2015 IEEE 18th International Conference on Intelligent Transportation Systems

Drift3Flow: Freeway-Incident Prediction using Real-Time Learning

Luis Moreira-Matias, *Member, IEEE*, Francesco Alesiani, *Member, IEEE*

***Abstract*— Traffic congestion is a major problem on today's urban mobility. This paper introduces a novel model for Automatic Incident Prediction (AID) on freeways: *Drift3Flow*. This stepwise methodology produces flow/occupancy rate predictions using an online weighted ensemble schema of two well-known time series analysis techniques: Autoregressive Integrated Moving Average (ARIMA) and Holt-Winters Exponential Smoothing (ETS). Then, it continuously monitors the probability distribution function (*p.d.f.*) of the prediction residuals to trigger alarms of an imminent prediction divergence, i.e. concept drift. Such alarm activates an update neuron which extends our model's reactivity by embedding a fully incremental learning schema inspired on the Delta Rule (DR) (derived from the BackPropagation (BP) algorithm). Our experimental test-bed used three weeks of data acquired from a real-world sensor network in Asia. The results validated its contributions by exhibiting a superior performance: 25% greater than the one obtained using ARIMA and ETS-based AID methods.**

I. Introduction

Today, the world's population is growing on a very high rate - especially in urban areas. Such growth is pressuring the Transportation Industry to maintain sustainable levels of urban mobility without large-scale investments. Recently, the advances on real-time data collectors through accurate sensors and cost-effective communicational infrastructure enabled the possibility of monitoring the mobility flows in a continuous and pervasive manner. Such networked sensors provide an inexpensive but massive source of human mobility information which can play a key role to accomplish such goal.

Road Traffic congestion is an effect of some sort of limitations existent on a given transportation network. It is possible to divide congestion in two types [1]: (i) *recurrent*, which happens on a regular basis within a given periodicity, e.g. peak hours on every Friday's evening, and a (ii) *stochastic* one, which is provoked by an external event, e.g.: car accidents, temporary construction activities, fast weather changes, etc.. These traffic state conditions are denominated as **incidents** [2]. After such occurrences, the traffic flow suffers a disruption which usually results on an abnormal behavior, e.g.: large travel delays. An early Incident Detection (ID) can help to restore a smooth traffic flow through Advanced Traveler Information Systems, e.g. via smartphone applications or variable message signs, or with Intelligent Transportation Systems (ITS) such as dynamic speed limits and/or lane allocation [3]. The impact of creating a reliable ID framework would be tremendous. Traffic incidents have an huge impact on our daily life by enforcing losses on distinct aspects. For example, 5 million of car accidents were reported in US during 2012 - which resulted on 34k deaths and a direct economic cost of 277$ billion U.S. dollars [4].

This paper is focused on Automatic Incident Detection (AID) for freeways. It presents a novel data driven learning algorithm named **Dynamic Drift Detection for traffic flow analysis** (*Drift3Flow*). The *Drift3Flow* is able to accurately anticipate such occurrences on a 15 minutes horizon. This stepwise Machine Learning framework relies on a real-time ensemble of two State-of-The-Art time series analysis techniques for traffic flow/occupancy prediction with a incremental learning schema derivated from the Backpropagation (BP) Algorithm [5]. This dynamic learner is triggered by a Change Detection method [6] whenever the original prediction's performance is below a predefined threshold. We demonstrate its effectiveness on a real-world dataset of traffic flow and lane occupancy rate collected by a major freeway operator in Asia throughout three non-consecutive weeks. The results are promising.

The remainder of the paper is structured as follows: Section 2 reviews the existing literature on AID. Section 3 formally presents the methodology employed. Section 4 firstly describes how the dataset was preprocessed for this task, along with some descriptive statistics. Section 5 details how the methodology was tested in a real scenario: firstly, the experimental setup and metrics used to evaluate the model are introduced; then, the results obtained are presented, followed by a brief discussion on their outcome. Finally, conclusions are drawn and topics for future work are outlined.

II. Background

AID algorithms can be folded into five categories [2], [7]: (1) traffic state/change detection; (2) data-driven event detection; (3) image-based processing, (4) traffic theory models and (5) traffic flow/occupancy prediction. The latter approach is the one followed by this paper. Yet, the reader can consult the Section 2 in [2] to know more about the other types of AID methods previously referred.

