Can we Learn from Outliers? Unsupervised Optimization of Intelligent Vehicle Traffic Management Systems

Tom Mertens and Marwan Hassani ✉[0000−0002−4027−4351]

Eindhoven University of Technology, The Netherlands
tom.mertens.97@gmail.com, m.hassani@tue.nl

Abstract. Vehicle traffic flow prediction is an essential task for several applications including city planning, traffic congestion management and smart traffic light control systems. However, recent solutions suffer in outlier situations where traffic flow becomes more challenging to predict. In this work, we address the problem of predicting traffic flow on different intersections in a traffic network under the realistic assumption of having outliers. Our framework, called OBIS, applies an existing LOF-based approach to detect outliers on each intersection in the network separately. Based on the spatio-temporal interdependencies of these outliers, we infer the correlations between intersections in the network. We use these outlier-based correlations then to improve the predictability of existing traffic flow prediction systems by selecting more relevant inputs for the prediction system. We show that our framework considerably improves the performance of LSTM-based models both under outlier scenarios and also under normal traffic. We test our framework under two real-life settings. In the first, we show how improving the predictability using our framework reduces the overall delays of vehicles on an intersection with a smart traffic light control system. In the second, we demonstrate how OBIS improves the predictability of a real dataset from four trajectories of intersections in the city of The Hague. We share the latter dataset together with an implementation of our framework.

Keywords: Outlier Detection, Correlations, Dimensionality Reduction, Traffic Flow Prediction

1 Introduction

Traffic flow modelling is a broad field with many applications, such as enabling city planners to better regulate traffic in a city [16] or reducing and better managing congestion [13]. Next to this, reducing the time spent in traffic jams is time has always been in the interest of researchers and practitioners. In 2014, the US economy lost around 160 billion dollars due to this lost time [15]. Improvements in infrastructure benefit the economy and the well-being of humans. However, upgrading the road capacity by increasing the amount of lanes can be expensive and requires space, something that is often scarce in urban settings. In urban

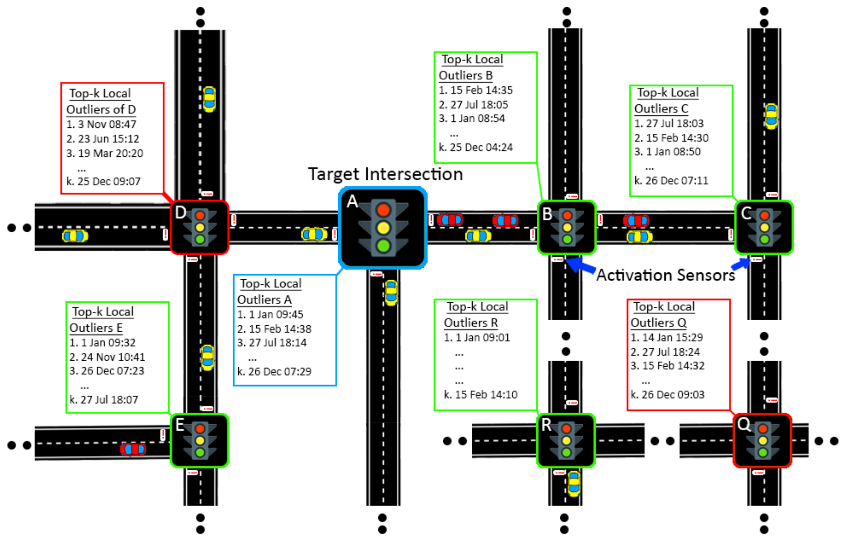


Fig. 1: A traffic network explaining the setting of the problem addressed in this work. Top-k outliers are calculated locally on each intersection. The correlation between the intersections is then checked based on the temporal correlations between their top-k local outliers within a window $w = 1$ hour before and after each top-k outlier of the target intersection.

settings, intersections managed by traffic light installations (traffic controllers) are very common, but they are not always optimal. One way of decreasing congestion and optimizing the traffic flow is to increase the efficiency of traffic controllers, the intelligent systems that control the traffic lights [8]. Ineffective traffic controllers can cause unnecessary delays (e.g. when there are less vehicles than predicted on a specific lane). These kinds of problems can cause congestion. If we can introduce more effective traffic controllers, we can reduce congestion, average travel time and average amount of stops required in an intersection, and as such create smoother and faster traffic flows.

In Figure 1, assume that the task is to predict the traffic flow on the target intersection (A). To count the real number of vehicles flowing on each lane, each intersection is equipped with several activation sensors that continuously collect these values and forward them close to the real time to a prediction model to estimate future traffic flows that are used by the intelligent traffic controller of that intersection. The model used to predict the near-future values of the traffic flow on intersection A can use previous readings on intersection A merely. Obviously this might work, but not as effective as one aims to. The dynamics in the traffic network allow for more connections between intersections. As such, including the traffic flows on neighbouring intersections B, C, D & E while predicting near-future values on A will add more context to the prediction model and should intuitively improve the model prediction accuracy. This is however not a golden rule as some intersection readings might be contributing much more noise to the prediction model than a useful input. Additionally, with slightly complex traffic networks, it becomes almost impossible to know when to stop including further intersections (e.g. is it meaningful to include the readings from the far intersections R & Q?).

