Article

Exploiting Recurring Patterns to Improve Scalability of Parking Availability Prediction Systems

Sergio Di Martino [1,*,†] and Antonio Origlia [2,†]

1 DIETI, University of Naples Federico II; sergio.dimartino@unina.it
2 UrbanECO, University of Naples Federico II; antonio.origlia@unina.it
* Correspondence: sergio.dimartino@unina.it
† The authors contributed equally to this work.

Version May 8, 2020 submitted to Electronics

Abstract: Parking Guidance and Information (PGI) systems aim at supporting drivers in finding suitable parking spaces, also by predicting the availability at driver's Estimated Time of Arrival (ETA), leveraging information about the general parking availability situation. To do these predictions, the most of the proposals in the literature dealing with on-street parking require to train a model for each road segment, with significant scalability issues when deploying a city-wide PGI. By investigating a real dataset we found that on-street parking dynamics show a high temporal auto-correlation. In this paper we present a new processing pipeline that exploits these recurring trends to improve the scalability. The proposal includes two steps to reduce both the number of required models and training examples. The effectiveness of the proposed pipeline has been empirically assessed on a real dataset of on-street parking availability from San Francisco (USA). Results show that the proposal is able to provide parking predictions whose accuracy is comparable to state-of-the-art solutions based on one model per road segment, while requiring only a fraction of training costs, thus being more likely scalable to city-wide scenarios.

Keywords: Internet of Vehicles; Parking availability Predictions; Smart Mobility; Dataset Reduction; Clustering; Scalability

1. Introduction

Finding a parking space is one of the main concerns of urban mobility, as it is well recognised that a significant fraction of the traffic in crowded urban areas is originated by drivers cruising in search of a parking space [1]. The motivation behind this problem lies in that drivers have no knowledge about where there could be a free parking space matching their expectations. Thus, they have to roam, with significant consequences in terms of additional traffic, pollution, and drivers' wasted time [1,2]. Moreover, parking search also affects road safety, since drivers cruising for parking are distracted, and thus more likely to hit other road users [3].

Among the various types of Intelligent Transportation Systems, the *Parking Guidance and Information* (PGI) solutions, integrated within in-vehicle navigation systems or intended as mobile apps, aim at significantly reduce this problem, by guiding drivers directly towards streets (or parking facilities) with current or future higher availability of free spaces. To this aim, PGIs require parking availability information to work. When dealing with on-street parking, this information can be collected from stationary sensors, or by means of participatory or opportunistic crowd-sensing solutions from mobile apps [4,5] or probe vehicles [6–8]. The collected availability information is then aggregated on a remote back-end, to get a dynamic map of the parking infrastructure. This up-to-date map can be either pushed to the interested PGI users, or used to feed some prediction algorithms, to forecast the parking availability at the Estimated Time of Arrival (ETA) of a PGI user [9,10], to allow drivers to better organise their transport before their departures or during their trips [2].

In the last years, the availability of sensor techniques to collect real on-street parking availability data triggered many researches on proposing solutions to predict parking availability (e.g.: [9–13]), mostly using advanced machine/deep learning approaches. Results are encouraging, with prediction errors of available stalls in the range of 10 - 15%, on a city-wide scale (e.g [9,10]). Nevertheless, most of these approaches require to train one model for each road segment with parking stalls, with significant scalability issues when dealing with large urban maps, which can likely comprise hundreds of thousands of them. The problem was firstly highlighted by Zheng et al., who investigated the effectiveness and the computational requirements of three Machine Learning techniques, being even unable to obtain results for some settings *"due to the long computation time [...]"* [13, p.5]). To the best of our knowledge, only one paper introduced a preliminary solution to reduce the computational requirements for a service of on-street parking availability prediction [14], based on simple clustering solutions.

To fill this gap, in this paper we present the results of an investigation meant to devise a pre-processing technique, leveraging the recurring patterns found in the dataset, to reduce the computational load required by an on-street parking prediction system, by minimising the prediction models and the training examples. More in detail, as a starting point we analysed the availability trends in a real on-street parking dataset from the municipality of San Francisco (USA), and we found that each road segment has a high temporal auto-correlation over itself, and a high cross-correlation among different trends. From this finding, we propose a pre-processing pipeline for parking prediction system where we firstly group together road segments showing a high similarity in parking availability trends, by means of a hierarchical clustering technique. The next step should be to train a shared prediction model on each of these clusters, to forecast future parking availability, but, as each of these clusters might include hundreds of road segments, with a potentially overwhelming number of training examples, we propose the use of the *Kennard-Stone* algorithm [15], to prune the training set, by maintaining only the most representative examples. Only after this training set filtering, on top of these reduced examples, we train a regressor, like for instance an SVR or a Deep Neural Network. As an additional observation, we found that on-street parking dynamics can be very fast: since each road segment has a limited number of parking spaces[1], each change in the sensed availability has a deep impact on the occupancy percentage. Therefore, a misreading leads to abrupt changes in sensed availability, that are not related to the actual state. This is a common scenario, as current state-of-the art on-street sensing technologies suffer of an intrinsic amount of misreadings, quantifiable in at least 10% probability [16–18]. Thus, this kind of data is challenging for machine learning techniques, both for model training and performance evaluation, since these time series exhibiting strong, abrupt and frequent changes from one sampling instant to the other. To cope with this *noise* in the data, masking the general trend underlying the measurements, which is the real information [12], in our pre-processing pipe-line we propose also the use of an optional filtering step, performed by means of specifically configured Kalman filters [19].