Traffic flow/occupancy prediction (5) for AID can be divided on two steps [2]: firstly, a prediction algorithm is used to infer the short-term future values of flow/occupancy based on historical data. Secondly, it compares the predicted values with a certain predefined threshold which defines the occurrence of a traffic incident and/or with the real values.

Manuscript submitted at April 15, 2015 and revised at June 19, 2015.

This work was supported by MOBiNET, a project co-funded by EC under DG Connect (FP7-ICT-2011-6.7)-N.318485.

Luis Moreira-Matias is with NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany (phone: 0049-6221-4342261; e-mail: luis.matias[at]neclab.eu).

Francesco Alesiani is with NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany (e-mail: Francesco.Alesiani[at]neclab.eu).

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.99

The model presented hereby also takes this approach by aiming to predict the future values of flow/lane occupancy on each section based only on the historical data captured by the sensor(s) on that section (i.e. single control point(s); univariate analysis; assuming statistical independence between different sensors/sections).

The literature in this topic is wide. The most commonly used methods to perform short-term traffic flow prediction include statistical filtering [8], time series analysis [9]–[12], ANNs [13], [14] and Kalman Filters (KF) [12], [15]. Among these, the most popular ones are the time series analysis methods due to their simplicity and high performance on this particular task [10]–[12]. From now on, we focus on the two main time series analysis methods employed on this type of problems: **Autoregressive Integrated Moving Average (ARIMA)** [16] and **Holt-Winters Exponential Smoothing (ETS)** [17]. A landmark reference on this topic was presented by E. Cook in 1974 [9], which showed that an ETS model could easily outperform the California algorithm [18]. The superiority of both ARIMA (using its seasonal version, i.e. SARIMA) and ETS were also demonstrated in [10], where both were compared against other non-parametric regression methods. The work in [11] provided the theoretical background on justifying why SARIMA outperformed the remaining methods on this particular application.

Recently, the State-of-the-Art on traffic flow prediction was advanced by considering hybrid models [12], [14], [19] and ensemble learning [20]. Chan *et al.* [14] used an ETS model to filter the flow signal from noise before employing an ANN (similarly to the wavelet employed in [7]). Then, the Levenberg-Marquardt algorithm [21] is employed to train such ANN. Experimental results showed that the ANN-LM methods were more robust to noise, preventing overfitting - one of the main ANN's drawbacks. Yet, the ANNs presented in this work are unable to handle novel and unobserved patterns of traffic incidents (as the training stages of these ANNs are finite). The survey work presented by Lippi *et al.* [12] suggested that the adaptive SARIMA presented in [19] is the best one on solving this task. The major advance of this method is to employ a KF to infer the unknown component of any ARIMA/SARIMA model (i.e. the moving average) incrementally. However, its formulation is *parametric*: the suggested KF formulation updates the ARIMA's models assuming the presence of *Gaussian* noise - **which may not reflect fully the stochastic nature of an traffic incident**. The *Drift3Flow* settles on an online bagging method [22] that serves as a dynamic selector of the predictive method which reflects our current system nature (independently on the noise's shape). By continuously monitoring the residual's probability distribution function (*p.d.f.*) through additive statistics, we achieve a flexible learning schema which activates a reactiveness neuron able to correct the output of our learning model sample-by-sample, whenever the individual learner's output (and consequent ensemble) is not satisfactory. By the abovementioned reasons, we believe that *Drift3Flow* meets no parallel in the existing literature of AID. This method is formally introduced in the subsequent section.

III. Methodology

The *Drift3Flow* is an online learning method for AID that settles on four steps: (A) Individual Prediction using Time Series Analysis; (B) Ensemble; (C) Update/Reaction and (D) Event Detection. These steps are introduced throughout this section. Fig. 1 presents an overview of this methodology.

A. Time Series Analysis

Let $F = \{f_1, ..., f_t\}$ and $O = \{o_1, ..., o_t\}$ be the averaged traffic flow and the lane occupancy rates on a given road section aggregated by periods of p-minutes, respectively, measured until the time instant t. Our first goal is to predict the future short-term values of these series, i.e. f_{t+1}, o_{t+1}. To do so, we suggest to employ two well-known time series analysis models: (1) ARIMA and (2) ETS, which are properly formulated in [16] and [23], respectively[1].