The problem we address in this work is how to decide which intersections are "relevant", such that their readings should be included in the prediction model of a target intersection to maximize its accuracy. In Figure 1, those are intersections marked in green. Due to the connectivity between traffic intersections, deviating traffic situations, or outliers, in the traffic flow in intersection A can propagate to intersection B (or the other way around). We use this propagation of outliers from an intersection to another in our proposed framework to infer the correlating intersections in an unsupervised setting. For each intersection, we find its local outliers individually and in a later step we check the spatio-temporal correlations between the outliers on different intersections. In Figure 1, the local outliers found on intersection B had a high temporal correlations with the local outliers found on intersection A. Additionally B has a shorter driving time to A than a specified threshold which makes it spatially correlated to A. The outliers found on D are not temporally correlated with the ones found on A, although it is a neighbouring intersection to A. Thus, we assume that D is not a correlating intersection with A. The same applies for Q but the other way around, although the temporal correlation is satisfied, it is not included because the spatial threshold is not satisfied.

Existing traffic prediction models perform relatively well, except when they have to handle an outlier traffic situation. By relaying outlier information to the prediction model, we hope to be able to improve the general performance of traffic prediction flow models. To this end, we propose OBIS, an Outlier-Based Intersection Selection framework which aims to improve an existing intelligent traffic controller that works on real intersections. The existing traffic controller is a product of our industrial partner, Siemens Mobility.

This controller, called DIRECTOR, aims to minimize vehicle traffic delays by using an LSTM-based model for predicting the queues in front of traffic lights. However, it suffers from the limitations mentioned above as it focuses only on the previous readings of the target intersection and the intersections directly preceding it for predicting its future traffic flows. Additionally, it does not perform well under outlier situations. Since both the code of DIRECTOR and the readings from intersections are the ownership of Siemens, we additionally test our method on a large open dataset from 30 real traffic intersections in the city of The Hague collected for 2 years and 3 months and with a total of $7,093,440$ readings (cf. Table 1). We share our implementation of OBIS which contains additionally a link to The Hague dataset.

More precisely, the contributions of this work are: **(1)** we introduce a novel outlier-correlation-based method, called OBIS, for improving the predictability of traffic flows on intersections by selecting more relevant input, **(2)** we test OBIS on the prediction models of DIRECTOR, a real intelligent traffic controller, **(3)** we show that the accuracy improvements introduced when using OBIS over DIRECTOR considerably reduces the delay time of vehicles on the intersections to the half, **(4)** we additionally test OBIS on an open dataset from 30 intersections in the city of The Hague with more than 7 Million readings and show that OBIS increases the accuracy of an LSTM prediction model by 17.9% under outlier situations and by 10.3% in general, and finally, **(5)** for reproducibility purposes, we share an implementation of OBIS and The Hague dataset too.

The remainder of this paper is organized as follows: Section 2 introduces the related work. Preliminaries and some notations are introduced in Section 3 after which the main OBIS method is presented in Section 4. The applied scenarios are introduced in Section 5, and then extensively experimented and evaluated in Section 6. Section 7 concludes the paper with an outlook.

2 Related work

Density-based outlier detection is one of the most common unsupervised ways to detect outliers due to its ability to compare the local outlierness values of data points by using the reachability distance of a data point relative to those of neighbouring data points. There are two main techniques for similarity-based outlier detection and both are based on the nearest neighbours concept. The kNN (k-nearest-neighbours) algorithm and the LOF (Local Outlier Factor) algorithm [5]. There are many dialect techniques which are adapted versions of those two base methods. For example, a kNN based algorithms is: kNN-weight [3] which uses the sum of distances to reduce the variation and sensitivity to the parameter k. Outlier Detection using Indegree Number (ODIN) [9] is a graph based kNN algorithm that defines outlierness as a low number of in-adjacent edges in the graph. An example of a LOF based algorithm is INFLO (Influenced Outlierness) [10] which combats the problem of outlier estimation based on local neighbours that occurs when a dense cluster is close to a data point in a sparse cluster. It does so by considering both neighbours and reverse neighbours of a data point when estimating its density neighbourhood for the LOF. Many more examples of modified kNN and LOF algorithms exist. Research into those different techniques has shown that the original LOF and kNN were still the state of the art in the field of outlier detection [6]. The authors in [17] have designed an LOF-based model that detects outliers over Probability Distributions of traffic flows [17]. A Flow Probability Distribution (FPD) [20] is a stream of multiple values that show what proportion of the traffic happened at which time [17]. The assumption here is that traffic is distributed in certain patterns which can be learned and that a clear deviation from the pattern might mean that we have obtained an outlier. To find these outliers, the work applies the LOF algorithm to the FPDs, creating the FPD-LOF method. In the outlier detection phase of our framework, we will apply an adapted version of FPD-LOF.

Traffic flow modelling has been thoroughly researched. Currently, due to the high effectiveness on time series, LSTM-based architectures are one of the most applied solutions in this field. In [19], several prediction techniques are tested and LSTM neural networks are considered the best option. LSTMs have also been used for trajectory prediction, for all traffic participants (not just cars, but bikes, pedestrians etc. as well) [12] [1]. In [11], the authors model traffic using a Recurrent Neural Network that also applies Diffusion Convolution and incorporates random walks on the road graph, better accounting for the spatial structure of traffic modelling, finally leading to a significant increase in prediction accuracy. However, in addition to traffic flow data, this technique requires traffic speed and the distances between the intersections. Particularly the former is not available in the majority of sensor settings on intersections. [14] expected a slightly similar input but applied a hierarchical linear vector autoregressive model and a relatively deep neural network to predict traffic flow. In [18], the authors proposed a neural network based traffic prediction model to capture region-level correlations, temporal periodicity and inter-traffic correlations.