To assess the effectiveness of the proposed solution, we conducted an empirical evaluation on a real dataset of five weeks of on-street parking data from the SFPark project in San Francisco [20], covering 321 road segments, available at [21]. We evaluated the prediction performances of a Support Vector Regressor, with an horizon of 30 minutes, considering the solution with and without Kalman filters, in combination with three different filtering levels of the Kennard-Stone algorithm. Let us note that more advanced prediction techniques might provide better prediction performances, but the focus of this investigation is to understand and quantify the impact of the proposed pipeline to prune the

[1] For instance, over 500 road segments in San Francisco (USA) downtown, the most frequent number of parking stalls per road segment is 6

dataset, rather than achieving the best possible predictions. Results show that the proposed on-street parking availability prediction solution performs in a way that is comparable with state-of-the-art techniques based on a model per segment, while requiring a fraction of the computational efforts. Indeed, by grouping the 321 segments in just five clusters, each with 4000 training examples of filtered data, provided practically the same prediction error (in terms of RMSE) of a model for each of the 321 models, each with more than 6000 examples, thus reducing by almost two orders of magnitude the required training efforts.

The main contributions of the paper are:

1. We provide the first analysis, to the best of our knowledge, on the temporal auto-correlation phenomenon for on-street parking availability.
2. We propose a technique to highly reduce the computational requirements of a parking availability prediction service, making it potentially scalable to a city-wide level, providing empirical evidence that it is able to provide parking predictions whose error is comparable with state-of-the-art solutions, based on one model per segment, at a fraction of their training costs.
3. We provide empirical evidence that, by applying a fast filtering step, the computational requirements for training can be further reduced.

The remainder of this paper is structured as follows: in Section 2 we present the related work on data-driven parking space prediction. In Section 3 we provide a detailed analysis on the temporal dynamics of parking availability. In Section 4 we present the approaches to predict the parking availability based on training data reduction. In Section 5 we describe the experiment design to assess the proposed approach, with the results we obtained. Finally, in Section 6 some conclusions are outlined, together with some future research directions.

2. Related Work

Parking Guidance and Information (PGI) systems require detailed parking space availability information [4,5] in order to support drivers. Occupancy information can be easily obtained for multi-storey car parks (also known as parking garages, or *off-street* parking) with controlled accesses [5,22], and consequently the most of the data-driven researches addressing the problem to predict parking space availability in the near future deal with this kind of facilities by means of different types of prediction techniques [23–26]. On the other hand, monitoring in real-time parking occupancy for on-street spaces is an open issue, with many challenges still to be faced [2].

2.1. IoT solutions to Sense On-Street Parking Availability

To date, two main on-street parking availability monitoring strategies are reported in the literature, both based on *Internet of Things* (IoT) approaches: one based on stationary sensors, and one on mobile sensors [2]. In the former group there are devices like magnetometers installed in the roadway below each on-street parking spot [27], or cameras on poles, overseeing parking lanes [28]. This approach produces availability information at a constant rate, but it is very expensive to deploy and maintain on a city-wide scale [4,29]. The other strategy exploits participatory or opportunistic crowd-sensing [30,31] via mobile apps [5] or probe vehicles [18]. Opportunistic mobile apps use smartphone sensors to estimate the subject state [32], or mode of transportation (e.g. driving or walking) and, from this information, to infer parking availability [22]. These apps are cheap to deploy, but require very high penetration rates to obtain an adequate amount of parking availability information [7]. On the other hand, probe vehicles, giving rise to *Internet of Vehicles*, can represent an advantageous trade-off between deployment costs and coverage of on-street parking monitoring. Many works proved that standard equipment on modern vehicles, like side-scanning ultrasonic sensors [18,33] or windshield-mounted cameras [34], can be used to detect free parking spaces along their routes, with a pretty high accuracy [17,18].

