

Micro-Adaptive Long-Term Traffic Forecasting with Evolving Spiking Neural Networks

Ibai Laña^{a,*}, Jesus L. Lobo^a, Elisa Capecci^d, Javier Del Ser^{a,b,c}, and Nikola Kasabov^d

^aOPTIMA Unit. TECNALIA. P. Tecnológico Bizkaia, Ed. 700, 48160 Derio, Spain

^bDept. of Communications Engineering. University of the Basque Country UPV/EHU. Alameda Urquijo S/N, 48013 Bilbao, Spain

^cBasque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain

^dKnowledge Engineering and Discovery Research Institute (KEDRI). Auckland University of Technology (AUT), 1010 Auckland, New Zealand

Abstract

Road traffic management is a critical aspect for the design and planning of complex urban transport networks for which vehicle flow forecasting is an essential component. Due to the nature of traffic itself, specially in urban contexts, most predictive models reported in literature aim for short-term forecasts, and their performance degrades when the prediction horizon is increased. Long term forecasting strategies are more scarce, and commonly based on the detection and assignment to patterns. These approaches can perform reasonably well unless an unexpected event provokes non predictable changes, or if the allocation to a pattern is inaccurate. This work introduces a method to obtain long-term pattern forecasts and adapt them to real-time circumstances. The method takes advantage of the architecture of evolving Spiking Neural Networks (eSNN) to perform online adaptations without retraining the model. Its performance is assessed over a real scenario with 5 minute data of a 6 month span of traffic in the centre of Madrid (Spain). Significant accuracy gains are obtained when applying the proposed online adaptation mechanism on days with special, non-predictable events that degrade the quality of their long-term traffic forecasts.

Keywords: Traffic forecasting; cluster analysis; spiking neural networks.

1. Introduction

Road traffic forecasting models have been developing for more than 40 years, trying to anticipate traffic variables like flow, occupancy, speed or level of service, in order to use them in Automated Traffic Information Systems (ATIS) and Automated Traffic Management Systems (ATMS). Traffic state prediction allows the first to provide accurate information to users, and the latter to take informed routing decisions. One of the main characteristics of traffic forecasting is its limitations in terms of prediction

*Corresponding author: ibai.lana@tecnalia.com (Ibai Laña). OPTIMA Unit. TECNALIA. P. Tecnológico, Ed. 700, 48160 Derio, Spain. Tel: +34 946 430 850.

horizon. The degradation of forecast accuracy, when the horizon is increased, is an assumed common ground in previous literature [1, 2, 3], which is often attributed to the stochastic nature of traffic itself, but also to external factors like events or weather [1]. Although short-term predictions (operating under 60 minute horizons) constitute the preeminent body of traffic forecasting literature [4], some researchers have highlighted the pertinence of long-term forecasting [5, 6, 7], as they are valuable for ATMS or logistics and planning [8].

Although there are evident obstacles to achieve long-term forecasts, recent data driven approaches, and the availability of big volumes of data and contextual information have pushed forward new advances in long-term forecasting methodologies. These methodologies are mainly based on defining patterns that represent typical traffic profiles in different circumstances, while trying to assign future situations to one of such patterns. The ways in which the patterns are defined and the assignation performed comprise an assortment of methods and techniques. In this line of work, [9] defined the long-term predictions as trends over which Principal Component Analysis is applied to detect abnormal cases. In [10], context information from surrounding measuring stations is used to build similarity patterns, and then short-term forecasting is performed with previously defined ground truth. Statistical models are used in [11] and [12]. Both use b-splines to estimate traffic flow and characterize types of days. A clustering and proxy dataset approach was presented in [13], and will be used as an starting point in this work, introducing some improvements.

Forecasting methods have involved time-series analysis and prediction with a wide diversity of variants and enhancements. Machine learning approaches have been dominant in this field for the last two decades, with abundant kinds of methodologies, algorithms and optimization procedures [3, 4, 14, 15]. More recently, Big Data technologies have allowed new angles to tackle the traffic forecasting problem [16, 17]. Online learning strategies are also a trend, seeking models that can operate with a few observations of current traffic. In [18], authors proposed an online traffic prediction algorithm that predicts with real time readings leaning on ensembles of weak predictors. In the wide range of machine learning techniques, neural networks and their variations are particularly popular [4, 19] for traffic forecasting; however, finding works that make use of the potential of the so-called third generation of neural networks is challenging.

Spiking Neural Networks (SNNs) [20] were developed to obtain more accurate representations of biological neural networks in mammals [21]. This technique and its variants simulate the operation of actual neurons, by communicating among them with sequences of spikes, and representing accurately the operation of synapses [22] as learning rules. Nevertheless, their modeling capacity goes beyond the representation of complex spatio-temporal patterns: they have been proven an efficient tool for a range of engineering problems and other fields [23, 24, 25, 26, 27]. However, it is surprising that no works on traffic forecasting have so far relied on SNNs, despite their specialized capability to represent spatio-temporal data [28], and being this one of the hot topics of traffic modeling [4]. Besides this relevant feature of the general SNN model, one of its variants, evolving Spiking Neural Networks (eSNN) [29], presents a certainly appealing characteristic for online traffic forecasting: an eSNN model can grow and learn new information without retraining it, by evolving, (i.e. incrementally adding), spiking neurons [22]. This particular trait provides eSNNs with a fast updating structure, of

utmost utility for the adaptive online learning proposed in this work.

The aforementioned trends, regarding the exploitation of large amounts of traffic data for long-term predictions in an online fashion, confront relevant challenges associated to its stochastic nature and evolution during long periods of time. Indeed, the analysis of large streams of data that may cover temporal ranges of years has become the focus of an exhaustive research area, in which *online learning* and *concept drift* detection and adaptation are central topics. Concept drift – namely, a change in the underlying distribution of streaming data [30] – is present in many fields where data streams are generated, such as computer and sensor networks, financial markets, mobile phones, intelligent user interfaces, and of course, traffic [31]. The data generation process may be affected by non-stationary phenomena (i.e. seasonality, periodicity, sensor errors, etc.), causing models trained over these data to become obsolete and consequently unable to adapt to new data distributions. In order to overcome these issues, concept drift detection, characterization and adaptation has gained popularity in recent literature [32, 33], essentially due to the necessity for adaptive prediction techniques that blend together drift detection and adaptation for these changing environments [34].

Aside from drifts provoked by the passing of time and inherent evolution of traffic, works that use big traffic data [35] have capitalized on other issues with more immediate effects that can be also addressed with similar change detection and adaptation tools. Among them, some very frequent in long-term traffic prediction are unplanned incidents or events, the planned ones that were not contemplated in the original model, or even a simple bad pattern assignment due to the classifier fallibility. All of them can render the long-term pattern model useless in different degrees, and detecting them and adapting quickly to new circumstances is crucial for a proper operation of the forecasting tools.

This work finds its motivation in this noted need for adaptation to changes in traffic forecasting. Specifically, we propose a novel methodology that yields long-term forecasts using similarity-based clustering of daily traffic volume data, and monitors them in real time to adapt them in case of a mismatch or contingency. All of this is achieved by means of eSNN techniques and a change detection and adaptation mechanism. The main contributions of this research work can be summarized as follows:

1. The implementation of an improved method to obtain traffic patterns for any location and date that can be used by ATIS and ATMS.
2. The use of eSNNs in the traffic domain and the implementation of a method to encode traffic data into spikes over the time domain that feed an eSNN.
3. The development of an online detection and adaptation mechanism that constantly evaluates and updates pattern-based forecasts, taking into consideration current traffic volume observations.
4. The proposal of a framework to effectively maintain long-term predictions with incremental short-term adaptations which can be self updated with new knowledge embedded in newly arriving data samples.

The rest of the paper is organized as follows: Section 2 describes the input data, the two step forecasting model and delves into the encoding of traffic volume to spikes of

an eSNN, as well as the adaptation process. Section 3 presents and analyses the performance of the proposed methods in different missing data scenarios. Finally, Section 4 draws concluding remarks inferred from the obtained results and prescribes future research lines related to this work.

2. Materials and Methods

This research work introduces a long-term forecasting model with a change detection and adaptation mechanism. The general operation of the model is shown in Figure 1. The method consists of two phases, each relying on a different subset of traffic data. A first subset is used in an offline learning phase, obtaining clusters that define daily traffic patterns and subsequently fed to an eSNN classifier. This classifier is then used in the online detection and adaptation phase, which is fed by the second subset of data. The classifier assigns a pattern to a day, and the change detection and adaptation mechanism allows to reassign the day to another pattern if it was originally incorrect. The source and selection criteria for the input data, as well as the hybrid forecasting method is detailed in the following subsections. Nevertheless, the proposed framework can process any other data source without any loss of generality.