In all of the previous applications, no attention was paid on outlier-based selection of input traffic flow data to the prediction model. The information from all intersections were considered when predicting near-future flow information on any intersection in the network. Our work focuses on finding, for each target intersection, the most relevant other intersections whose traffic flow is correlating with that of the target intersection. For checking the correlations between intersections, we detect the outliers individually on each intersection using an adapted method from the one presented in [17]. Consequently, we check the temporal correlations between the outliers found on different intersections and use that to decide on the list of correlating intersections in general. We show through an extensive experimental evaluation on two real-life scenarios that this considerably improves the predictability of models that blindly include traffic input from all intersections in the network, in general but specifically during outlier situations, where most delays occur. Similar to the most related literature, this work will apply an LSTM-based model in the prediction part without claiming any contribution on the model itself. Also because an LSTM-based architecture that considers the data from the target stream data merely is already in use by Siemens for the traffic flow prediction. This paper extends our proof of concept results presented in [7] by broadening the correlation scope, applying two large real-world datasets and including real KPIs beyond the prediction accuracy.

3 Preliminaries and Notations

The traffic data used in this work are sensor data from inductive loops on any intersection from the set of intersections $\mathbb{I}$. Those sensors are activated when enough metal passes over them, such as a vehicle. Singular activations do not tell us much about the patterns in the traffic, thus these sensor measurements are aggregated every 5 minutes. The intersection that is controlled by the traffic controller is called the **Target Intersection** and let us call it A (cf. Figure 1). To control this intersection, models are trained to predict upcoming traffic. A model is trained for each road leading to the target intersection. Intersections that are along those roads form a **trajectory**, denoted as T. The intersections are then formally noted as T_B, with $B \in \mathbb{I}$ being the name of an intersection along trajectory T, and the target intersection is noted as T_A, with $A \in \mathbb{I}$. Thus, only sensors that include data relevant to the trajectory are included in the models, creating **streams** of aggregated activations (x) for each 5 minutes (h) of relevant sensors at an intersection B. This stream is denoted by x_{hB}.

Flow Probability Distributions (FPDs): to obtain representations of traffic flows that make them comparable with each other, distributions of traffic over a period of time H with a set number of time intervals h within H are created. In this work, $H = 1$ hour and $h = 5$ minutes making 12 time intervals. In short, those FPDs are sets of 12 values, with each value representing the proportion of traffic of that hour within the time interval of 5 minutes. Let $FPD(H_B)$ be the FPD for time period H on intersection B and let $X_{H_B} = \langle x_{hB1}, x_{hB2}, \cdots, x_{hB\frac{H}{h}} \rangle$ be a collection of aggregated traffic flow values x_{hB} of length $\frac{H}{h}$ for intersection B, the $FPDs$ are calculated as:

$$FPD(H_B) = \langle \frac{x_h}{\sum X_H} \rangle, \forall x_h \in X_H, h = 1, ..., \frac{H}{h} \tag{1}$$

The Bhattacharyya distance measure: to compare FPDs, a distance measure that compares two distributions should be applied. We use the Bhattacharyya distance [4]. Given two distributions $p(x)$ and $q(x)$ with $x \in X$, the Bhattacharyya distance $\mathcal{D_B}$ between $p(x)$ and $q(x)$ is defined as:

$$\mathcal{D_B}(p(x), q(x)) = -ln(BC(p(x), q(x))) \tag{2}$$

where $BC(p(x), q(x)) = \sum_{x \in X} \sqrt{p(x)q(x)}$ is the Bhattacharyya coefficient for discrete probability distributions. Other distance measures than $\mathcal{D_B}$ can be also applied [2].

The weekly intersection periodic pattern: in this work, we used the domain knowledge to decide the length of the period after which a repetitive traffic flow pattern on an intersection is expected. Intuitively, this is one week. In particular, this is suitable when we choose $H = 1$ hour. This means that, for instance, the traffic flow on a Tuesday between 9 AM and 10 AM is comparable with all of the traffic flows of Tuesdays in the same period on the same intersection. A reading deviating from the other flows on some Tuesday 9 AM to 10 AM in the measurements on a specific intersection is an indication of an outlier. As such, our task becomes to calculate for each intersection B, the distances between the weekly hour flow probability distributions using $\mathcal{D_B}$. Note that it is then to be expected that some of the found outliers are caused by holidays, days with extreme weather or special events in the city. We have purposely considered those outliers in our analysis, as we are still interested in how they correlate with other outliers on other intersections under this unusual setting. An important sub-goal of our work is to predict the traffic flow under abnormal scenarios.

Local Outlier Factor over FPDs: the LOF algorithm [5] is used to calculate outlier scores of FPDs within an individual intersection. For an FPD denoted as $\hat{f}$, let us use $reach_k$ to denote the reachability of $\hat{f}$ to $\hat{f}_k$, the k Nearest Neighbour FPD of $\hat{f}$.