B. Online Ensemble

One of the main issues on working with traffic flow data is the noise due to the drivers and discretized nature of the sensors, [2], [7]. One way to smooth such noise is to employ a larger aggregation period by averaging the existing measures on larger timespans of size $P >> p$. Yet, this solution still brings two issues: (1) information loss, as we may not be able to adequately model a true peak due to our previous smoothing, and (2) low update rate due to the longer aggregation level (e.g. if $P = 30$ minutes, you will only be able to build up a prediction every 30 minutes because it must stick to bin boundaries previously defined).

One of the main ways to handle this type of problems is to perform an incremental discretization (e.g. [24]). A count f_t in an interval $[t, t+P]$ will be very *similar* to the count f_{t+1} in the interval $[t+p, t+p+P]$ (as much as $p \to 0$). We can formulate it as

$$f_{t+1} = \left(f_t \cdot {}^{P}/_{p} + f'_{[t+P, t+P+p]} - f'_{[t, t+p]} \right) \cdot \frac{1}{{}^{P}/_{p}} \qquad (1)$$

[1]Throughout this section, we describe the prediction tasks using only the notation for the flow time series to simplify its comprehension. However, the prediction schema employed for both just differs on the different thresholds/cutoffs employed.

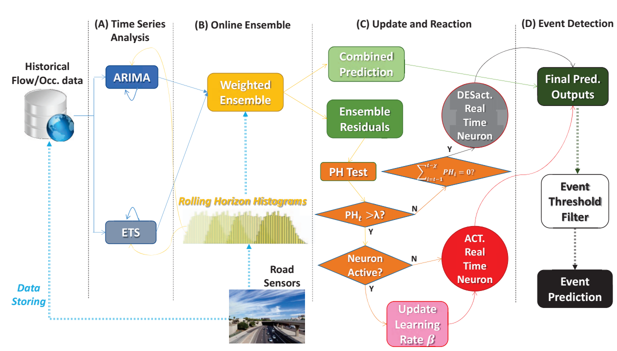

Fig. 1: Illustration of the predictive framework proposed.

where f' represents both the continuous event count on the first p-minutes of the interval $[t, t+P]$ and on the p-minutes immediately after the same period. We take advantage of the *additive* characteristics of any histogram time series to rapidly calculate a new series of interest maintaining two aggregation levels/layers: P and p.

Time series analysis methods are known to be highly effective on dealing with short-term prediction horizons [10]–[12]. However, the selection of the best algorithm is highly dependent on human expertise. Moreover, a model may be superior on some periods of the day while others may present higher performance under specific conditions (e.g. traffic jam). To handle with such variability, we propose to employ an online ensemble learning schema along with the abovementioned rolling horizon histograms. One of the most popular models of this type is bagging (e.g. [25]). Here, we apply a variant. It is now properly defined.

Consider $M = \{M_1, M_2, ..., M_l\}$ to be a set of l models (hereby, $l = 2$) of interest to model a time series and $G_t = \{g_{1t}, ..., g_{lt}\}$ to be the set of forecasted values for the next period on the interval t by those models. The ensemble forecast E_t is obtained as

$$E_t = \sum_{i=1}^{l} \frac{g_{it} \cdot (1 - \rho_{it})}{\Upsilon}, \Upsilon = \sum_{i=1}^{l} (1 - \rho_{it}) \quad (2)$$

where ρ_{it} represents the *error* of the model M_i in the periods contained on the time window $[t - H, t]$ (H is an user-defined hyper-parameter to define the window size). As the information is arriving continuously for the next periods $t, t+1, ...$, the window will also slide to determine how the models are performing in the last H periods. To calculate such error, a normalized version of the Symmetric Mean Percentage Error, i.e. $sMAPE_n$, is employed. Consequently, we can compute $\rho_{it} = sMAPE_n(i,t)^2$ as

$$sMAPE_n(i,t) = \frac{1}{H} \cdot \sum\nolimits_{t'=t-1}^{t-H} \frac{|g_{it'} - f_{t'}| + c}{g_{it'} + f_{t'} + c} \quad (3)$$

where $sMAPE_n(i,t)$ is the error produced by the model M_i in the instant t and c is a smoothing constant to handle very low values (hereby, we employ $c = 1$). Notwithstanding its validity, this schema model requires a good performance of at least one predictive model on every time instant t to produce outputs similar to the real ones. In the subsequent section, we describe how we can counter such issue with an incremental learning schema.