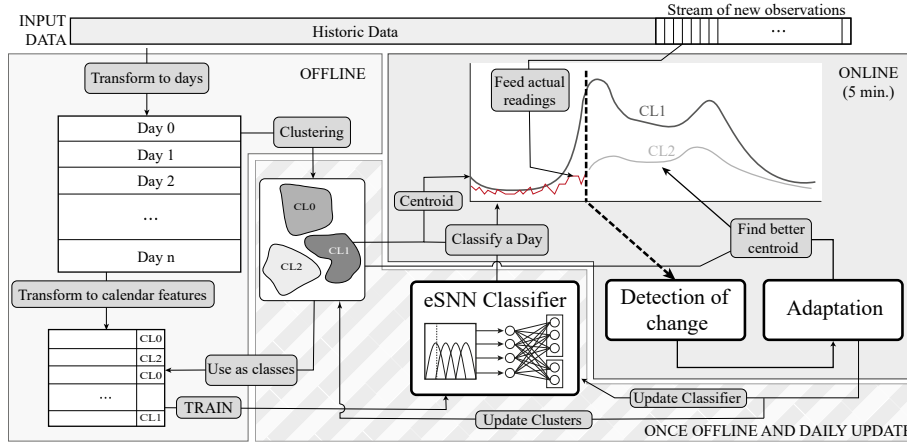


Figure 1: Offline-online model for pattern classification and adaptation

2.1. Input data

Real traffic data are collected from a public source maintained and provided by the City Council of Madrid (Spain). More than 3600 automatic traffic recorders are deployed around the city, mainly in urban contexts, and their readings are published in a live feed in Madrid Open Data portal [36]. A collection tool was implemented to capture readings from a set of sensors in a set of locations in the one of the business areas of the city. Data from November 2016 to June 2017 were gathered and aggregated into $\Delta T = 5$ minutes intervals. The selection of the sensors for the experiments was made attending to a data completeness criterion, as for most locations, reading errors

are produced, resulting in frequent gaps in the data streams [37]. In a real life scenario, a certain amount of available historic data are used to build a model that will be applied later to new data as they are received. In order to mimic this scheme, our stream of data consists of 59328 consecutive observations o_t , which amounts up to 206 days. These data were then divided into two datasets, \mathcal{H} , for historic data that allow to detect and define patterns, and \mathcal{P} for new data that will be assigned to patterns. The subset \mathcal{H} contains 75% of observations o_t , that are rounded in terms of full days, to $|\mathcal{H}| = 154$ days or 44352 observations. The remaining 25% constitute the subset \mathcal{P} , adding up to $|\mathcal{P}| = 52$ days.

The division of the dataset is strictly chronological and not stratified, thus leaving out the first offline phase a model that is trained with less information than we have available. For instance, in our case, with data starting in November and ending in June, the partition is made in the middle of April. This means that the offline model will have learned particularities of the Christmas period, but it will not be trained for special holiday days happening in May. This intends to mirror the behavior of the model in real life, where beyond its best efforts in the offline phase modeling, it could find circumstances producing traffic profiles very different from the assigned pattern in the online phase. With this division, a need for adaptation is forced into the test data \mathcal{P} in order to assess properly the change detection and adaptation mechanism. If a non predictable circumstance has happened before, the model will be able to recognise it and provide a pattern, having potential to deal with other unpredictable events like accidents or abrupt weather changes.

2.2. Offline processing: Clustering Traffic Data and Building a Classifier

In the online phase of our forecasting method, traffic patterns are assigned to days for which there is no known traffic profile. Thus, the offline phase is an initialization procedure that produces:

- A well defined set of patterns that suffice for representing the variety of kinds of days for a given location.
- A classification model capable of predicting the most suitable pattern for a day without having any of the traffic observations for that day, thus relying on features not related to traffic that are available for future days.

In previous research [13, 37], we have successfully performed traffic pattern discovery and classification with different purposes. The initial processing phase of our proposed model builds upon these previous works, introducing further changes that allow for a fast online classification and improvements in the modeling of a dataset with features not based in traffic data. As in these previously developed methods, and in other works [38, 39, 40, 41, 42], the continuous sequence of traffic volume observations is split into day-wise vectors such:

$$\mathbf{o}^d = [o_{t_d}, o_{t_d+1}, \dots, o_{t_d+P-1}], \quad (1)$$

where o_{t_d} is the traffic volume captured at time t_d , being t_d the first observation of day d ; and $P = 288$ is the number of observations obtained within a day for a capture period

of $\Delta T = 5$ min. These vectors conform a dataset \mathcal{H} over which a clustering procedure is performed, in order to find groups of similar days, aiming to label each day with the cluster it belongs to. The high dimensionality of \mathcal{H} , with 288 features per sample, can hinder the clustering performance and its results could be distorted by the noise, which is inherent to individual samples. Therefore, the original $P = 288$ dimensions of the dataset are reduced to $\lceil P/K \rceil$ by averaging every K samples, smoothing each sample without loosing their ability to represent the traffic profile of every day.

2.2.1. Clustering

The clustering process has been performed with DBSCAN algorithm [43]. This clustering method defines areas in the data space by grouping samples delimited by their density in each area. Its suitability to our study is supported by two of its main features: it does not require a predefined number of clusters, a parameter that is not known *a priori*; and grouping noise samples in a separate cluster, otherwise they would be assigned to another existing cluster, producing biases when obtaining the centroid. The selection of the DBSCAN main parameters (i.e. the maximum distance between samples to belong to the same cluster, and the minimum number of instances to be considered a cluster) can lead the algorithm to perform between two extremes: 1) an overly high number of clusters composed by very few data examples, and many noise instances; or 2) a low number of clusters, as a result of a high value of distance, to the point that if the distance parameter is high enough to fit all data in the same cluster, there is only one pattern, and no sample is left outside. Therefore, a balance between noise instances and number of clusters must be met in order to avoid such extremes and extract relevant patterns from the traffic data. To this end, an iterative process was conducted aimed to both reduce the number of noise instances and to maximize the number of clusters. No right amount of clusters was considered, but it was assumed that more clusters represent more types of days and more of their particularities.

For each of the C resulting clusters $\{\mathcal{H}_c\}_{c=1}^C$, a centroid $\mathbf{o}^{\odot c} = [o_{t_d}^{\odot c}, \dots, o_{t_d+P-1}^{\odot c}]$ is obtained by averaging all of the cluster members element-wise:

$$o_p^{\odot c} = \frac{1}{|\mathcal{H}_c|} \sum_d o_{t_d+p} \quad (2)$$

where $o^d \in \mathcal{H}_c$ and $|\cdot|$ denotes cardinality of a set.

2.2.2. Proxy Dataset

After the clustering, the dataset \mathcal{H} is defined with 288 features and a class, i.e., the cluster identifier. However, it is not suitable to train a classifier aimed to assign patterns to future days, because no readings will be available those days to build the samples. Consequently, a proxy dataset is built with the same number instances and classes as \mathcal{H} , but a set of features that are known for the days to come. An analysis of the clusters obtained in \mathcal{H} can lead, along with field expert knowledge, to the extraction of features that can define each cluster. The choice of features is a crucial step of the presented modeling scheme. A feature selection process should be conducted individually for each location, like in any other classification or regression problem. In this work we consider a set of calendar features, which are among the most frequently used for this purpose [35,

44, 45, 46]. Although there are some other variables that are commonly very relevant for traffic, such as weather or traffic incidents, they present the same problem for this scenario as traffic features: they are unknown for future days, so the model would work with estimations. Disregarding them isolates the results from a noise inherent to a bad input prediction.

The selected calendar features are described below, and they will be initially used for all locations, although some of them could be removed if they are irrelevant for a certain location:

- Day of week: Represented by a number between 1 and 7
- Month: Represented by a number between 1 and 12
- Public holiday: a number between 0 and 3 representing respectively: normal day, local holiday, region holiday, national holiday. Different types of public holidays impact the traffic in different ways [47].
- Academic holiday: this feature takes value 1 if the day is within a period of academic holidays and 0 otherwise. During these periods, not only the traffic is affected by a reduction of scholar transport, but it is also frequent that entire families go on holidays, affecting traffic considerably.
- Bridging days: Binary feature that represents when a working Monday precedes a holiday Tuesday or a working Friday is after a holiday Friday. Depending on the location, this days tend to have a different traffic pattern than normal Mondays or Fridays.
- Proximity to holiday: Traffic volume is also affected when a public holiday is close in time, as part of the population may decide to take vacation days to extend the break [47]. This has been encoded in the proxy dataset with this feature that takes values from 0 to 5, representing 0 no proximity, 5 holiday day, and 1 to 4, the inverse of the time distance to the holiday of the 4 surrounding days, before and after the holiday day (e.g., the next and previous days to the holiday would have value 4, and so forth).

The combinations of these features are able to portray most of the situations and types of days that can happen during a year and affect the traffic profile (without considering other unpredictable incidents, special events or meteorology extremes that can cause even more severe changes in traffic). With them, a dataset \mathcal{H} is formed with 6 features is built and used to train the classifier described in the following Section.

2.2.3. Classification with Evolving Spiking Neural Networks

Evolving Spiking Neural Networks are intelligent machines able to apply incremental learning rules to adapt their structure to the data. This feature of eSNNs makes them efficient to handle online classification problems and, since they were first proposed by [29], they have been applied to different types of data [48, 49, 50]. To the best of our knowledge, no applications have been found within the online traffic forecasting domain and only in [28] traffic data are used as a benchmark of their spatio-temporal

representation capabilities. However, an eSNN classifier is well-suited to build a fast online learning model, as neurons are generated incrementally to allow the system to self-grow and learn new information using only one-pass data propagation.

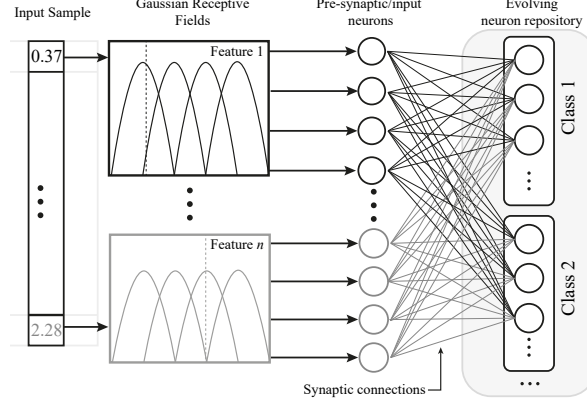


Figure 2: An eSNN architecture with its three main layers: input layer, a pre-synaptic layer and an output layer.