Despite specific pros and cons, all these IoT-based approaches to monitor on-street parking data are characterised by issues in the quality of the data coming from sensors, which can present a significant amount of noise and sudden variability. An empirical evidence of these problems can be drawn by the large experimental parking project *SFPark* project, ran in 2011 by the San Francisco, whose costs exceeded $46 million. In the project, about 8,000 parking spaces were equipped with specific sensors embedded in the asphalt, broadcasting availability information [27]. At the end of the project, many problems with the sensors were reported. As an example of misdetections, they found that *"High levels of electromagnetic interference from overhead wires, underground utilities, and other sources made it more difficult than expected for the magnetometer sensors to properly detect vehicles. [...] During three years of operation, interference remained pervasive and unpredictable."* [16]. As for the abrupt changes in the values, changes observed at each time sample are reflected as steps in a square wave with a magnitude of about 10%. Spikes and changes of direction due to cars leaving and arriving at subsequent observation times or due to the reported electromagnetic interference are visible, too. This problem becomes exacerbated when considering road segments with a very small number of parking stalls, which are pretty common in the dataset from San Francisco that we used for our experiments, described in details in Section 3.1. Indeed, in that dataset, the average number of parking stalls per segment is 6, meaning that each parking/leaving event produces a change in the relative availability of about 17%. This scenario is in contrast with what is theoretically and experimentally known in the literature on parking, i.e. that there is a strong temporal correlation in the availability, which should not change drastically within around 30 minutes (e.g.: [9]). The consequences of this noise are twofold. On one hand, it becomes difficult to train a generalised model for meaningful predictions. On the other hand, it becomes also problematic to evaluate the prediction performances obtained by such a model, since the test set is noisy, too.

2.2. Solutions for On-Street Parking Availability Predictions

Focusing on researches conducted on predicting on-street availability, they are by far less than those of off-street, for two main reasons: (I) it is hard to find suitable datasets for the experiments, and (II) *"the prediction of parking availability for on-street parking is more difficult than off-street parking since the variance of on-street parking is relatively higher"*[35]. Zheng et al. [13] compared three different prediction techniques, Regression Trees, Neural Networks and Support Vector Regression, on the dataset from SFpark and from the municipality of Melbourne. Differently from current work, they applied SVR on raw data, with a single prediction horizon of 15 minutes. Rajabioun and Ioannou [10] proposed a technique to predict on-street parking availability based on the SFpark project data, by using multivariate autoregressive models considering both spatial and temporal correlations of parking availability. More recently, Monteiro and Ioannou [9] compared four different techniques to predict on-street parking availability, based on a new dataset coming from the municipality of Los Angeles. In [12] we preliminary investigated the idea of reducing noise in the data before running predictions. By means of a 2-step technique, including a specifically-customised Support Vector Regression smoother, we were able to outperform, in a statistically significant way, parking availability predictions obtained using standard regression techniques, as the one presented in [13]. In a subsequent paper ([36]) we extended that work by defining and assessing two smoothing techniques, characterised by significantly different computational requirements. Moreover, we considered also new prediction techniques, including one of those described in [9], to better evaluate the achievable performances of the entire solution. The solution we propose in the current paper includes the smoothing solutions defined in [36].

It is worth noting that the most of the related works in the literature propose the use of advanced prediction techniques to get good predictions, with strong generalizability properties, like Support Vector Regression (SVR), Neural Networks [13], Autoregressive models [10], and so on. The drawback is that these methods have high computational requirements, making difficult to deploy these solutions to a nation-wide scale. For example, Zheng et al. [13] were unable to produce results with SVR on few hundreds of road segments in San Francisco, "*due to the long computation time*". Another key denominator of all these papers is that they use a number of models which is close or equal to the number of road segments with parking stalls. This also leads to significant computational issues, making these solutions pretty hard to scale up to a nation-wide, or even to a city-wide level.

To the best of our knowledge, the only paper investigating how to reduce the number of models needed to predict on-street parking availability is the one presented by Richter et al. [14] in 2014. In that paper, the authors evaluated different strategies to predict parking space availability, using a sample of data from the SFpark project, with the goal to minimise the number of prediction models, and thus the total space required to store data, by using different spatial and temporal clustering strategies. Nevertheless, that paper was meant for a totally different system architecture. Indeed, authors focused on proposing something suitable to be fitted in the on-board navigation device of a vehicle, meant as an off-line solution, based exclusively on historical data, without any dynamic update. Moreover, they designed the solution as a classification problem, predicting a range of parking availability (high, medium, low), rather than as a regression model, which can lead to much more refined solutions.