The Local Reachability Distance (LRD) of $\hat{f}$ is defined as follows:

$$LRD(\hat{f}) = 1/\left(\frac{\sum_{\hat{f}_k \in kNN(\hat{f})} reach_k(\hat{f}, \hat{f}_k)}{|kNN(\hat{f})|}\right)$$

with $reach_k(\hat{f}, \hat{f}_k) = max\{\mathcal{D_B}^{kNN}(\hat{f}_k), \mathcal{D_B}(\hat{f}, \hat{f}_k)\}$ where $\mathcal{D_B}^{kNN}(\hat{f}_k)$ is the distance from $\hat{f}_k$ to its kNN for any $\hat{f}_k \in kNN(\hat{f})$. The LOF score of $\hat{f}$ is then defined as:

$$LOF(\hat{f}) = \frac{1}{|kNN(\hat{f})|} \sum_{\hat{f}_k \in kNN(\hat{f})} \frac{LRD(\hat{f}_k)}{LRD(\hat{f})} \tag{3}$$

Outlier Correlations and intersection Selection: to use these outlier scores for selecting the right intersections, correlations are determined. For each trajectory T, the Pearson correlation C_{AB} is found between the target intersection T_A and all other intersections in that T. Then, to select intersections, a correlation threshold is determined, the intersections that meet the correlation threshold are included in the prediction model. In practice, this means that the intersections that are included in the prediction model often experience similar outliers as the target intersection T_A. Additional spatial filtering is performed such that correlating intersections are considered only if they are spatially closer than ϵ to the target intersection. With ϵ being the spatial threshold for all intersections that are within a driving time of τ of the target intersection. The list of included intersections is noted as $\mathbb{I}^T$. Optimizing the traffic controller can be done by improving its main Key Performance Indicators (KPIs). For a traffic controller, the most important KPIs are the delay and the amount of stops.

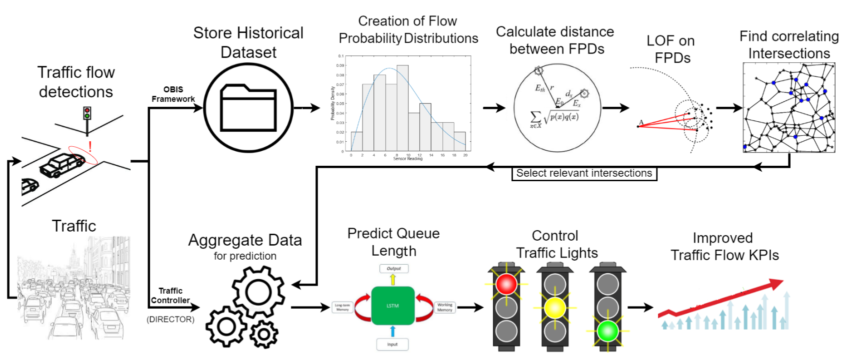


Fig. 2: The context with OBIS framework components in the upper row.

4 OBIS: Outlier-Based intersection Selection Framework

This section explains each part of the OBIS Framework for traffic controllers, as shown in the upper row of Figure 2. We will explain each component and introduce all steps taken in the main parts of OBIS framework (not the preprocessing) by referring to the psuedo-code in Algorithm 1. whose output is a list of most relevant intersections $\mathbb{I}_A^T$ to be included in the prediction model of a target intersection T_A from a Trajectory T. As per Figure 2, traffic data is recorded

Algorithm 1: The main components of OBIS

Data: T, $LOFscores_{A_H}, Threshold, \epsilon, w$
Result: $\mathbb{I}_A^T$
1 $\mathbb{I}_A^T = [\,]$ `// Initialization` ;
2 **for** ***each*** $B \in T$ **do**
3     $FPDs_B = [\,]$;
4   **for** $H \in X_{H_i}$ **do**
5         $FPDs_B += FPD(X_{H_B})$ `// Eq. 1`
6   **end**
7     $Bha\_matrix = [\,]$;
8   **for** ***each*** $FPD_i \in FPDs_B$ **do**
9   **for** ***each*** $FPD_j \in FPDs_B$ **do**
10             $Bha\_matrix_{ij} += \mathcal{D}_{\mathcal{B}}(FPD_i, FPD_j)$ `// Eq. 2`
11   **end**
12   **end**
13     $LOFscores_{B_H} = LOF(FPDs_B, Bha\_matrix)$`// Eq.3`
14 **end**
15 **for** ***each*** $B \in T$ **do**
16     $C_{AB} = Pearson(LOFscores_{B_H}, LOFscores_{A_H}, w)$;
17   `/* calculate the correlations within a window` w `before & after each top-`k `outlier of` A `*/`
18   **if** $C_{AB} > Threshold$ ***and*** $dist(A, B) \leq \epsilon$ **then**
19         $\mathbb{I}_A^T += B$;
20   **end**
21 **end**

and stored as a historical dataset, sets of 1 hour with 5 minute aggregations, so 12 values x_{hB} for a flow X_{H_B} per hour H. From there, Flow Probability Distributions, $FPDs$ are created as per Equation 1, in Line 5 of the algorithm. These FPDs are compared with regards to the Bhattacharyya distance between them (Line 10), after which the LOF algorithm is applied to find traffic flows deviating from the norm on that intersection, which receive higher LOF scores than inliers (Line 13). The Pearson correlations between those LOF scores and the LOF scores of A are then calculated to find out which intersections' outliers correlate with those of the target intersection A (Line 17). Intersections that sufficiently correlate are then selected and stored in $\mathbb{I}_A^T$ to be included included in the predictive model of the traffic controller (Line 19) if they are spatially closer than ϵ to the target intersection A.