As shown in Fig. 2, an eSNN model is structured into different layers: firstly, every feature in a sample is transformed into a train of spikes using a number G of Gaussian receptive fields, as proposed by [51]. Centers and widths of Gaussian curves are defined for every feature, and each field represents a pre-synaptic neuron. This method uses overlapping Gaussian activation functions to populate continuous inputs over multiple neurons. This is a biologically inspired approach which simulates cortical neural processing of external inputs [52, 53, 54], and has been successfully applied to represent real-valued data [51, 55, 56, 57]. The points in which each curve is cut by the real value define the times in which spikes are produced, as depicted in Fig. 3.

After encoding input data, a repository of trained output neurons is created for each class, and connections to all pre-synaptic neurons are established through the computation of a vector of weights that depends on the order of spike transmission, as defined by [58]. Each neuron i has a firing threshold $\vartheta^{(i)}$ that is obtained through a model parameter c and the maximal potential of the neuron, defined by its weights, the order of spike transmission and a modulation parameter m . In this way, a reservoir of trained output neurons is generated during supervised learning. The total weight value of each trained neuron is then compared with the weight value of each stored neuron and the minimal Euclidean distance calculated. If this is less than a set similarity threshold s , the two neurons are considered “similar” and they are merged by averaging their weights and firing thresholds $\vartheta^{(i)}$; otherwise, if a new neuron is added (evolved) as a new output neuron of the SNN. When training is performed and output neurons are defined, classification is made by propagating a sample through the network; the assigned class is that of the output neuron with the shortest response time. The details of the operation of eSNN model are amply presented in [59]. Thus, adding new samples to the trained model only implies merging them with their corresponding neuron repository.

With the operation scheme presented in Fig. 1, clusters are obtained from traffic

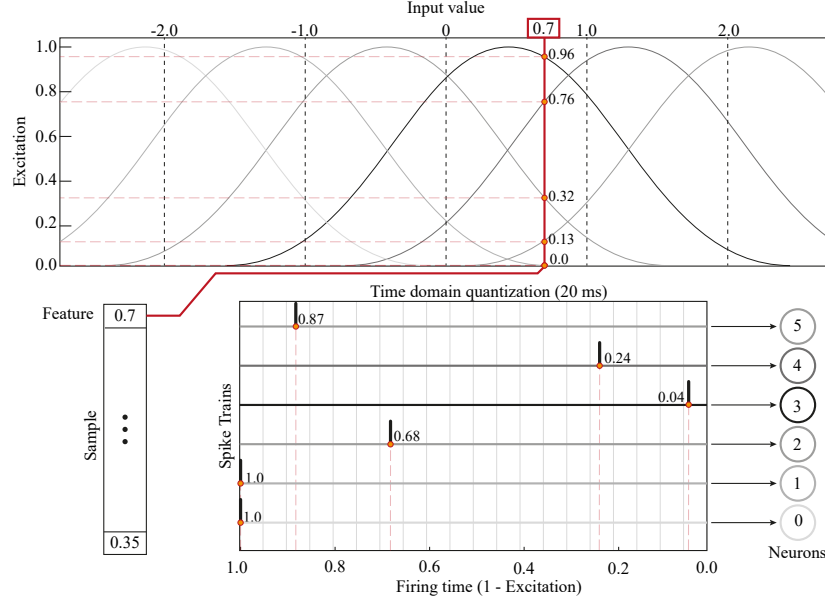


Figure 3: Example of how a random continuous value is encoded into spike trains for a number of input neurons. The value of 0.7 activates six overlapping receptive fields, which excite six corresponding input neurons at different firing times.

data in \mathcal{H} and the proxy dataset $\tilde{\mathcal{H}}$ is built. The eSNN is then initially trained with this proxy dataset based on historical observations. In order to obtain the optimal values for the main eSNN parameters, i.e., the similarity threshold s , parameters c and m , and the number of Gaussian receptive fields G , the classifier is run through a parameter-tuning optimization procedure based on a genetic algorithm to efficiently search over the parametric configuration space for each location under study. The trained network will then be used online for classification, and evolved daily, by virtue of the incremental evolution capability of output neurons [59]. New knowledge can be easily introduced to the network by encoding it and processing it to the neuron repository, with the above described mechanism.

2.3. Online processing: Classification and Adaptation to Change

The main contribution of this work is the online phase depicted in the right part of Figure 1. Predictions obtained in the offline phase, with the discovery of patterns and assignment of new days to them are assumed to be representative of the diversity of traffic profiles in the data of the loop under study to the moment a prediction is queried. After the pattern clustering and classification procedure is executed, most future days will be classified accurately and the assigned traffic profile will match the actual one within a certain tolerance [13]. However, there are circumstances for which this condition is not satisfied. The most immediate obstacle for it to be met is the clustering and classification accuracy. In the first place, it is expectable that the clustering process does not recognise all possible patterns that occur in the data at hand. In addition, even

under the assumption that it is possible to adjust the clustering process to yield a set of clusters enough to describe all the existing kinds of days in the available training dataset, it is likely that the classification model fails to classify some of the days. Other circumstances are less immediate, but are encompassed in these two. For instance, an accident will probably change completely the traffic profile of a certain location, but if the clustering process contemplates a cluster for the pattern of that kind of day including this kind of event, the method will have an option to adapt and change to that pattern. In the same way, if there is a drift in the long-term traffic, and if the clustering and classification models are updated accordingly they will be able to effectively cope with the new traffic profiles.

The theoretical frame proposed in this work deals with the update of the clustering and classification models without considering any incident or event that alters traffic. Detection of change and adaptation are hence oriented to detect and correct misclassification problems and to update the clustering model with information that it has not received before. Including external modifying factors like traffic accidents or weather conditions would extend the model capabilities, but would not entail an essential change in its concept. It would require to increase the number of clusters and to add new features to the proxy dataset with data paired to traffic observations. These data are not currently available, hence the conceptual model is developed just for the calendar features, which pose themselves a challenge.

Dealing with changes in the traffic profile in an online fashion has two facets: the detection and adaptation to change of the ongoing traffic observations stream, and the update of classification and clustering algorithms. Both parts are described in following subsections.

2.3.1. Detection and Adaptation to Change

The first part of the online phase consists of detecting if the received traffic observations are excessively deviating from the predicted baseline. In that case, the model must adapt the predicted traffic profile by finding a better baseline from the available ones, being the baselines the long-term predictions obtained in the offline phase.

Anomaly detection is a wide field of study applicable to a variety of contexts, specially to those where a network of sensors gathers data [60]. The techniques employed to detect an anomaly in a sequence of sensor readings can be easily extrapolated to change detection, thus their common grounds have been applied in our model. The change detection mechanism is based in the definition of a threshold U and a window \mathcal{W} of warnings which has a maximum size of W_{max} warnings. Whenever the difference between the observed actual value ($o_t \in \mathcal{P}$) and the predicted one ($o_{t_d}^{\odot c}$) exceeds U , a warning is raised and stored in \mathcal{W} . When W_{max} consecutive warnings have been raised, the sequence is interpreted as a change in the data. As this happens, the adaptation mechanism is triggered. The process is described in Algorithm 1.

The key aspect of this process leans on the definition of U and W_{max} . Decreasing the threshold or the warning window size would result in more change alerts, and an increase of false positives, *i.e.* days that are correctly classified and the assigned centroid represents adequately the traffic of that day, but slight changes due to noise provoke a change detection and thus and adaptation (search for another traffic profile). On the other extreme, increasing the threshold and window size makes the detector less

Algorithm 1: Change detection mechanism

Input: Prediction $o^{\odot c}$ array for the current day, individual observation $o_t \in \mathcal{P}$,
array of previous warnings \mathcal{W} , maximum size W_{max} , threshold U

Output: Change detected, updated \mathcal{W}

```
1 if  $abs(o_t - o_{t_d}^{\odot c}) > U$  then
2    $\mathcal{W} = \mathcal{W} \cup o_t$  else
3      $\mathcal{W} = \emptyset$ 
4   end
5 end
6 if  $|\mathcal{W}| = W_{max}$  then
7   return TRUE,  $\mathcal{W}$ 
8 else
9   return FALSE,  $\mathcal{W}$ 
10 end
11 end
```

sensitive, and days that should be reassigned could be left out of the adaptation process. This becomes even more convoluted as the prediction baseline $o^{\odot c}$ is smoothed, for it is an average of all of the elements in the cluster, but the readings o_t are noisy, making the discrepancy more likely to occur. Intuitively, a change should be detected and corrected as soon as possible, in order to make the rest of the day predictions accurate, thus sensitivity should be boosted. Nonetheless, some parts of the day are more sensitive *per se*. This is a result of the considerable differences between traffic variability during different parts of a day: for instance, between 2 a.m. and 5 a.m., traffic profiles tend to be barely variable, and a even a low threshold could trigger change alerts. However that same threshold would be useless during rush hour, where great variations happen from day to day. If u or w are too low, many false alarms might be risen every day during the first hours.