3. An Analysis of an On-street Parking Availability Dataset

Recurring dynamics, in time series, present an important opportunity to be exploited for prediction systems. Indeed, even if machine learning algorithms are capable of capturing these dynamics, by knowing in advance the existence of significant temporal regularities in the data, a system designer may develop more efficient processing pipelines. More in detail, in this scenario, many techniques are available in the literature to help reduce the size of the training sets and/or the number of needed prediction models, thus reducing the computational requirements of the processing pipeline. These techniques are often employed for traffic predictions (e.g.:[37,38]), but, to the best of our knowledge, they have been applied to on-street parking predictions only in one preliminary paper [14], also due to the lack of investigations focused on qualitative analyses of parking dynamics. Specifically, in [14], the presence of day-by-day, and weekdays/weekend recurring patterns was highlighted.

As a consequence, in this paper we start by providing an analysis on real data about on-street parking availability dynamics, to verify and quantify the presence of recurring temporal patterns in the data. In the following, we describe the dataset we collected about on-street parking availability from the Municipality of San Francisco (USA). We made available a part of the which is an extension of the one provided in [21]. Then we discuss the analysis of these data, that allowed us to get some insights on parking dynamics, motivating the proposal presented in this paper.

3.1. The Considered Dataset

A common problem when conducting experimental evaluations for approaches dealing with the on-street parking domain is the lack of suitable datasets. Indeed, while many smart cities are collecting parking data (e.g. Santander (Spain) [39] or Los Angeles (USA) [9]), usually these data are not publicly available. For our study, on-street parking availability data was collected from the *SFpark* project [27]. In 2011 the San Francisco Municipal Transportation Agency started a large experimental smart parking project, called *SFpark*. The main focus of this project, whose costs exceeded $46 million, was the improvement of on-street parking management in San Francisco, mostly by means of demand-responsive price adjustments [27].

One of the key points of the project was the collection of information about parking availability in six districts in San Francisco between 2011 and 2014. To this aim, about 8,000 parking spaces were equipped with specific sensors embedded in the asphalt of some pilot and control areas, periodically broadcasting availability information. Even though 8,000 equipped stalls is a remarkable number, this is less than 3% of the total number of on-street legal parking spaces in San Francisco [27]. These numbers make clear the problems and the costs to scale the instrumentation of on-street parking stalls to a city-wide dimension.

The SFpark project made available a public REST API, returning the number of free parking spaces and total number of provided parking spaces, for each involved street segment in the pilot

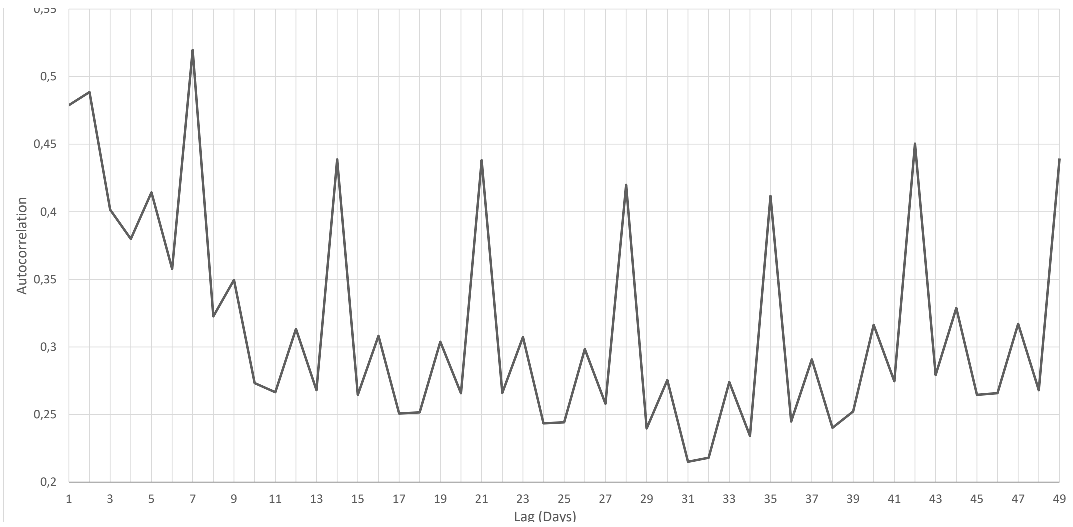

Figure 1. Average autocorrelation values of all road segments during period 3.

areas. By exploiting those APIs, we collected parking availability data from middle of June, 2013 to end of December, 2013. In some cases, due to malfunctions in the collection procedure, we lost some weeks, giving rise to three trunks of data. Thus, the final dataset we used in our investigation consists of three subsets of data including, respectively, 5 weeks (Period 1), 6 weeks (Period 2) and 14 weeks (Period 3). Only road segments having at least 4 parking spaces are considered, in this work. Also, road segments that were never occupied for more than 85% of their capacity or showed missing/constant readings for more than 3 days were removed from the dataset, as we assumed that sensors were severely malfunctioning. The final number of considered segments is 321.