5 Applied Scenarios

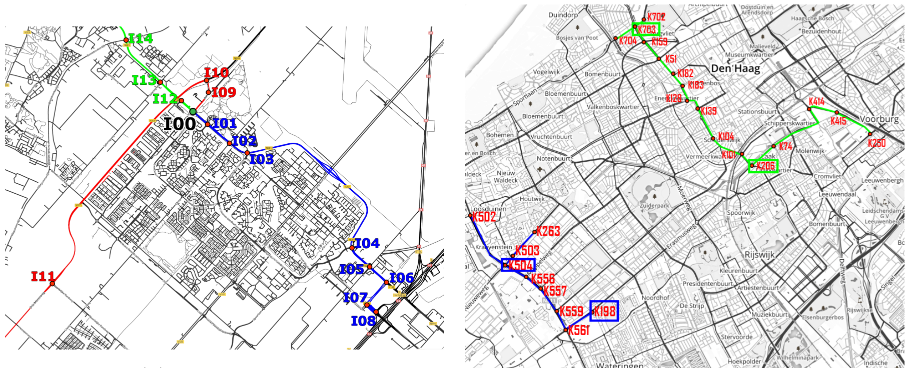
The OBIS Framework is applied to two scenarios. First with DIRECTOR, the traffic controller system which was provided by Siemens using a dataset collected from a real intersection owned by Siemens. This allows for testing with the use of a traffic simulator that can keep track of important KPIs such as the delay and the number of stops. We refer to this scenario by the DIRECTOR scenario. Second, OBIS is tested on a public dataset, provided by the city of The Hague in a fully reproducible scenario, with the code and dataset publicly provided. For this scenario, to which we refer to by The Hague scenario, only the prediction accuracy can be taken into account, as we cannot test it with the traffic simulator nor the traffic controller developed by Siemens. For both scenarios, the preprocessing and output of the OBIS algorithm are the same, as described above. For both scenarios, LSTM-based neural networks are used for prediction. The goal is to achieve a lower prediction error while using the OBIS framework to decide the input as compared to using input from: (a) all preceding intersections, (b) no other intersections or (c) merely using the directly preceding intersection (the default setting for the DIRECTOR traffic controller).

DIRECTOR Scenario: The target intersection here is called intersection $I00$. A schematic overview of $I00$ is given in Figure 3. Each white box is an available sensor. The numbers in front of the traffic lights indicate the *signal group*. This is a set of lanes that are controlled by the same signal, e.g. two lanes crossing over belong to the signal group for crossing over. Thus, for the traffic approaching from the left in Figure 3, Signal Group 03 goes straight while Signal Group 02 goes left. DIRECTOR works by predicting the queues for each lane, with 3 mod-

Fig. 3: DIRECTOR: Target intersection $I00$ schematic overview.

els for the three trajectories for each road approaching the intersection. These are the trajectories for this scenario and along these trajectories are intersections which can be included in the trajectory's model. The available intersections and the trajectories are shown in Figure 4a. Firstly, the target intersection is intersection $I00$. The first trajectory, Trajectory 0, relates to the queues for Signal

Groups 02 and 03 and is shown in green on the map, Trajectory 1 relates to Signal Groups 04 & 06 and is shown in red on the map, lastly Trajectory 2 relates to Signal Groups 07 & 08, is shown in blue on the map. Thus, three prediction



(a) DIRECTOR

(b) The Hague

Fig. 4: Trajectory & intersection maps for DIRECTOR and The Hague scenarios.

models are working simultaneously, predicting queue lengths for each trajectory, for the next 10 seconds. From these predicted queue lengths, DIRECTOR seeks to optimize a scheme for the traffic lights by minimizing delay. This scheme is simulated in a professional traffic simulator and the KPIs are recorded. The characteristics of the dataset are available in Table 1.

The Hague Scenario: The Hague Scenario is meant to provide an additional proof to the utility of the OBIS Framework and is fully reproducible. The dataset features traffic data for 30 intersections, aggregated per 5 minutes. In this scenario, only the prediction quality is assessed since the traffic controller DIRECTOR and the traffic simulator are owned by Siemens. Two trajectories are formed and traffic data from vehicles heading up and down these trajectories is preprocessed through the OBIS Framework. Figure 4b shows the intersections included and the two trajectories found. Trajectory 1 is in blue and features all intersections in the South-West, with $K198$ being the target intersection for traffic going South and $K504$ for traffic going North. Trajectory 2 is in green, with $K206$ being the Southern target intersection and $K703$ the Northern.

An overview of the datasets used in this work is given in Table 1. Not all data in the datasets could be used, as the sensors used sometimes have noise, such as no activations for a long time or extremely high activations in a short amount of time. We filtered all such noisy data from each dataset. An access to The Hauge dataset is available under the implementation link here: https://github.com/Tom-Mertens/OBIS.

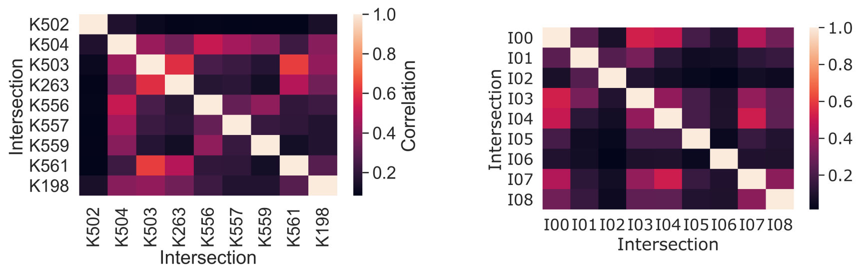
Table 1: The characteristics of the two datasets used in this work.