For the reasons exposed above, both U and W_{max} should be well defined to maximize the detection of true alerts while minimizing the false positives. Either U or W_{max} should be adaptive to mitigate the effects of the variability of traffic during the day. As their adaptation results in similar outputs (increase of any of them reduces sensitivity and their decrease enhances it), for this research one of them (U) has been fixed, while W_{max} has been tailored for an optimal change detection performance. The value of U is obtained for each 5 minute period of the day of a certain cluster and it is the standard deviation of all the measurements taken at that certain slot p within the days of that cluster, as per (3), resulting in an array U_c of 288 thresholds u_p^c per cluster. Such a standard deviation can be estimated as:

$$u_p^c = \sqrt{\frac{1}{N-1} \sum_{o^d \in \mathcal{H}_c} \left[o_p^c - \left(\frac{1}{N} \sum_{o^d \in \mathcal{H}_c} o_p^d \right) \right]^2}, \quad (3)$$

where o_p^c are all observations of a certain 5 minute slot p of cluster c

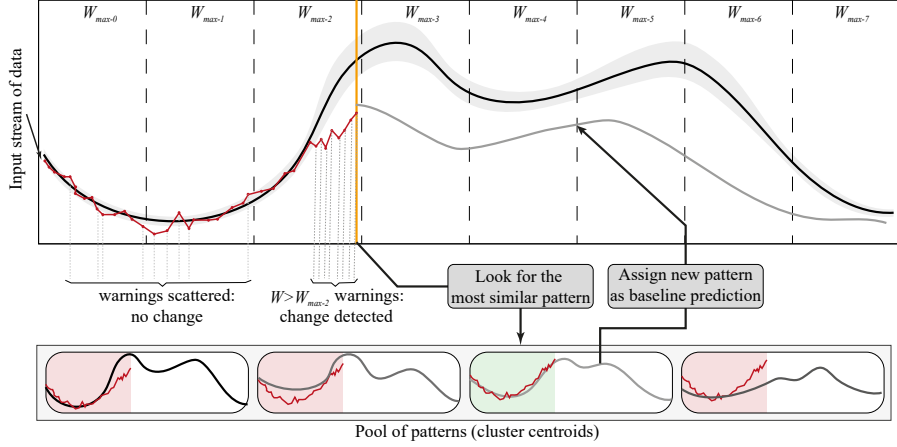


Figure 4: Mechanism for change detection and adaptation. When a set of warnings exceeds the maximum size of the warning window it raises a detection alert and finds the closest pattern

This definition allows for greater variations during peak hours, when the volume of vehicles can be very different even for the same type of days, and lower tolerances in periods with more flat traffic profiles. The adaptive window size W_{max} permits to compensate (if needed) this sensitivity variation. This adaptive nature is furnished by means of 8 values of W_{max} , one for each 3-hours segment of the day. Thus, the number of warnings needed to raise an alert is different throughout the day, and can be forced to be less sensitive during some periods. Optimal sizes for each period are obtained after a grid search procedure is performed over a set of validation days from \mathcal{H} that aims at minimizing false positives and maximizing true alerts.

Adaptation is performed whenever Algorithm 1 returns a TRUE value. This process consists of seeking a better candidate for the current traffic profile. To this end, the sequence of available observations for the current day is compared to all the cluster centroids $\{o_p^{\odot c}\}_{c=1}^c$ in terms of Euclidean distance. The closest cluster is assigned as the prediction for the rest of the day. If the new cluster is found to be the same as the previous one, the day can be a member of the cluster, but it is possible that is closer to other members than to the centroid itself. Thus, the distance to each of the members is computed and a distance-weighted average is assigned as traffic profile for that particular day as a baseline prediction. This intracluster averaging procedure smooths slightly the assigned profile, as assigning the values of a particular member (the closest) would transfer all the noise embedded in that member. However, this smoothing procedure is not enough, as more weighted days introduce more noise; for this reason, an additional moving average smoothing is performed in these cases. This process is repeated for each observation of the day in course, so several changes can occur during the day, and if a new assignation is erroneous it can be corrected later, although it is penalized in the prediction performance. The whole detection and adaptation process is graphically summarized in Figure 4.

2.3.2. *Clustering and eSNN Update*

A key feature of the online detection and adaptation mechanism is the update of clustering and prediction models when new knowledge is found in incoming days. While the detection and adaptation can be processed completely online as new observations are received, the update of the classifier model requires that a whole day is processed, for its samples are daily-based. Accordingly, between 23:55 of one day and 0:00 of the next day, the eSNN classifier is updated with new knowledge acquired during the last day. As defined in Section 2.2.3 the structure of eSNN allows for a quick update that can operate immediately after the last observation is received, and the classification for the new day can be obtained before the first observation of the next day. The following scenarios are considered:

1. No change detected, implying that the consecutive warnings condition has not been met for that particular day. Hypothetically, this could lead to interpret that the day has been correctly classified, although it might have risen abundant non-consecutive warnings, suggesting a day with a noisy traffic profile. Anyhow, this result suggests that a fair amount of observations distributed along all the day have fallen within the area defined by the pattern thresholds. The classifier is updated by adding a new instance with the original class and evolving the eSNN model incrementally.
2. Change detected, implying that one or more sequences of consecutive warnings has risen at least one alert. The alert is followed by an adaptation which requires a change of assigned pattern, which ultimately can result in a change of cluster belonging. Besides, more than one alert and cluster change can happen for each single day. This results in a final predicted set of values that can have been obtained from different clusters. As the time passes for each change detection, more observations are available in order to compare them to a sub-sequence of other patterns. This means that the later the detection is produced, the similarity found with other pattern is more significant, for it is based in more values. For the online detection and adaptation phase this situation is inevitable, and the early adaptations will work hypothetically worse than the later ones. Nonetheless, the classifier adaptation phase is performed in the end of the day, so all the real observations can be used to estimate the most proper cluster. When a change is detected, the whole set of actual observations is compared with the available cluster centroids using the same Euclidean distance that is used during the clustering phase. The label of the closest cluster is used as class for this day, and the eSNN classifier is updated accordingly.

An update mechanism is also devised for the clustering process. Once data of a complete day are available, the classification, detection and adaptation system can define a proper class for that day by using the criterion explained before. This class also identifies the cluster to which the day belongs. The clustering updating mechanism consists of aggregating the 288 actual observations to this selected cluster for which their membership is the most adequate. The aggregation encompasses adding the day as a new instance to the pool of selected cluster members, and recomputing the cluster centroid considering also the newly added member. Thus, an update of the cluster members and centroids is performed each day, at the same time that the eSNN is evolved. This allows not only the clusters and classifier to be updated, but also to grow and eventually promote single member clusters that derive from the noise cluster.

3. Results and Discussion

Methods described in previous Section have been tested with traffic profiles registered by sensors deployed in 6 different locations over the Madrid city network, all placed in urban areas with different traffic profiles. Four of them (A, B, C, D) are located in main roads while the other two (E, F) are installed in side residential streets with lower amounts of traffic. Particularly, location E is a street with no points of interest and almost no buildings, holding exceptionally low levels of traffic, with long periods of 0 vehicles passing by. For the sake of space, the location-based plots contained in this Section represent data of only of the sensor in location A , although the deeper analysis is presented and discussed for all of them. This sections presents in the first place the outcomes of the clustering process and the performance of the predictive model without any adaptation mechanism. In the second place, the results after applying our proposed adaptation method are analyzed.

3.1. Offline Prediction Analysis

The initial clustering stage is crucial for a proper operation of the whole proposed method. It provides the classes for the proxy model that will be used in the classification stage, but it also favors a deeper understanding of the traffic behavior in each site, and to examine visually the types of days. Figure 5 shows the clustering results for the traffic of loop A . Analogously, the optimization process that leads to these 9 clusters for location A has yielded different groups for the other placements which are shown in Table 1. The number of clusters is similar for most of them, although differences in the number of days in the noise cluster are more noticeable, and symptomatic of the large differences in variability among the loops. Reducing the amount of noise implies making the distance metric more flexible to include those days. Thus, more days are clustered together and the number of clusters is reduced, being able to represent less particular circumstances and entailing greater errors in subsequent phases.

Table 1: Number of clusters and noise elements after performing DBSCAN clustering with optimized parameters at each location

Location	A	B	C	D	E	F
Number of clusters	9	8	8	7	8	7
Days in Noise Cluster	6	5	4	1	16	7

In the particular case of location A , most days in \mathcal{H} belong to cluster 0, which represents regular working days without Fridays. These have a different behavior in this particular location, and most of them are assembled in their own cluster. The same happens with Saturdays and Sundays. Beyond these 4 *basic* clusters, other special cases can be found with a very distinct traffic profile such as the last 4 days, that belong to Easter, and the Christmas week. Days highlighted in black correspond to noise, i.e., days that do not fit in any of the existing clusters but are not able to conform a cluster by themselves, according to the epsilon (density metric) and minimum elements per cluster constraints. Days like New Year, with a highly active night and low traffic during day, and Christmas are in this cluster, but also the day of Epiphany, an important festivity in

Spain, and the day previous to Easter. The same analysis in location *E* reveals that most of 16 noise days belong to Christmas and Easter periods. This anticipates bad results in following stages, as only a few holidays or special days are characterized in the clusters, and the test set contains at least 10 of these days.