C. Update and Reaction

The predictive framework described above possess its own mechanisms to update itself using the novel series terms. However, those may not be enough to react on-time to a change on the underlying learning model required to perform such prediction (e.g. a smooth flow series which drops abruptly after a car crash). Such reaction time is key for AID. To overcome such limitation, we propose a two fold incremental learning framework which aims to correct our prediction values whenever they are failing to explain the nature of our time series. The first problem is to define what is a *satisfactory* performance of our model. We propose to solve it by employing an well-known Change Detection method: the Page-Hinkley (PH) test [6]. It considers a cumulative variable m_i^T, defined as the sum of the differences between the observed values and their mean till the current moment. Hereby, we use it to monitor the evolution of the prediction's residuals, i.e. $r_t = f_t - E_t$. We can define it as

$$m_t^T = \sum\nolimits_{y=t_0}^{t} (|r_t| - \bar{r}_t - \delta), \bar{r}_t = \sum\nolimits_{y=t_0}^{t} \frac{|r_t|}{T} \quad (4)$$

where δ corresponds to the magnitude of changes that are allowed (i.e. two user-defined parameters for flow/occupancy, δ_f, δ_o), t_0 stands for the test's starting timestamp and $T = t - t_0 + 1$. The test's output, i.e. PH_t^T can be obtained as

$$PH_t^T = m_t^T - \min(m_y^T, y = t_0...t) \quad (5)$$

whenever $PH_t^T > \lambda$, it throws an alarm and resets the test variables (i.e. $t_0 = t$). λ depends on the admissible false alarm rate on the change of our residual's distribution (λ is also decomposed on two for flow/occupancy, i.e. λ_f, λ_o).

The alarm triggered by the PH_t^T can be translated on a significant change on residuals *p.d.f.* (e.g. the right tail is going considerably thicker) which is provoked by a change on concept embedded on our learning model. This concept's absence force us to somehow fade our previous mid-term learning schema into something more reactive. To do it so, we propose to employ a single Perceptron's [26] neuron to force our system to learn incrementally on a fast rate. This neuron aims to correct our prediction outputs based on the newest residual available. Such reactive learning behavior is inspired on the Delta Rule (DR) [27], which is part of one the most well-known learning schemas for Feedforward Neural Networks (FNN): the BP algorithm [5]. This procedure is formally described below.

Let E_t be our predicted value while r_{t-1} stands for the newest residual available. After been triggered by PH_t^T, this neuron updates E_t as follows

$$E_t' = E_t + \Delta_{E_t}; \Delta_{E_t} = \beta \cdot r_{t-1} \quad (6)$$

where the starting value of β, i.e. β_0, is an user-defined parameter. To improve the model ability to react, the learning rate β may also be updated: if any alarm is triggered by the PH test while the neuron is activated, the β value is updated as $\beta' = \min(1.2 \cdot \beta, 1)$. This variant of the DR is denominated Exponential DR and it is commonly used to model floating concepts or their absence (e.g. concept drift on travel time prediction in [28]). Finally, the neuron is deactivated if there are χ consecutive periods for whose the inequation $PH_t^T \leq \lambda$ is true.

Note that, despite the *rolling horizon* approach's (described in Section III-B) ability to produce predictions each p minutes using an aggregation level of $P > p$ minutes, the residuals as well as all the update rules described throughout this Section follow the largest periodicity, i.e. P. Even so, this framework is designed for this specific context, where an incident is usually characterized by a sudden change on traffic flow parameters [2], [7]. By doing so, this schema outputs a single prediction value employing a **real-time learning** process. Then, these flow/occupancy rate predictions are input, along with historical data, to the event detection framework. This latter is described below.