Name	Intersections	Start	End	Aggregation Interval
DIRECTOR	18	15-09-2019	16-09-2020	10 ms
The Hague	30	01-01-2018	31-03-2020	5 minutes

6 Experimental Results

To evaluate the performance of OBIS, several measures are used. Firstly, the accuracy of the traffic predictions is measured with the Root Mean Squared Error. Furthermore, the eventual performance of the traffic controller in the DIRECTOR scenario is measured with regards to the KPIs: (i) the delay in seconds, and (ii) the amount of required vehicle stops in the intersection. To elaborately test OBIS, it needs to be proven that using it to select intersections is beneficial to the prediction accuracy of the traffic prediction model and that this increase in accuracy can also minimize the delay and amount of stops. Firstly, the parameter tuning of the minimum correlation threshold is discussed, which regulates which intersections will be included in the eventual prediction model. Then, the models for both scenarios are discussed with a general perspective, evaluating the performance in normal settings, before diving deeper into the material and discussing the performance in outlier situations, which are most important for the traffic controller KPIs delay and stops. Lastly, these two KPIs are specifically discussed with regards to the simulated performance of the traffic controller DIRECTOR.

Parameter Tuning To determine the correlation threshold for including an intersection in the dataset, this section presents the correlations between the LOF scores within Trajectory 1 in The Hague and also those within Trajectory 2 of the DIRECTOR Target Intersection $I00$ from the DIRECTOR dataset. The correlations found in the The Hague dataset for Trajectory 1 (South) are shown in Figure 5a. Highly correlating intersections are often close to each other. While this result is not surprising, it is confirming the correctness of our concept. The intersection that barely correlates with the rest of the dataset is $K502$, which is also a bit outside of the trajectory; is only connected to the rest of the network through $K504$ (cf. Figure 4b) . The correlation heatmap of Trajectory 2 of the DIRECTOR dataset in Figure 5b is somewhat surprising. In particular, the total lack of correlation for Intersection $I02$ with any other intersection which is an intersection into the neighboring city. Further analysis indicated that this is because of the reduction in the quality of the data which might be related to noise or absence of the data during several outlier situations. We have chosen to remove data points from $I02$ due to this reduced quality. For many other intersections, the correlations do make sense, for example, $I03$ and $I04$ correlate much more with target Intersection $I00$; these are other major gateways into the city whose readings are almost complete in the dataset.



(a) The Hague Trajectory 1 South.

(b) DIRECTOR Trajectory 2.

Fig. 5: Correlations for The Hague (a) and DIRECTOR (b) scenarios.

Table 2: Correlation threshold setting using MSE for The Hague scenario.

Threshold	T 0	T 0.1	T 0.2	T 0.3	T 0.4	T 0.5	T 0.6	T 0.7	T 1.0
Trajectory 1_N	0.310	**0.215**	0.260	0.218	0.218	0.228	0.351	0.351	0.351
Trajectory 1_S	0.230	0.251	0.258	0.258	**0.153**	0.380	0.380	0.380	0.380
Trajectory 2_S	0.318	0.287	0.357	0.304	**0.271**	0.305	0.305	0.354	0.354
Trajectory 2_N	0.303	0.424	0.415	0.415	0.415	0.202	**0.196**	0.196	0.196
Average	0.29	0.29	0.32	0.30	**0.26**	0.29	0.31	0.32	0.32

To decide which intersections to include in the prediction model, a threshold correlation level needs to be selected. The higher the threshold, the fewer intersections to be considered in the model. Thus, for both datasets, many thresholds are selected and models are trained. Eventually the models which perform the best are selected. For the DIRECTOR model, prediction models are trained that are also used by the traffic controller and these are more complex and deal with time intervals of 10 seconds, while for the The Hague dataset, traffic for all incoming lanes of the target intersection is predicted per 5 minutes by the use of a simple LSTM model. The introduced errors when trying different correlation thresholds over the selected trajectories are listed in Table 2 for The Hague scenario and in Table 3 for the DIRECTOR scenario. As can be seen, a threshold of 0.4 constitutes the best results for for the The Hague scenario and a threshold of 0.35 constitutes the best results for the DIRECTOR scenario. We chose those values for the rest of our experiments. Tables 2 and 3 show some non-consistent trends wrt the threshold. This is to be expected as OBIS aims at striking a balance between the one extreme of using the readings of all intersections and the other extreme of using only the readings of the target intersection.

Evaluation under normal settings Although OBIS is meant to mostly boost performance during outlier situations, the performance in general should not suffer. Thus, for The Hague scenario, the performance is compared to the threshold

Table 3: Correlation threshold setting using MSE for DIRECTOR scenario.