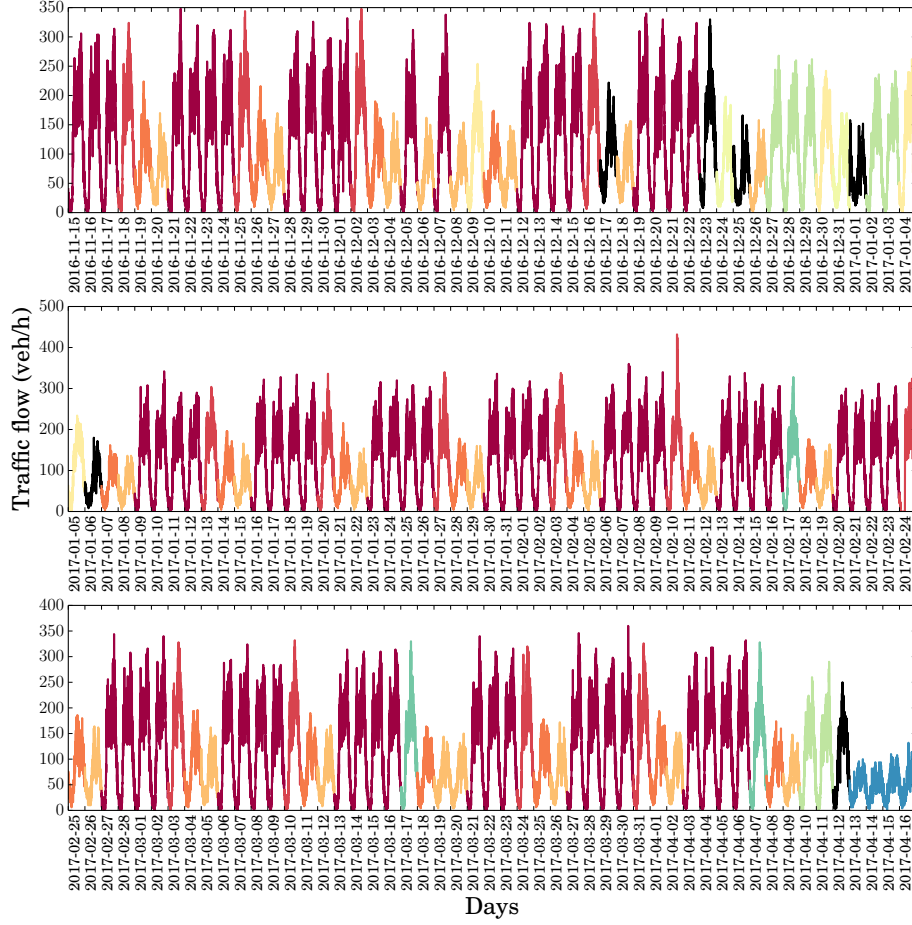


Figure 5: $|\mathcal{H}| = 153$ days of traffic in location A after clustering. Days are colored by their cluster belonging. For visualizing purposes, total observations are divided into three consecutive blocks of 51 days, shown in a row-wise manner

While a noise cluster exists (with all the noise days in it), a noise class can be assigned to some of the samples in the training set, and consequently, when classifying the test set, an instance could be classified as noise. Grouping all the noisy instances and giving them a class entity can bring classification problems, but more importantly, adaptation problems. The variability inside the noise cluster would be too high, and thus, the low amount of alarms based on the cluster standard deviations would lead those days to stay in the noise cluster. Hence, days reported as noise are regarded as clusters with one element. This allows them to be represented in the classification model and con-

sidered for the change and adaptation mechanism. Each noise day is expressed as one sample with different class in the training dataset, so their impact in the classification is expected to be reduced; however, when a similar cluster is sought in the adaptation phase, they compete as new cluster candidates in the same way than the other clusters. Moreover, as the clusters are updated every day, new daily patterns can be included in these single-member clusters, allowing them to grow and be more representative also in the classification phase.

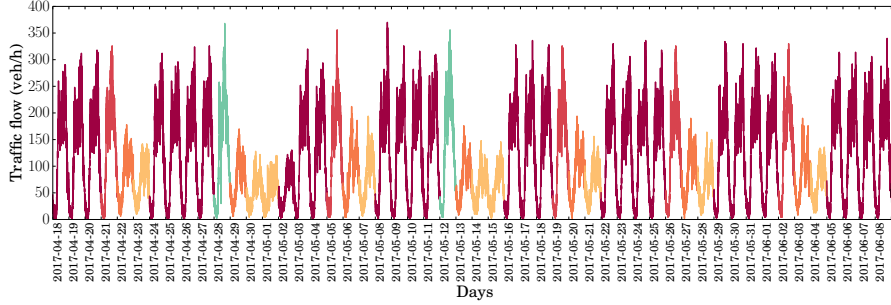


Figure 6: $|\mathcal{P}| = 52$ test days of traffic in location A after classification process.

In order to have a reference for comparison after change and adaptation mechanism is tested, an initial offline classification has been performed for all the 52 test days. The coefficient of determination R^2 , which shows the likelihood of real values to fall within the predicted ones, and the Normalized Root Mean Squared Error ($NRMSE$) are presented for each day, according to

$$NRMSE \doteq \frac{\sqrt{\frac{1}{N} \sum_{\forall t_d} (o_{t_d} - \hat{o}_t)^2}}{\bar{o}_{t_d}}, \quad (4)$$

where N denotes the number of actual values of each day, \bar{o}_{t_d} stands for the average of real observations for day d , and \hat{o}_t is the predicted value for o_t ; and

$$R^2 \doteq 1.0 - \frac{\sum_{\forall t_d} (o_{t_d} - \hat{o}_t)}{\sum_{\forall t_d} (o_{t_d} - \bar{o}_t)}. \quad (5)$$

RMSE is normalized with respect to the average of vehicles passing each day, otherwise each RMSE measurement would have different meanings, depending on the day. Figure 6 exhibits the results for location A, and in Figure 7, the R^2 values for the rest of locations. These results are obtained training the model with half of the year, while the test days are part of the other half; some of the days in the test set can correspond to situations that have never been observed by the model, and thus have traffic profiles unknown to the model. Even considering this, forecasts obtained in A obtain a R^2 value superior to 0.9 for 33 of the 52 days, and as it can be observed in each individual graph, for these cases the actual observations are fit to the prediction lines (their assigned cluster centroids). This means that a basic approach that models patterns without adapting

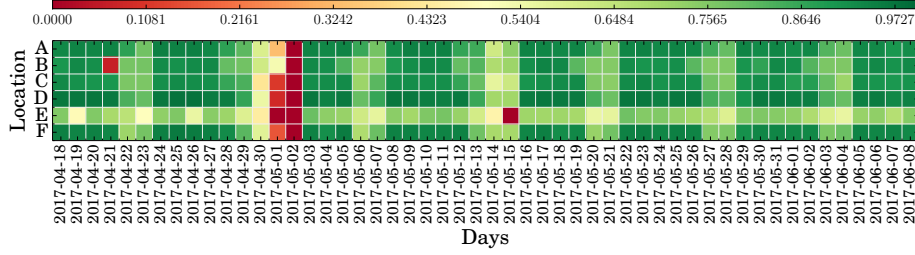


Figure 7: R^2 values for each test day of each location. Any value lower than 0 has been assigned value 0 in order to provide a representative color map.

is useful for more than half of the days, and the error produced in each day will be imputable to slight unpredictable variations.

As opposed to this initial good classification of what could be deemed *normal* days, there are some other days for which the prediction fails, in some cases they could have been assigned to a better cluster. This is the case, for instance, of May 2nd, while others like June 3rd and 4th are apparently in the correct cluster there are greater variations that make the predictions less useful. In the first case, a local holiday that has no similar traffic profile in the training set, and in the second, the celebrations of the winning of a sports championship affected a large area in the city center. By observing the rest of locations in Figure 7 an analogous conclusion holds: the classification model is completely unable to properly assign patterns to the long weekend in the beginning of May, and the performance is also poor for other days in the different locations. For instance, in location *B*, Fridays are consistently predicted worse than in other locations. In general, predictions for weekends are less effective; although they are correctly classified, the information available for the cluster of Sundays is less accurate, there are less previous Sundays than weekdays to learn from, and it is likely that the activities that take place on weekends are very different in the test months (May, June), than in the train months (mainly winter). A model trained with longer periods could have yielded different clusters attending to the season or other circumstances. It is also observable that for location *E*, performances are lower, mainly due to the low dynamic range of its registered traffic data: for most days, volume readings oscillate between 0 and 25 vehicles with long series of real values equal to 0. Also, the mean traffic volume in this location is of about 11 vehicles and its standard deviation is close to 10. This produces highly noisy instances, reducing the possibilities of grouping days in meaningful clusters, which are more similar among themselves. Besides, in this scale, light changes imply broader errors, and a minor traffic increase of 5 more vehicles at some point, can represent a 20 or 25% relative increase of the expected measurement.

Anyhow, any of the previously described shortcomings are useful to assess the performance of the adaptation mechanism presented in this work, whose results are discussed next.

3.2. Online Processing Results

Once an offline clustering and classification iteration is completed as a comparison reference, the proposed change detection and adaptation mechanism, along with the

