for Transport Planning and Systems (IVT), Department of Civil, Environmental and Geomatic Engineering, Swiss Federal Institute of Technology (ETH) Zürich, since August 2018. Prior to joining IVT, he was a Research Scientist with the Urban Transport Systems Laboratory (LUTS), EPFL (2014–2018), and a Post-Doctoral Fellow with Partners for Advanced Transportation Technology (PATH), University of California at Berkeley (2012–2014). Before this, he was appointed as an Adjunct Professor with the Technical University of Crete in 2011 and a Research Associate with the Centre for Research and Technology Hellas (CERTH), Information Technologies Institute, Thessaloniki, Greece (2011–2012). In 2009, he was a Doctoral Visiting Scholar with the Center for Advanced Transportation Technologies (CATT), Viterbi School of Engineering, Department of Electrical Engineering, University of Southern California, Los Angeles. He has been awarded with the 2012 Best IEEE ITS PhD Dissertation Award from IEEE Intelligent Transportation Systems Society. He has been a member of the Transportation Research Board (TRB) Standing Committee (AHB15) on Intelligent Transportation Systems since 2019. He has been serving as an Associate Editor for IET Intelligent Transport Systems journal since 2017. He has guest-edited several special issues in transportation journals.