Threshold	T 0.35	T 0.5	T 0.6	T 0.7	T 1.0
Trajectory 0	**0.750**	0.812	0.801	1.076	1.127
Trajectory 1	**0.651**	0.806	0.822	0.746	0.772
Trajectory 2	**0.858**	0.934	0.937	0.907	0.910
Average	**0.753**	0.851	0.853	0.910	0.936

1.0 (T 1.0) and threshold 0.0 (T 0.0) scenarios which are: adding no intersections, and adding all intersections respectively. For the DIRECTOR scenario, it is compared to the original setting of the traffic controller, adding only the intersection directly preceding the target intersection to the model. As seen in Table 2, the average errors for T 1.0, T 0.0 and T 0.4 are, respectively, 0.32, 0.29 & 0.26. Using OBIS framework in this case yields an improvement of 10.3% over the T 0.0 baseline. Thus, the T 0.4 threshold is significantly better than both baselines, while also being much more efficient than the 0.0 threshold, since less intersections are included.

For the DIRECTOR scenario, the modified director has a mean RMSE of 0.75, a slight improvement compared to the mean RMSE of the original DIRECTOR (0.79). The RMSEs for the original DIRECTOR and DIRECTOR + OBIS (with T035) are listed in Table 4.

Table 4: RMSEs for DIRECTOR and DIRECTOR + OBIS.

	DIRECTOR	**DIRECTOR+OBIS**	**Difference (%)**
Mean	0.791	0.753	4.7%
Trajectory 0	0.960	0.749	22.0%
Trajectory 1	0.661	0.648	2.0%
Trajectory 2	0.751	0.863	-14.8%

Thus, DIRECTOR + OBIS performs especially well on Trajectory 0 but is lacking on Trajectory 2. With this improved prediction accuracy, DIRECTOR should be better able to predict the length of the queue and therefore better optimize the signal schedule to them.

Evaluation under outlier setting For all 4 trajectories in The Hague scenario, the predictions of T0.4 are compared to the real values, the predictions for T1.0 (Target intersection only) and for T0.0 (all intersections) and the errors are recorded. T1.0 and T0.0 can be considered two baselines for the prediction. To see how well the model is performing during outlier situations, for each model the predictions from all three models are compared to the actual values in terms of MSE, during the 5 largest outlier situations in the test set. The results are shown in Table 5. As can be seen, the 0.4 threshold is performing much better than both baseline models. The T 0.4 threshold has a **17.9% lower error** than the best performing baseline, T 0.0. For Trajectory 2, it seems that including just the target intersection also works quite well, this might be because these are extreme outliers. On average, the T 0.4 threshold performs better.

Table 5: MSE for baseline models and T 0.4 model under top outliers setting.

Trajectory	T 1.0	T 0.4	T 0.0
T1_North	0.776	**0.371**	0.532
T1_South	0.989	**0.422**	0.720
T2_South	**0.195**	0.704	0.585
T2_North	**0.160**	0.173	0.201
Mean	0.530	**0.418**	0.509

For the DIRECTOR scenario evaluation under outlier settings, one particular outlier situation is tested and discussed, for which traffic simulations in AIMSUN 8.4 have been used. This is a three hour scenario from 15:00 - 18:00 on Tuesday 07-01-2019, where outlier traffic behaviour is heavily seen in the middle hour, which got a high LOF score. The FPD for these hours is shown in Figure 6.

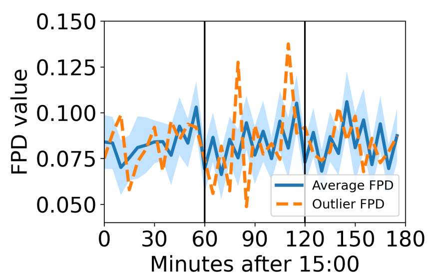


Fig. 6: FPDs for 07-01-2019 15:00 - 18:00. Outliers in middle flow.

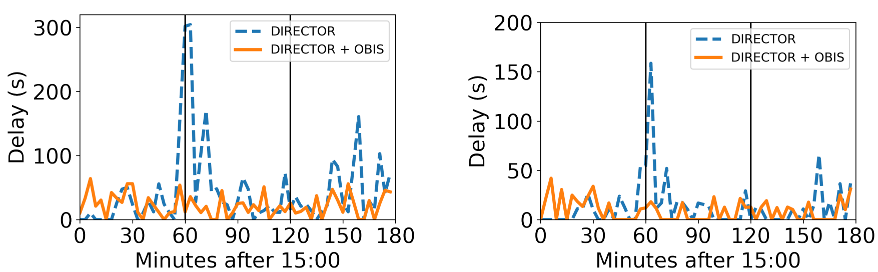
KPIs for DIRECTOR In Table 6, the upper table shows the mean aggregated delay in Seconds and the number of stops per model, the lower table sets them apart per Signal Group (SG), showing the delay in Seconds and the amount of stops.The OBIS-optimized DIRECTOR only has half the waiting time of the original DIRECTOR, which is a considerable improvement. Even though the accuracy in terms of RMSE actually decreased on Trajectory 2 (Signal Groups 7 & 8) as shown in Table 4, the performance in terms of KPIs is much better for

Table 6: Aggregate simulation results (delay in seconds), SG = Signal Group.

Means	Delay (s)	Stops
DIRECTOR	442.7	**186.5**
DIRECTOR + OBIS	**221.8**	194

SG	DIR+OBIS Delay (s)	DIR Delay (s)	DIR+OBIS Stops	DIR Stops
02	**10.3**	230.2	162.0	**156.0**
03	**355.4**	628.9	219.0	**205.0**
04	**137.7**	305.9	**171.0**	172.0
06	**401.5**	482.1	192.0	**187.0**
07	**285.5**	635.9	219.0	**197.0**
08	**140.1**	373.1	**201.0**	202.0

the OBIS-optimized DIRECTOR. The number of stops is not always reduced because of OBIS. To see how the models perform per hour, the stops and the delay are aggregated per 3 minutes and for all lanes and signal groups. The results can be seen in Figure 7a.