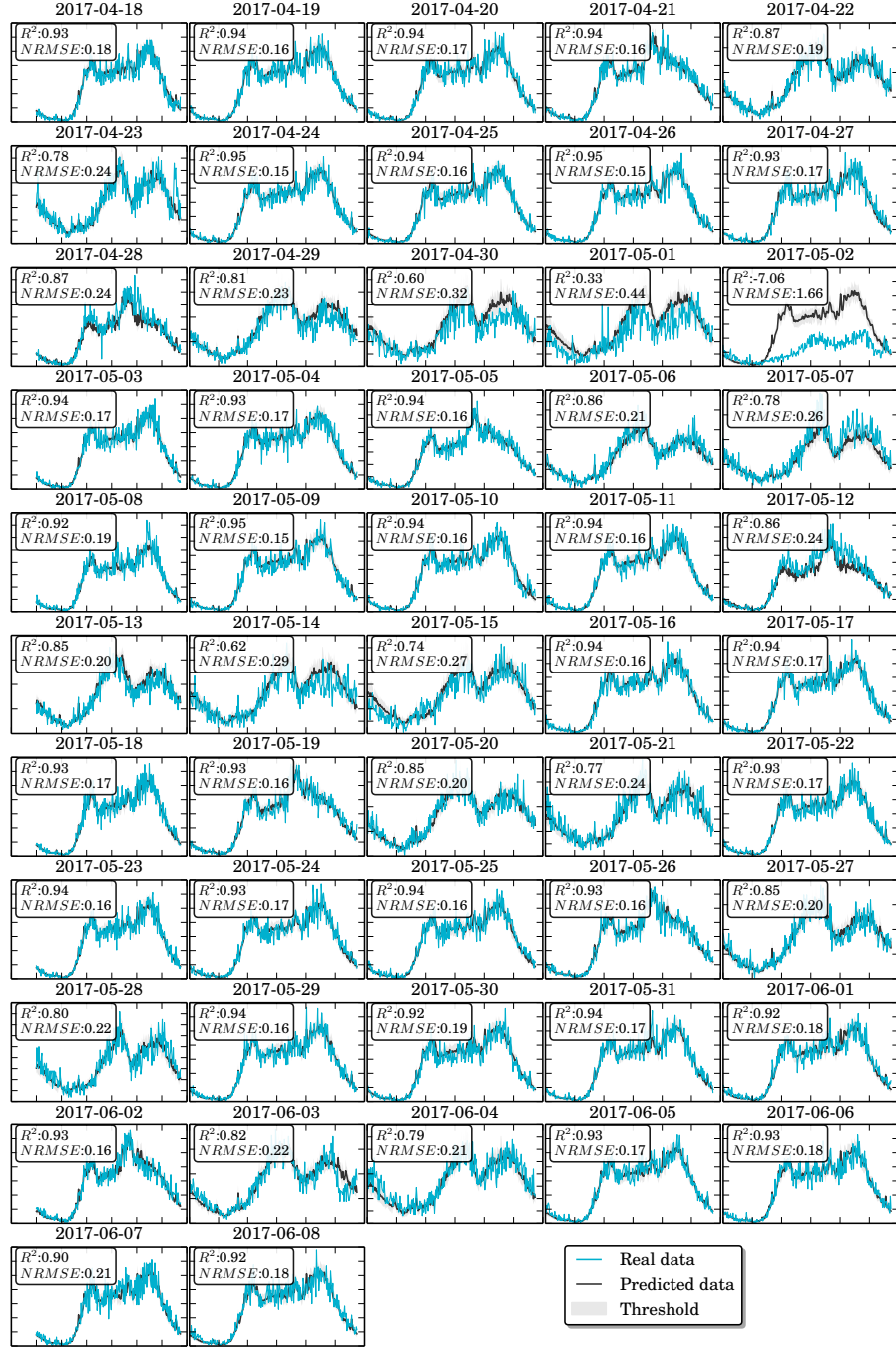


Figure 8: Test days, with real values, data predicted by eSNN model, and R^2 and NRMSE values

clustering update system are activated to assess their performance. An averaged result per location is provided in Table 2 for both considered metrics.

Table 2: R^2 and NRMSE measurements obtained for each location, averaged for the 52 days in the sample, and Wilcoxon p-values for each pair of sample sets.

Location		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
R^2	No adaptation	0.722	0.782	0.756	0.781	0.582	0.729
	Adaptation	0.878	0.840	0.849	0.901	0.654	0.845
	Wilcoxon p-value	0.001	0.009	0.001	8.8e-4	0.005	6.5e-6
<i>NRMSE</i>	No adaptation	0.229	0.277	0.255	0.216	0.476	0.250
	Adaptation	0.195	0.257	0.233	0.194	0.454	0.240
	Wilcoxon p-value	0.001	0.009	0.001	7.2e-4	0.005	3.8e-6

Reduction of error is visible from any of the error perspectives: in general, in every location, the introduction of a change detection and adaptation mechanism has improved the non-adaptive pattern prediction counterpart. The statistical significance of these results has been tested via a Wilcoxon test, comparing the set of results obtained for each day without change detection and adaptation, and those with this mechanism active. The null hypothesis on this test states that there is no statistically meaningful difference between the two measurements, Therefore, these results evince strong evidence against this hypothesis, and the improvements obtained by our proposed method can be declared as statistically significant. However, these averages conceal some relevant aspects of the overall improved outcomes; it has been shown in Figure 7 that a great fraction of test days is properly classified and predicted, with some days for which a bad classification produces a large prediction error. Hence, a better classification for these particular days might make a remarkable difference in averaged results. In order to examine this postulated hypothesis, we inspect particularized results in terms of coefficient of determination for each day and location in Figure 9.

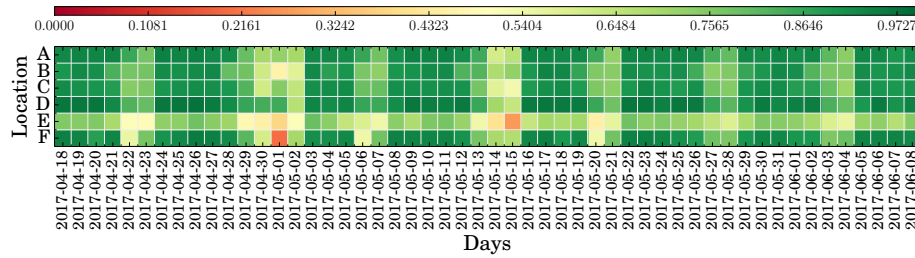


Figure 9: R^2 values for each test day of each location after adaptation. Any value lower than 0 has been assigned value 0 in order to provide a representative color map.

It can be observed in this Figure that although there are still days with low quality forecasts, and most predicted days remain similar, a general improvement has been achieved. Days for which pattern predictions were ineffective in the non-adaptive setting get better approximations after adaptations are made. For instance, May 1st and 2nd are clearly improved for all locations, and very poorly predicted days like April 21st

in location *B* or May 15th in location *E* have been also adapted for the better. On the other hand, there are days with slightly worse predictions, due to diverse factors, that can be noted for location *A* in Figure 10, where a daily detail of the previous results is shown.

The change detection points are signalized with vertical lines, while black line represents originally assigned pattern, and the red line shows the final set of predictions that have been assigned to a day, after the initial assignment and subsequent changes. It is relevant to note that originally assigned patterns can be different that the ones presented in Figure 8, as an adaptation of the classifier and the cluster centroids is performed every day in this online version. This is noticeable for example in April 29th, which was assigned offline to cluster 2, the cluster corresponding to Saturdays, and in the online version it is initially assigned to cluster 8, although then, early in the day is reassigned to cluster 2. Updates in the classifier have uncha this bad classification and the adaptation mechanism has corrected it, resulting in a day predicted with a little less accuracy that with the online version (NRMSE of 0.24 versus the previous 0.23); this situation is also found in May 19th and 23rd, with no prediction accuracy loss, as a result of a good adaptation. A similar case is produced when the same pattern is originally assigned in both offline and online methods, but the online one looks for a better candidate within the same cluster for days like May 29th or June 3rd and 4th. In all of them, the prediction performance is slightly reduced and keeping the original pattern would result in a better forecast. On its counterpart, the same scenario in May 12th results in a considerable improvement; this is more obvious for days like May 1st and 2nd, for which the original prediction was utterly useless, and the updated classifier and adaptation mechanism have jointly provided a more accurate forecast.

Besides the particular results for location *A*, Table 3 displays the number of days for which the prediction has been significantly improved or degraded (more than 5% gain or loss with respect to the original error) for each location, as well as the average accuracy gain or loss that those have entailed. In general, for all locations more than 60% of days have remained without significant change. For some locations there are more days with worse predictions that those daily predictions with good ones, for the reasons described above.

Table 3: Number of days and amount of NRMSE that have been improved and deteriorated after applying the change detection and adaptation mechanism and their average gain and loss amounts.

Location		A	B	C	D	E	F
Improvement > 5%	Days	8	10	7	8	10	5
	Avg. gain	0.207	0.113	0.202	0.212	0.168	0.271
Degradation > 5%	Days	4	5	9	11	8	14
	Avg. loss	0.024	0.032	0.024	0.043	0.054	0.053
No significant change	Days	40	37	36	33	34	33

This greater amount of days with worse forecasting results, however, does not involve a greater error, as for most of these days, the error gap is negligible. For instance, in location *F*, 14 days have obtained worse forecasting results, but they only account for an NRMSE increase of 0.053 points, while the improvements introduced in just 5