As for the distribution of provided parking spaces per road segment[2], the most frequent number of parking stalls per road segment is six, (8.8 % of the total), while the average is about 7.9. These numbers show that long parking lanes seldom exist in the evaluation regions and therefore each parking/leaving event has a relevant impact on the *parking availability rate*, which is defined as the ratio between the free and total stalls.

The reader interested in further statistical details on the distribution of available/free parking spaces per segment is referred to our previous work [21].

3.2. Recurring Patterns in the Dataset

Starting from the observations in [14], we looked for temporal regularities in the data considering a temporal granularity at a day level. More in detail, we used the *autocorrelation* operator to detect recurring patterns for each road segment. This operator is used to evaluate at which lag a signal is maximally similar to itself. In presence of periodic dynamics, the autocorrelation plot will show strong local peaks, corresponding to lags at which the signal has a high recurrence. In our analysis, we searched for lags in a range from one day up to half the days available in each considered data collection period. This is to keep the number of superimposing samples sufficiently high to obtain reliable autocorrelation values. As an example, Figure 1 shows the plot of the average autocorrelation values for all road segments in San Francisco during Period 3. The spikes due to the recurring patterns at 7 days lag are clearly visible in the autocorrelation curve, indicating that the on-street parking phenomenon has a recurring dynamic with a period of one week.

---

2 In the context of the SFpark project, a *road segment* (also named *block face*) is defined as one side of a road between two intersections

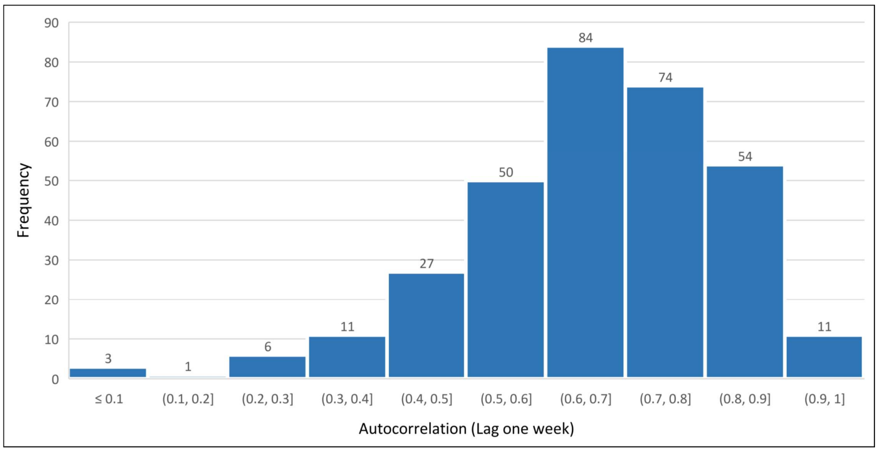

Figure 2. Autocorrelation frequency distribution over Period 1

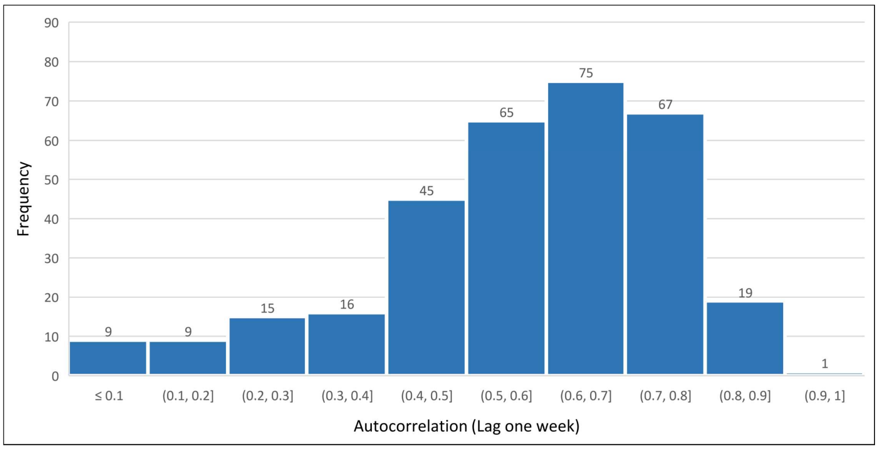

Figure 3. Autocorrelation frequency distribution over Period 2

Considering the whole set of segments in the dataset and a 7 days lag, histograms shown in Figures 2, 3 and 4 highlight that the majority of the road segments present a consistent pattern repeating itself at 1 week period.