(a) Delay all Trajectories. (b) Delay Trajectory 1.

Fig. 7: Delay per 3 Minutes for (a) all trajectories and (b) Trajectory 1 alone.

To get a better insight, the means per hour can be found in Table 7. From the table and the figures it becomes clear that the OBIS-optimized DIRECTOR is better prepared to deal with outlier situations in terms of delay.

The outlier was on Trajectory 1, so to get a closer look at those results, Figure 7b shows the delay per 3 minutes for Signal Groups 4 and 6 which belong to Trajectory 1. DIRECTOR model incurs a large delay around 16:15, while the OBIS-optimized DIRECTOR incurs much smaller delays for that peak.

Table 7: Mean delay (D) in Seconds and stops (S) per hour for Trajectory 1.

Hour	15:00		16:00		17:00	
Delay: D (s), Stops: S	D (s)	S	D (s)	S	D (s)	S
DIRECTOR	22.8	**20.1**	66.7	**18.4**	43.3	**17.5**
DIRECTOR + OBIS	**26.8**	20.3	**17.6**	18.8	**22.3**	19.2

7 Conclusion and Outlook

In this work, we proposed the OBIS Framework, which applies an existing LOF-based approach to detect outliers on each intersection in the traffic network separately. Based on the spatio-temporal interdependencies of these outliers, we infer the correlations between intersections in the network. We use these outlier-based correlations then to improve the predictability of traffic flow prediction systems by choosing more relevant inputs to the system. We showed through an extensive experimental evaluation that our framework considerably improves the performance of LSTM-based models both under outlier scenarios and also under normal traffic. The prediction accuracy during outlier situations was improved by 19.7% over the baselines in the The Hague scenario and the delay KPI was optimized by 50% in the traffic simulation of the DIRECTOR scenario.

In the future, we would like to investigate the traffic network dynamics and/or the correlation settings that are potentially leading to the increase in the number of stops after applying OBIS to DIRECTOR. Also, we would like to see whether other outlier detection methods or distance metrics can be much more effective than the Bhattacharyya distance. Recent results [2] indicate that the Earth Movers distance is showing more promising results.

Acknowledgment

The authors would like to thank Marco Hennipman and Siemens Mobility for the support with the data, the access to DIRECTOR and the domain expertise.

References

1. Altché, F., de La Fortelle, A.: An LSTM network for highway trajectory prediction. In: ITSC. pp. 353–359 (2017)
2. Andersen, E., Chiarandini, M., Hassani, M., Jänicke, S., Tampakis, P., Zimek, A.: Evaluation of probability distribution distance metrics in traffic flow outlier detection. In: MDM. p. to appear (2022)
3. Angiulli, F., Pizzuti, C.: Outlier mining in large high-dimensional data sets. IEEE Trans Knowl Data Eng **17**(2), 203–215 (2005)
4. Bhattacharyya, A.: On a measure of divergence between two multinomial populations. Sankhyā: The Indian Journal of Statistics (1933-1960) **7**(4), 401–406 (1946)
5. Breunig, M., Kriegel, H., Ng, R., Sander, J.: LOF: Identifying Density-Based Local Outliers. SIGMOD, Dallas, Texas (2000)

6. Campos, G.O., et al.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. DMKD **30**, 891–927 (2016)
7. Fitters, W., Cuzzocrea, A., Hassani, M.: Enhancing LSTM prediction of vehicle traffic flow data via outlier correlations. In: COMPSAC. pp. 210–217 (2021)
8. Ghazal, B., et al: Smart traffic light control system. In: EECEA. pp. 140–145 (2016)
9. Hautamäki, V., Kärkkäinen, I., Fränti, P.: Outlier detection using k-nearest neighbor graph. In: ICPR (2004)
10. Jin, W., Tung, A., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. In: PAKDD (2006)
11. Li, Y., Yu, R., Shahabi, C., Lui, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: ICLR. Vancouver (2018)
12. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D.: Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. AAAI **33** (2019)
13. Pasquale, C., et al: Two-class freeway traffic regulation to reduce congestion and emissions via nonlinear optimal control. Transportation Research **55**, 85–99 (2015)
14. Polson, N.G., Sokolov, V.O.: Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies **79**, 1–17 (2017)
15. Schrank, D., Eisele, B., Lomax, T., Bak, J.: Urban mobility scorecard. In: Texas A M Transportation Institute (2015)
16. Tanaguchi, E., Thompson, R.G., Yamada, T.: Modelling city logistics. Cairns, Queensland, Australia (1999)
17. Y. Djenouri, A.Z., Chiarandini, M.: Outlier detection in urban traffic data. In: ICDM. pp. 935–940 (2018)
18. Yuan, et al: An effective joint prediction model for travel demands and traffic flows. In: ICDE. pp. 348–359 (2021)
19. Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. ITS **11**(2), 68–75 (2017)
20. Zimek, A., Djenouri, Y.: Outlier detection in urban traffic data tutorial. In: 8th International Conference on Web Intelligence. Mining and Semantics (2018)