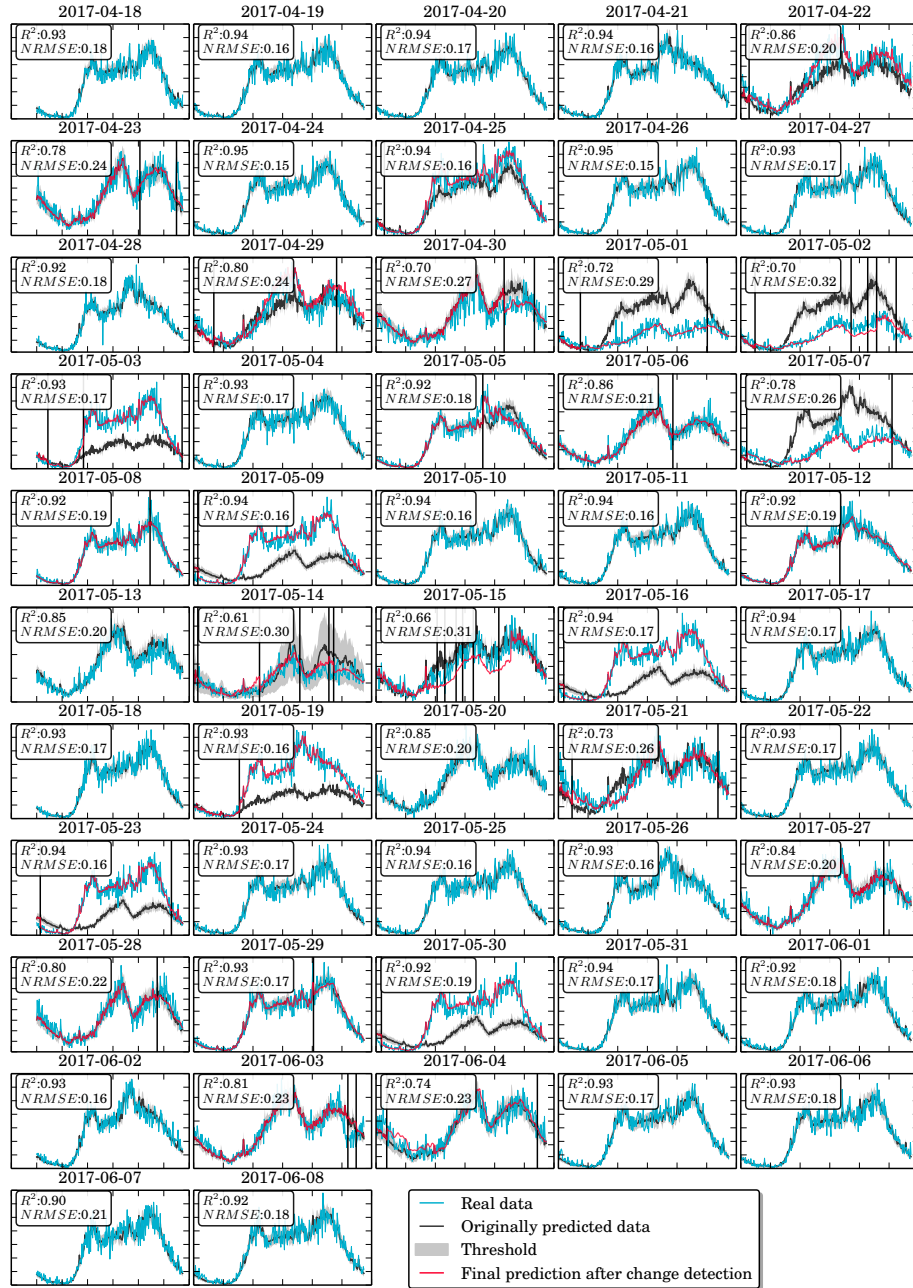


Figure 10: Test days, with real values, data predicted by eSNN classifier and new predictions after detecting changes and online updating classifier and clusters for location A. Vertical lines represent pattern change detection points.

days have reduced the error on an average of 0.271 points. These averages are detailed in Figure 11, where the accuracy gain (error reduction) and loss (error increase) for each location are shown as violin plots. The error gaps for most days are very close to 0, whereas few days for each location are accountable for most the accuracy gain presented in Table 2.

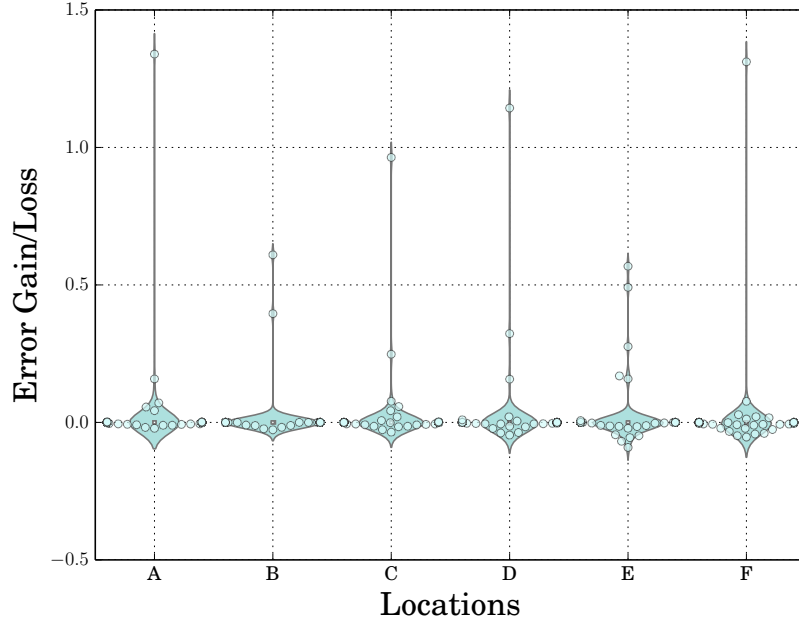


Figure 11: NRMSE error differences distribution for each location. Values above 0 represent days with less error, (and quantify their relative gain), while negative values represent days with more error and correspondingly their relative loss.

All of the presented results and assessments of the model heretofore point at the same direction: an online adaptation mechanism leads to a model that can obtain better predictions even for days that have not been observed previously, but at the cost of a negligible penalty on prediction accuracy for some other days that would have been predicted better otherwise. With only 4 months of data to train, predictions are obtained for another 2 months, never observed by the model. Most test days (more than 70% for all locations and more than 80% for A and D) are predicted with great accuracy ($R^2 > 0.8$) in the long-term, without any adaptation. Location E is excluded from this analysis, since its noisy data have lead to poor results. However, it should be remarked that the characteristics of this location, (with a very low-valued traffic profile, no points of interest, almost no residences, and no connections with major arteries), make this case less conclusive for the study of the traffic of an urban area.

Most difficult days to predict in the used data are May 2nd and 15th, as they are local holidays exclusive to Madrid city and region, and they are next to weekends and other holidays. Such circumstances have never been produced in the train dataset, neither

have been observed by the model. The clustering classification model fails to predict accurate patterns for these days, and they represent an example of how the outcomes are when unexpected conditions are playing. The adaptation mechanism enhances the accuracy of predictions and they can be used as a more approximate estimation of the real traffic. In all cases, even in noisy-profiled location E , baseline predictions have benefited from the adaptation in global terms.

4. Conclusions and Future Work

In this paper we have presented a long-term urban traffic volume forecasting method with an adaptation mechanism and tested it with real data in different locations of the city of Madrid. A pattern clustering and classification scheme has been proven to comprise a useful tool to obtain long-term predictions. The initial design of this scheme is crucial, and when all its parameters are optimized and the features used are carefully selected, it is able to provide accurate predictions months away for the latest day that the training has observed, as a favorable by-product of the traffic seasonality. However, unforeseen circumstances can happen and evolve to a very different traffic profile than the expected one, ultimately misleading the model. For these hindmost cases, the adaptation mechanism provides the means to obtain acceptable levels of forecasting accuracy. The presented case study is intended to serve as a reference of the performance gains the long-term prediction and adaptation engine can achieve. In the absence of other sources of traffic related data, the model has been built and assessed under the premise that pattern classification issues (to which the model adapts) would arise from calendar related conditions. Nonetheless, the model could and should be further enhanced upon the availability of more and more diverse data, such as events, planned road works, traffic incidents, or weather.

In a real application case, it is likely that traffic data would be available throughout a whole year or more, and could enrich significantly the training data substrate for the predictive model. Almost all calendar situations could have been observed and trained previously, and hypothetically, better outcomes could be obtained in the offline phase. However, there are other considerations that could be added to the model to form more fine-grained clusters and classification, such as recurring events. In our case study there are two days clearly affected by a sports event celebration. The offline model provided a pattern that was mostly correct except for the night between the two days spanned by the event at hand. The proposed adaptation mechanism detected the anomaly, but was not able to provide another closer pattern. If in the clustering modeling phase this kind of event was contemplated, a cluster with these kinds of days would be available for the any of the forecasting phases, and the prediction could have been more accurate. Introducing events is an obvious next step in the development of this kind of urban traffic forecasting system. Sports events, demonstrations or parades can have a high impact even on spatially distant traffic, and they are foreseeable, so they can be used for modeling without relying on predictions. On the other hand, other factors like weather are proven to affect traffic notably, but if a model is built on past weather conditions, future weather conditions would be necessary to obtain forecasts with it; the quality of weather forecasts would have then a notorious impact on the model predictive perfor-

mance. Nevertheless, with the proposed adaptive scheme, weather forecasts could be also valuable inputs, as they could be corrected during the adaptation phase.

Beyond including other features, the long-term operation of a system like the one described in this paper would require other type of adaptations. The use of an eSNN allows the model to be constantly updated without retraining, and in combination with the clusters update, permits the model to operate indefinitely. However, in pursuance of a system that keeps learning, and adapts to long-term drift in data, the update of the whole clustering process is desirable. Unlike with the eSNN, introducing new knowledge to the clustering algorithm would imply running the whole process so as to forget the old data distribution, which could dramatically alter the cluster space, centroids and proxy dataset. The cluster space is stable to a certain extent due to traffic data seasonality, so the clustering should not be recomputed afresh on a daily basis, but rather once a year, once all types of days have occurred and they start to happen again. Thus, year after year, slight variations in traffic behavior would be captured by the representation of clusters, and the way in which the proxy dataset is built. If variations in traffic are a result of the alteration of the road profile (for instance, due to road works), the whole process described in Section 2.2.1 should be also performed again considering only observations taken after the alteration. An automated high level adaptation mechanism that inspects the evolution of traffic whole days compared to known patterns could be implemented to detect these kind of variations.

Lastly, we consider that at the possibility of a random shift, the robustness of a model does not pass through the prediction of the random event, but through the ability to adapt to it. Lacking the proper data, we have tested our model behavior against the unexpected with unknown traffic profiles, which are what in essence would produce an unforeseen circumstance. Traffic incidents are stochastic events that produce severe traffic alterations, specially when they occur during certain time frames. Although they are rare in urban contexts, they tend to be recurrent in certain locations. The proposed forecasting system should be fit to deal with this kind of events; if it is possible to find traffic profiles of days with an incident and cluster them apart, the change and adaptation mechanism should be able to reassign the traffic profile to another one in which an incident has happened. Future developments of the presented model could potentially involve all the major aspects that affect traffic in order to have better and more automated forecasting systems.

Acknowledgments

This work has been supported by the Basque Government through the EMAITEK program, as well as by the Erasmus Mundus Action 2 PANTHER, “Pacific Atlantic Network for Technical Higher Education and Research”¹ (Agreement No. 2013-5659/004-001 EMA2).