D. Event Detection and Prediction

Typically, AID systems which employ traffic flow prediction models rely on fixed thresholds to predict the incident's occurrence beforehand. The same approach is taken here. Let $\Theta(t)$ be a binary event time series, computed as

$$\Theta(t) = \begin{cases} 1 & \textbf{if } f_t < \varphi_f \wedge o_t > \varphi_o \wedge corr(f_{t,H}, o_{t,H}) < \vartheta \\ 1 & \textbf{if } f_t < \varphi_f \wedge o_t > \varphi_o \wedge \Theta(t-1) = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

where $\varphi_f, \varphi_o, \vartheta$ are user-defined parameters for the flow, occupancy and correlation coefficient thresholds, respectively. $f_{t,H}, o_{t,H}$ stands for the H most recent data points on each time series. Then, we can define the AID framework, i.e. Θ', using the same schema by replacing the latest series terms f_t, o_t by our predictive model's output, E_t^f, E_t^o, and by increasing/decreasing the thresholds φ_f, φ_o on 10% of their original value, respectively. This entire framework was validated on real-world case study - described below.

IV. Case Study

This study was conducted using data collected through a traffic monitoring system of a major freeway deployed in an Asian country. This system both collects and broadcasts traffic-based measurements in real-time with distinct temporal granularities (depending on the type of sensor's installed on each lane). Each sensor measures traffic flow, lane occupancy rate and instantaneous vehicle's speed. Yet, just the data of the first two was used for this study. The largest time granularity of this data collection system ($p = 5$ minutes) was used to normalize all the collected time series. This step aims to establish a common comparative testbed for different sections, independently on its lane number or sampling frequency. Hence, it disables the possibility to distinguish main lanes from input/output ramps flows.

This dataset used data collected from 106 sensors which includes both freeway's transit directions. Its total length is roughly 20km while its sensors are deployed each 500m. This data was collected through 3 non-consecutive weeks.

The online ensemble method enunciated in Section III-B consider two aggregation levels, i.e. p and P. As preprocessing task, the second layer of aggregations was defined as $P = 15$ minutes. Fig. 2 exhibits the smoothing effect of including these two-layer schema (p and P on the parts A and B) due to the continuous discretization of our time series. Fig. 3 illustrate five sample-based probability distribution functions obtained using a (gaussian) kernel density estimator over all the flow measurements available - one global and four specific for each of the considered timespans (divided by Periods I-IV, identified by the same display order as Fig. 3 legend). Table I includes descriptive statistics on our dataset. The top 10 sensors regarding the number of observed incidents were highlighted. As it is observable, the

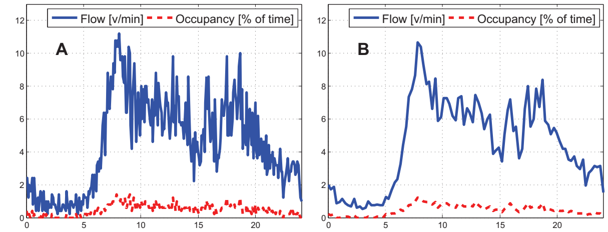

Fig. 2: Example of flow/occupancy plot with the two aggregation levels employed (A/B for 5m/15m, respectively).

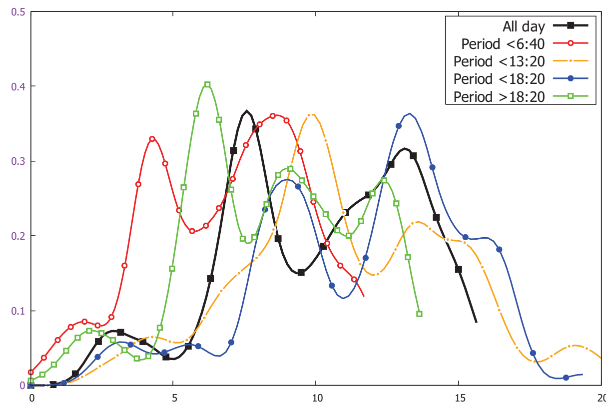

Fig. 3: Flow-Based Probability Density function estimated from the all available samples aggregated by 15m-periods.

occupancy rate is higher in these sensors. Not surprisingly. the most critical period is the morning peak, comprised between 6:40 and 13:20.

V. Experiments

We conducted our experiments using three distinct predictive methods: ARIMA (ARI), Holt-Winters Exponential Smoothing (ETS) and the hereby proposed *Drift3Flow*. All the three used the same Event Detection framework described in Section III-D. On the top of the sensor-based division described on the previous section, we also assumed statistical independence between the data of each one of the three weeks (as they are non-consecutive). Consequently, it resulted on a total of 318 experiments (106 sensors x 3 weeks).