Having confirmed that the most of the road segment has a recurring pattern over a 1 week lag, an immediate conclusion that may be derived from this analysis is that it could be possible to predict the occupancy value for the current time and day by replicating the observation collected at the same time during the same day of the preceding week. Should this strategy pay off, it would be useless to proceed with machine learning at all. A simple preliminary experiment testing this hypothesis was, therefore, conducted to assess the possibility that the *naive* strategy is adequate to predict occupancy rate. The boxplot of the RMSE value obtained using this strategy is shown in Figure 5 and it highlights that the prediction error, is more than two times the one found in [36], which used the same dataset. Moreover, also the distribution of the RMSE value is very large, making the predictions unreliable. As a consequence, even if recurring trends are present, there is still the need for more advanced prediction approaches. In the following we propose a parking availability prediction technique meant to exploit this characteristic, in terms of a strategy aimed at significantly reducing computational requirements.

4. The Proposed Processing Pipeline

Many solutions presented in the on-street parking prediction literature use a pipeline like the one shown in Figure 6 [9,10,13,35]. In detail, a dataset of historical parking availability contains the examples to train a supervised predictive model. Depending on the employed prediction technique, for each road segment, the dataset is windowed to generate a set of records, i.e. the *features* for the

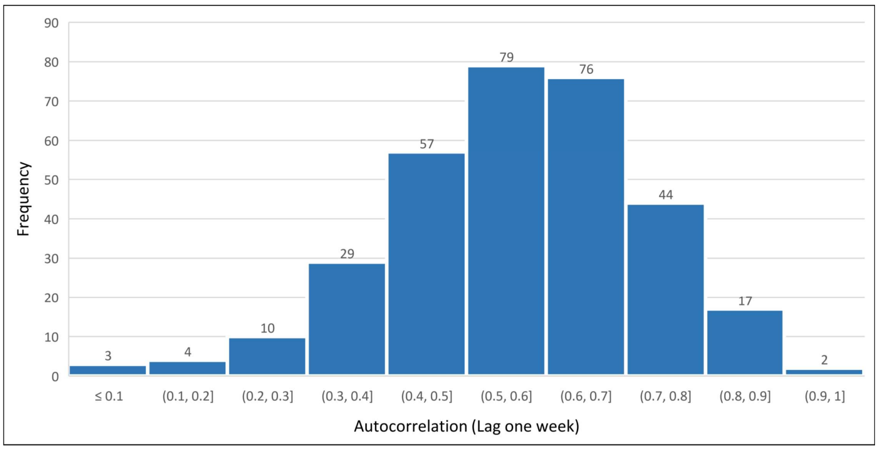

Figure 4. Autocorrelation frequency distribution over Period 3

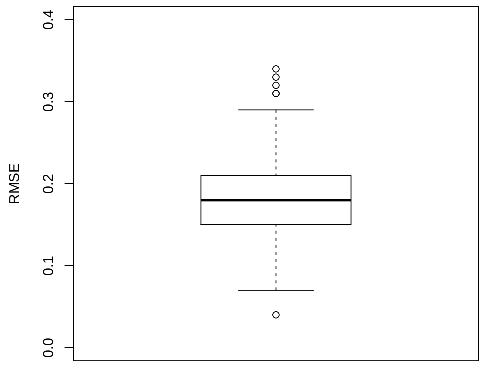

Figure 5. Boxplot of the RMSE obtained by replicating the occupancy value, for each street segment, of the observation found at the same day and time of the previous week

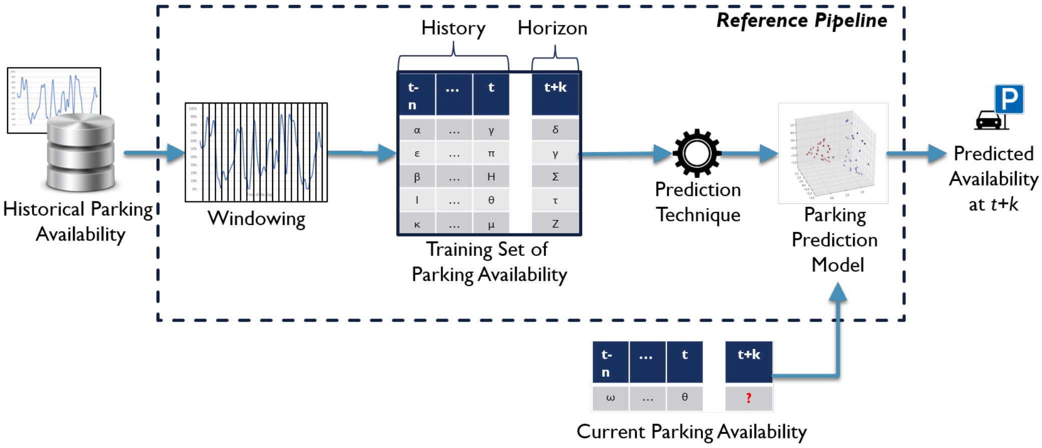

Figure 6. The reference parking prediction processing pipeline, as adopted in many related works. This pipeline is instantiated for each per road segment.