Bibliography

- [1] B. Van Arem, H. R. Kirby, M. J. M. Van Der Vlist, and J. C. Whittaker, “Recent advances and applications in the field of short-term traffic forecasting,” *International Journal of Forecasting*, vol. 13, no. 1, pp. 1–12, 1997.

- [2] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport Reviews*, vol. 24, no. 5, pp. 533–557, 2004.
- [3] C. Van Hinsbergen, J. Van Lint, and F. Sanders, "Short term traffic prediction models," in *Proceedings of the 14th World Congress on Intelligent Transport Systems (ITS)*, 2007.
- [4] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.
- [5] P. Næss and A. Strand, "Traffic forecasting at 'strategic', 'tactical' and 'operational' level," *Journal of Critical Realism*, vol. 51, no. 2, pp. 41–48, 2015.
- [6] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian, "Long-term forecasting oriented to urban expressway traffic situation," *Advances in Mechanical Engineering*, vol. 8, no. 1, pp. 1–16, 2016.
- [7] C. Kai, K. Zhang, and Y. Hamamatsu, "Traffic Information Real-Time Monitoring Based on A Short-Long Term Algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Taipei, 2006, pp. 651–656.
- [8] C. Lamboley, J. C. Santucci, and M. Danech-Pajouh, "24 or 48 Hour Advance Traffic Forecast in Urban and Periurban Environment: The example of Paris," in *4th World Congress on Intelligent Transport Systems, Mobility for Everyone*, Berlin, 1997.
- [9] L. Li, X. Su, and Y. Zhang, "Trend modeling for traffic time series analysis : An integrated study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 1–10, 2015.
- [10] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under Big Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1786–1796, 2016.
- [11] F. Crawford, D. Watling, and R. Connors, "A statistical method for estimating predictable differences between daily traffic flow profiles," *Transportation Research Part B: Methodological*, vol. 95, pp. 196–213, 2017.
- [12] I. M. Wagner-Muns, I. G. Guardiola, V. Samaranayake, and W. I. Kayani, "A functional data analysis approach to traffic volume forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2017.
- [13] I. Laña, J. Del Ser, and I. Olabarrieta, "Understanding Daily Mobility Patterns in Urban Road Networks using Traffic Flow Analytics," in *Proceedings of the Network Operations and Management IEEE Symposium (NOMS)*, Istanbul, 2016.

- [14] M. A. Abdel-Aty, R. Kitamura, and P. P. Jovanis, "Using stated preference data for studying the effect of advanced traffic information on drivers' route choice," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 1, pp. 39–50, 1997.
- [15] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Spatio-temporal short-term urban traffic volume forecasting using genetically optimized modular networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 5, pp. 317–325, 2007.
- [16] F. Schimbinschi, X. V. Nguyen, J. Bailey, C. Leckie, H. Vu, and R. Kotagiri, "Traffic forecasting in complex urban networks: Leveraging big data and machine learning," in *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2015, pp. 1019–1024.
- [17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [18] J. Xu, D. Deng, U. Demiryurek, C. Shahabi, and M. Van Der Schaar, "Mining the situation: spatio-temporal traffic prediction with Big Data," *IEEE Journal on Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 702–715, 2015.
- [19] I. Laña, J. Del Ser, M. Vélez, and I. Oregi, "Joint feature selection and parameter tuning for short-term traffic flow forecasting based on heuristically optimized multi-layer neural networks," in *International Conference on Harmony Search Algorithm*. Springer, 2017, pp. 91–100.
- [20] A. M. Andrew, "Spiking neuron models: Single neurons, populations, plasticity," *Kybernetes*, vol. 32, no. 7/8, 2003.
- [21] F. Ponulak and A. Kasinski, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta neurobiologiae experimentalis*, vol. 71, no. 4, pp. 409–433, 2011.
- [22] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Adaptive learning procedure for a network of spiking neurons and visual pattern recognition," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2006, pp. 1133–1142.
- [23] S. Kulkarni, S. P. Simon, and K. Sundareswaran, "A spiking neural network (SNN) forecast engine for short-term electrical load forecasting," *Applied Soft Computing*, vol. 13, no. 8, pp. 3628–3635, 2013.
- [24] V. Sharma and D. Srinivasan, "A spiking neural network based on temporal encoding for electricity price time series forecasting in deregulated markets," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

- [25] D. Reid, A. J. Hussain, and H. Tawfik, “Spiking neural networks for financial data prediction,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–10.
- [26] L. Yang and T. Zhongjian, “Prediction of grain yield based on spiking neural networks model,” in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2011, pp. 171–174.
- [27] M. Schmuker, T. Pfeil, and M. Nawrot, “A neuromorphic network for generic multivariate data classification,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 6, pp. 2081–2086, 2014.
- [28] E. Tu, N. Kasabov, and J. Yang, “Mapping temporal variables into the neucube for improved pattern recognition, predictive modeling, and understanding of stream data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1305–1317, 2017.
- [29] N. Kasabov, *Evolving connectionist systems: the knowledge engineering approach*. Springer Science & Business Media, 2007.
- [30] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.
- [31] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, “Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum],” *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.
- [32] G. I. Webb, R. Hyde, H. Cao, H. Nguyen, and F. Petitjean, “Characterizing concept drift,” *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.
- [33] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, “Discussion and review on evolving data streams and concept drift adapting,” *Evolving Systems*, pp. 1–23, 2016.
- [34] I. Žliobaitė, M. Pechenizkiy, and J. Gama, “An overview of concept drift applications,” in *Big Data Analysis: New Algorithms for a New Society*. Springer, 2016, pp. 91–114.
- [35] A. Wibisono, W. Jatmiko, H. A. Wisesa, B. Hardjono, and P. Mursanto, “Traffic big data prediction and visualization using Fast Incremental Model Trees-Drift Detection (FIMT-DD),” *Knowledge-Based Systems*, vol. 93, pp. 33–46, 2016.
- [36] “Madrid open data portal,” <http://datos.madrid.es>, accessed: 2018-01-31.
- [37] I. Laña, I. I. Olabarrieta, J. Del Ser, and M. Vélez, “On the imputation of missing data for road traffic forecasting: New insights and novel techniques accepted pending to be published,” *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 18–33, 2018.

- [38] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 2, pp. 139–166, 2004.
- [39] J.-M. Chiou, Y.-C. Zhang, W.-H. Chen, and C.-W. Chang, "A functional data approach to missing value imputation and outlier detection for traffic flow data," *Transportmetrica B: Transport Dynamics*, vol. 2, no. 2, pp. 106–129, 2014.
- [40] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial–temporal correlation," *Physica A: Statistical Mechanics and its Applications*, vol. 446, pp. 54–63, 2016.
- [41] L. Li, X. Su, Y. Zhang, J. Hu, and Z. Li, "Traffic prediction, data compression, abnormal data detection and missing data imputation: An integrated study based on the decomposition of traffic time series," in *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 282–289.
- [42] W. C. Ku, G. R. Jagadeesh, A. Prakash, and T. Srikanthan, "A clustering-based approach for data-driven imputation of missing traffic data," in *2016 IEEE Forum on Integrated and Sustainable Transportation Systems (FISTS)*. IEEE, 2016, pp. 1–6.
- [43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.
- [44] S. Li, Z. Shen, and F.-Y. Wang, "A weighted pattern recognition algorithm for short-term traffic flow forecasting," in *2012 9th IEEE International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2012, pp. 1–6.
- [45] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1786–1796, 2016.
- [46] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 308–324, 2015.
- [47] Z. Liu and S. Sharma, "Statistical investigations of statutory holiday effects on traffic volumes," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1945, pp. 40–48, 2006.
- [48] S. Schliebs, M. Defoin-Platel, and N. Kasabov, "Integrated feature and parameter optimization for an evolving spiking neural network," in *International Conference on Neural Information Processing*. Springer, 2008, pp. 1229–1236.
- [49] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Evolving spiking neural networks for audiovisual information processing," *Neural Networks*, vol. 23, no. 7, pp. 819–835, 2010.

- [50] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, “Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition,” *Neural Networks*, vol. 41, pp. 188–201, 2013.
- [51] S. M. Bohte, J. N. Kok, and H. La Poutre, “Error-backpropagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.
- [52] C. Enroth-Cugell and J. G. Robson, “The contrast sensitivity of retinal ganglion cells of the cat,” *The Journal of physiology*, vol. 187, no. 3, pp. 517–552, 1966.
- [53] S. Nirenberg and P. E. Latham, “Population coding in the retina,” *Current Opinion in Neurobiology*, vol. 8, no. 4, pp. 488 – 493, 1998.
- [54] M. J. McMahon, O. S. Packer, and D. M. Dacey, “The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-gabaergic pathway,” *Journal of Neuroscience*, vol. 24, no. 15, pp. 3736–3745, 2004.
- [55] S. M. Bohte, H. La Poutre, and J. N. Kok, “Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer rbf networks,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 426–435, 2002.
- [56] S. Schliebs, M. Defoin-Platel, S. Worner, and N. Kasabov, “Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models,” *Neural Networks*, vol. 22, no. 5, pp. 623–632, 2009.
- [57] Q. Yu, H. Tang, J. Hu, and K. C. Tan, “Rapid feedforward computation by temporal encoding and learning with spiking neurons,” in *Neuromorphic Cognitive Systems*. Springer, 2017, pp. 19–41.
- [58] S. Thorpe and J. Gautrais, *Rank Order Coding*. Boston, MA: Springer US, 1998, pp. 113–118.
- [59] S. Schliebs and N. Kasabov, “Evolving spiking neural networks: a survey,” *Evolving Systems*, vol. 4, no. 2, pp. 87–98, 2013.
- [60] C. Alippi, *Intelligence for Embedded Systems*. Springer, 2014.