A. Experimental Setup

The parameter setting employed is described in Table II. The ARIMA model (p, d, q values and seasonality) was firstly set (and updated each 24h) by learning/detecting the underlying model running on the historical time series curve of each stand during the first two days of each week. For that, an automatic time series function was employed, i.e. auto-arima. The weights/parameters for each model are specifically fit for each period/prediction using the function *arima* from the built-in R package [stats]. The ETS model (trend and seasonality) are automatically estimated for each and every prediction using the function **ets**, along with the α weight. The automatic forecast procedure followed by this

TABLE I: Descriptive Statistics on the present case study on all sensors and using the incident-based top-10 ones.

Quantity	Flow				Occupancy			
	Mean	SD	Min.	Max.	Mean	SD	Min.	Max.
All day	9.9486	3.6514	0.1811	24.4104	2.1409	1.3276	0.0726	13.9967
P.I	6.9531	2.9221	1.0316	20.0635	1.8205	1.0811	0.1168	8.9752
P. II	11.4231	2.6404	0.634	21.8770	2.2923	1.2306	0.2481	10.3593
P. III	12.5800	2.5469	1.0557	21.6421	2.4477	1.3093	0.4264	10.4852
P. IV	9.5587	2.5047	2.3807	18.9269	2.0853	1.0633	0.4511	9.2519
Top 10, all day	8.2397	3.7001	0.1600	36.7667	3.4303	4.0282	0.0700	23.5567
Top 10, P. I	6.2533	3.4739	1.1367	33.1067	3.0842	2.9594	0.1400	21.6267
Top 10, P. II	9.2869	3.2070	0.5000	30.8133	4.2062	4.5805	0.2000	22.2867
Top 10, P. III	10.686	2.6526	1.0100	22.4333	3.6623	4.3789	0.3800	21.1733
Top 10, P. IV	7.2711	2.3673	0.9400	16.4200	2.6818	2.6731	0.1800	20.5333

function is described in [23]. Then, the resulting model is used by the function *forecast*. The functions *ets*, *auto-arima* and *forecast* are part of the R package [forecast] [29].

B. Evaluation Metrics

The evaluation of these experiments were performed on two distinct dimensions: (1) numerical prediction and (2) event detection. In (1), we used *Root Mean Squared Error* (RMSE) and *Mean Absolute Error (MAE)* as evaluation metrics. On the other hand, we employed *Precision* (PRE) and *Recall* (REC) as the two main evaluators. To know more about these metrics, the reader may consult the Sections 5.7-5.8 in [30]. The results were aggregated using an weight average of these metrics, where each sensor's weight is equivalent to the total number of events that they experienced.

C. Results

The results are presented in three distinct folds: Table III presents the aggregated results for all the considered weeks. Fig. 4 introduces an time-evolving evaluation in terms of RMSE produced by the three flow prediction methods hereby presented. The neuron activation and incident's boolean states are also exhibited on this chart. It is possible to observe that the ensemble prediction error is always lower than the one obtained from other methods. On the other hand, we can also conclude than the incident detection is independent on the neuron activation. Finally, Fig. 5 illustrates the prediction behavior along sensor with an increasing incident rate (on x-axis). The recall values are averaged using a sliding window considering just the recall values for the latest ten sensors with respect of the current one. By doing so, it is possible to conclude that the our method performance increases along with the number of incidents observed in each sensor. The averaged computational time required to predict a series term was 15 seconds.

TABLE II: Parameter Setting used in the experiments.

	Value	Description
H	1	sliding window size to compute our ensemble
δ_f	1.0	max. admissible flow prediction's residual for PH test
δ_o	0.1	max. admissible occupancy prediction's residual for PH test
λ_f	20	cumulative flow-based threshold to trigger an alarm on PH test
λ_o	4	cumulative occupancy-based threshold to trigger an alarm on PH test
β_0	0.3	initial value of the reactivability rate applied by the real-time neuron
χ	6	time window size necessary to turn off the real-time neuron
φ_f	10	flow-based min. threshold to trigger an event
φ_o	5	occupancy-based max. threshold to trigger an event
ϑ	6	time window size to compute the recent correlation between flow/occupancy

TABLE III: Comparative results on comparing *Drift3Flow* (D3F)'s with the two State-of-the-Art methods.