regressor, containing a sequence of parking availability information in the time interval $[t-n, t]$, $X = \{A_{t-n}, ..., A_{t-1}, A_t\}$, referred to as *history* in the rest of the paper.

A further point $Y = A_{t+k}$ in the record represent the observed availability at $t+k$, which is the target value for regressor, referred to as *prediction horizon*. The regression technique is thus trained to learn, for each road segment, a *model* representing the relationship between the parking history $[t-n, t]$ and the prediction horizon $t+k$ on these examples. Specifically, the historical data can be windowed at the desired length (for example using a history of 60 minutes in the past and predict availability at 30 minutes in the future), to generate the examples (i.e. the training set) on top of which a regressor can be trained, as proposed by Zheng et al. [13]. Once a PGI user requests a prediction of parking availability at a given time $t+k$ in the future for a given road segment, the PGI queries the prediction model with the parking data collected from sensors in the last n time frames for that segment, and obtains as output the availability prediction for $t+k$. Let us note that training data in this scenario can be either raw or smoothed. In the rest of the paper, this *Reference Pipeline* will be referred to as RP.

The key limitation of RP is that a model is required for each segment to be monitored. Most of the related papers deal with a few hundred road segments, still highlighting computational issues (e.g. [13]). To give a reference, the map of the urban area of San Francisco from OpenStreetMap includes more than 200,000 road segments, making it very hard for the solutions proposed in the literature to scale up to a city-wide dimension. To face this issue, we propose a strategy, intended as an evolution of RP, by adding two pre-processing steps:

1. Reduce the number of models, by clustering road segments with similar parking availability dynamics;
2. Reduce the number of training examples, for each cluster, by selecting the n most informative ones.

The key advantage of using a clustering technique is that the number of models to train grows sub-linearly with the number of road segments to monitor, with clear computational advantages. Thus, the final solution will be more likely to be able to scale to a city-wide level.

As in [36], this pipeline can include also an optional step to smooth data, to compensate the potential presence of strong noise caused by the sensing solution.

4.1. Clustering Road Segments

The first step to exploit recurrent temporal dynamics in the data consists in aggregating road segments based on the correlations among their occupancy rate curves. Specifically, a cross-correlation matrix C is computed considering the smoothed occupancy rate curves among all segments in the dataset. Being $C_{i,j}$ the cross-correlation value between the $i-th$ and the $j-th$ road segments, the *Pairwise Distance Matrix D* is obtained by computing $D_{i,j} = 1 - C_{i,j}$, so that the higher the correlation, the lower the distance among the considered segments. On the basis of the data contained in D, the hierarchical clustering *Ward Variance Minimization Algorithm* is used to obtain the segments clusters. A Hierarchical clustering approach was selected as the number of clusters is not known *a priori*. The algorithm is used to iteratively group the road segments, by minimising the internal variance of each cluster [40], where the distance between two clusters u and v is defined as follows:

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T}d(v,s)^2 + \frac{|v|+|t|}{T}d(v,t)^2 - \frac{|v|}{T}d(s,t)^2} \tag{1}$$

where u is the new cluster generated by merging two clusters s and t, v is every other cluster different from u, on which we compute the distance from u, and $T = |v| + |s| + |t|$.

The output of the hierarchical clustering algorithm is a dendrogram, which can be cut at different levels of similarity, to get different groupings, where the higher the cutting value, the lower is the number of obtained clusters. Many strategies are described in the literature to select the cutting threshold, often being domain-dependent [41]. In our case, we adopted a simple criterion, using the default strategy implemented by both *SciPy* and *Matlab*, where the cutting threshold is computed as 70% of the maximum linkage distance among clusters.

Given the considered problem, through this clustering, we are able to group road segments that behave in a similar way, from an on-street parking dynamics point of view. The subsequent problem is how to train a single parking prediction model for each cluster, representative for all the segments in that cluster. Indeed, for a single cluster, if we simply merge together all the windowed examples from all the road segments belonging to that cluster, we will obtain a very large training set, containing a lot of very similar examples, as the road segments were grouped together on the basis of the similarity between their temporal dynamics: this will lead to very redundant datasets. While machine learning algorithms are, of course, designed to manage this situation, computational requirements can be greatly reduced if redundant information is filtered out of the dataset before the training phase. This is what we propose in the subsequent step.