Method	Week	Flow Prediction		Occ. Prediction		Event Detection	
		RMSE	MAE	RMSE	MAE	PREC	REC
ARI	1	1.5471	1.0799	2.3899	1.5393	0.8919	0.3672
ETS	1	1.5476	1.0629	2.5118	1.4899	0.8963	0.2978
D3F	1	**1.5289**	**1.0564**	**1.9553**	**1.2325**	**0.9156**	**0.4742**
ARI	2	1.4653	0.9634	2.0885	1.2417	0.7349	0.3072
ETS	2	1.5311	0.9781	2.5316	1.3561	0.7349	0.3228
D3F	2	**1.5289**	**1.0564**	**1.9553**	**1.2325**	**0.6156**	**0.3282**
ARI	3	2.0330	1.1729	1.8527	1.1025	0.7715	0.1759
ETS	3	2.0889	1.1819	1.9080	1.0782	0.8005	0.2808
D3F	3	**1.9355**	**1.1116**	**1.5889**	**0.9208**	**0.8024**	**0.3128**
ARI	ALL	1.6875	1.0743	2.1088	1.2939	0.8002	0.2823
ETS	ALL	1.7280	1.0765	2.3111	1.3057	0.8116	0.3000
D3F	ALL	**1.6389**	**1.0379**	**1.8151**	**1.0730**	**0.8199**	**0.3719**

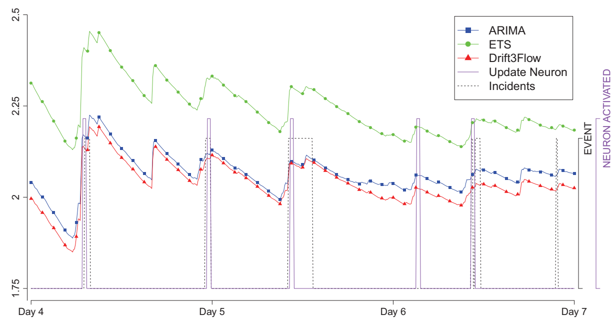

Fig. 4: Time-evolving flow-based evaluation on the top-event sensor (RMSE). Note *Drift3Flow*'s behavior.

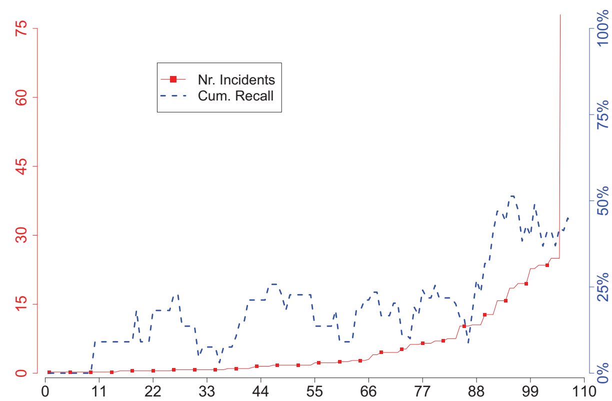

Fig. 5: Average recall for all sensors (on x-axis) ordered by their number of incidents.

D. Discussion

At first glance, the high number of parameters (ten) may appear a major drawback of our methodology. However, just three of them must be adapted for different case studies due to their paper on the learning process: H, β_0 and χ. $\varphi_f, \varphi_o, \vartheta$ address the definition of incident, which is something that must be known before carrying out any supervised learning task. The parameters of the PH test ($\delta_f, \delta_o, \lambda_f, \lambda_o$) follow the same logic, addressing the sensibility of our system to divergences on the residual's *p.d.f.*. It is translated on setting the reactivability of our method - which is highly dependent on the AID's application scenario (e.g. dynamic lane selection or just driver's information through variable message signs).

Yet another issue uncovered by Table III is the low recall values, which indicate a significant percentage of false negative alarms. However, this is a characteristic common to this type of approach taken by these flow prediction methods to the AID problem [2]. Nevertheless, Fig. 5 shows that the recall increases for sensors with larger incident rates. Moreover, by selecting appropriate values for these thresholds, it is possible to find an adequate compromise among the false positive/true negative. Yet, we must highlight that our **method outperformed the current State-of-The-Art** on this particular problem - which is well illustrated by Fig. 4. Its adaptive characteristics (i.e. incrementality) are key to do so. By taking a full non-parametric approach, our method gets some sort of *freedom* to react earlier to changes on the flow/occupancy signal. Such approach may explain its success on road sections with high-incident rate - as illustrated by Fig. 5.

VI. Final Remarks

This paper presented a novel methodology for AID: *Drift3Flow*. This data driven methodology surpasses conceptually the latest solutions to this task by taking no prior assumptions on the residual's *p.d.f.*. Its effective contribution was validated through an experimental setup which considered a real world case study. Its results were 25% better in terms of Recall than the ones obtained by the traditional methods for flow prediction-based AID.

Fig. 3 uncovers the bimodal structure of the flow *p.d.f.* on our samples. It may be explained by the existence of road sections/sensors with distinct behaviors. This pattern may justify to adapt our learning model to such distinct contexts. Hence, it comprises an open research question.

References

[1] T. Nagatani, "The physics of traffic jams," *Reports on progress in physics*, vol. 65, no. 9, p. 1331, 2002.

[2] S. Tang and H. Gao, "Traffic-incident detection-algorithm based on nonparametric regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 38–42, 2005.

[3] B. Baskar, L. De Schutter and H. Hellendoorn, "Model-based predictive traffic control for intelligent vehicles: Dynamic speed limits and dynamic lane allocation," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 174–179.

[4] NHTSA - National Highway Traffic Safety Administration , "Traffic safety facts 2012," U.S. Department of Transportation, Tech. Rep., 2012. [Online]. Available: http://www-nrd.nhtsa.dot.gov/Pubs/812032.pdf

[5] J. McClelland, D. Rumelhart, P. R. Group *et al.*, "Parallel distributed processing," *Explorations in the microstructure of cognition*, vol. 2, 1986.

[6] E. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[7] A. Karim and H. Adeli, "Incident detection algorithm using wavelet energy representation of traffic patterns," *Journal of Transportation Engineering*, vol. 128, no. 3, pp. 232–242, 2002.

[8] P. Ross, *Exponential filtering of traffic data*, T. R. Record, Ed. TRB, 1982, no. 869.

[9] A. R. Cook and D. E. Cleveland, *Detection of freeway capacity-reducing incidents by traffic-stream measurements*, 1974, no. HS-015 791.

[10] B. Williams, P. Durvasula, and D. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1644, no. 1, pp. 132–141, 1998.

[11] B. Williams and L. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[12] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[13] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, no. 2, pp. 253–261, 2001.

[14] K. Chan, T. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[15] J.-B. Sheu, "A sequential detection approach to real-time freeway incident detection and characterization," *European Journal of Operational Research*, vol. 157, no. 2, pp. 471–485, 2004.

[16] G. Box, G. Jenkins, and G. Reinsel, *Time series analysis*. Holden-day San Francisco, 1976.

[17] C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International Journal of Forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[18] P. Everall, "Urban freeway surveillance and control: the state of the art," Washington D.C.: Federal Highway Administration, Tech. Rep., 1972.

[19] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2024, no. 1, pp. 116–125, 2008.

[20] L. Chen and C. Chen, "Ensemble learning approach for freeway short-term traffic flow prediction," in *IEEE International Conference on System of Systems Engineering*. IEEE, 2007, pp. 1–6.

[21] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[22] H. Wang, W. Fan, P. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 226–235.

[23] R. Hyndman, A. Koehler, R. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[24] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "On predicting the taxi-passenger demand: A real-time approach," in *Progress in Artificial Intelligence*, ser. LNCS. Springer, 2013, vol. 8154, pp. 54–65.

[25] ——, "Predicting taxi-passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[26] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[27] G. Stone, "An analysis of the delta rule and the learning of statistical associations," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, pp. 444–459, 1986.

[28] L. Moreira-Matias, J. Gama, J. Mendes-Moreira, and J. Freire de Sousa, "An incremental probabilistic model to predict bus bunching in real-time," in *Advances in Intelligent Data Analysis XIII*, ser. LNCS. Springer International Publishing, 2014, vol. 8819, pp. 227–238.

[29] K. Yeasmin and J. Rob, *Automatic Time Series Forecasting: The forecast Package for R*, 1999. [Online]. Available: http://oai.repec.openlib.org

[30] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.