4.2. Training Set Reduction

To obtain a sub-sample of the dataset in each cluster, that prioritises diversity with respect to the amount of data, we propose the use of the Kennard-Stone [15] algorithm. This is a widely used technique, designed to select the set of n most different examples from a given dataset, using the Euclidean distance as a reference measure. The rationale behind the use of the Kennard-Stone algorithm is to obtain a set of examples that is maximally informative for each cluster, rather than uniformly distributed like the set that could have been achieved by random sub-sampling. Indeed, this is also in line with the way Support Vector Machines represent prediction models, through the identification of informative support vectors.

Thus, the procedure followed by the algorithm can be summarised as follows, for each cluster:

- Find the two most separated points in the original training set;
- For each candidate point, find the smallest distance to any already selected object;
- Select the point which has the largest of these smallest distances.

In this paper, we considered different values for n in order to evaluate how much the dataset used to train the model dedicated to each cluster can be reduced while limiting performance drops.

4.3. Kalman filters

As reported in the SFPark description, data provided by the sensors were affected by noise due to multiple factors. In our previous works, we considered, for evaluation purposes, the trend line, computed as an SVR model fitting the raw data, as a target for predictions [12]. This is because, at the decision level, it is more important to understand the underlying behaviour of the temporal series rather than predicting the exact occupancy of parking slots in a specific road segment. This is particularly important in the considered case, as the reported number of parking slots is affected by noise so that, by considering the occupancy rate, strong jumps in the series may be caused by random events. The SVR model representing the underlying trend, however, is computed using the full curve so that, while it is possible to use it as a prediction target, it is not possible to use it to provide features to machine learning algorithms. In order to approximate the trend line and filter out as much noise as possible, the proposed technique makes use of online Kalman filters.

Kalman filters are a well-known unsupervised approach to estimate systems' states in presence of missing and noisy observations [19]. While being relatively simple in their formulation, they possess a number of practical advantages. First of all, Kalman filters can be trained in a fast way without assuming the use of big data. Also, once the model is trained, it does not require significant memory space nor computational power to be queried and response time is fast. It is often useful, in the field, as it can handle missing observations and it can be continuously updated as data arrives. Kalman filters estimate the state of a system in terms of affine functions of state transitions and observations. A Kalman filter is entirely defined by its initial transition matrix A and by its covariance matrix Q. Optionally, in the case of noisy observations, a covariance matrix R can be provided to describe Gaussian noise in the observations. These matrices are continuously updated as more data arrive and represent the model by themselves. It is therefore important to use domain knowledge, when designing Kalman filters, to provide an initial state that reasonably approximates the behaviour of the system, leaving fine tuning to training.

In this work, we use the same configuration of the Kalman filters we described in [36] to compensate the problem that, in the case of on-street parking, raw observations are affected by random events that end up *masking* the underlying dynamics of street segments. The filter uses, for each road segment, the total number of parking spaces to estimate the Gaussian falloff of the true state probability space, centred on the last observation. To estimate the transition covariance matrix using the dynamics of each road segment, as observed in the training set, we introduce use the Expectation-Maximisation approach. The parameters are then used, using a sliding windows approach, to simulate online state estimation with a Kalman filter on each road segment. The reader is referred to [36] for more details about the Kalman filters configuration. An average RMSE of 0.05 between the Kalman curve and the trend curve was obtained on the dataset and an example comparison of the three curves is presented in Figure 7.

5. The Empirical Evaluation

In this section we describe the experimental protocol, in terms of experiment design and employed metrics. Then we present and discuss the obtained results.

5.1. Experimental Design and Configurations

To the best of our knowledge, in the literature there is no other work exploiting the dataset we used in our experiments that can be used as a benchmark. Rajabioun and Ioannou defined a spatio-temporal parking prediction model on data they collected from the SFPark APIs, but they used a different sampling rate and a different time frame of data collection w.r.t. to our dataset [10]. As a consequence, since no direct benchmark are available in the literature, to assess the effectiveness of the proposed approach, we had to define two **baselines**: the first one is trained on raw data, as described in the RP approach, while the second one is on smoothed data using Kalman filters. As for the regression technique to adopt, Zheng et al. compared the effectiveness of three solutions, namely Regression Trees, Support Vector Regression (SVR) (with RBF kernel and no hyper-parameter optimization), and Neural Networks (NN) on SFPark data. They found that the first two techniques performed very similarly, always outperforming NN. Consequently we chose to adopt SVR, in its

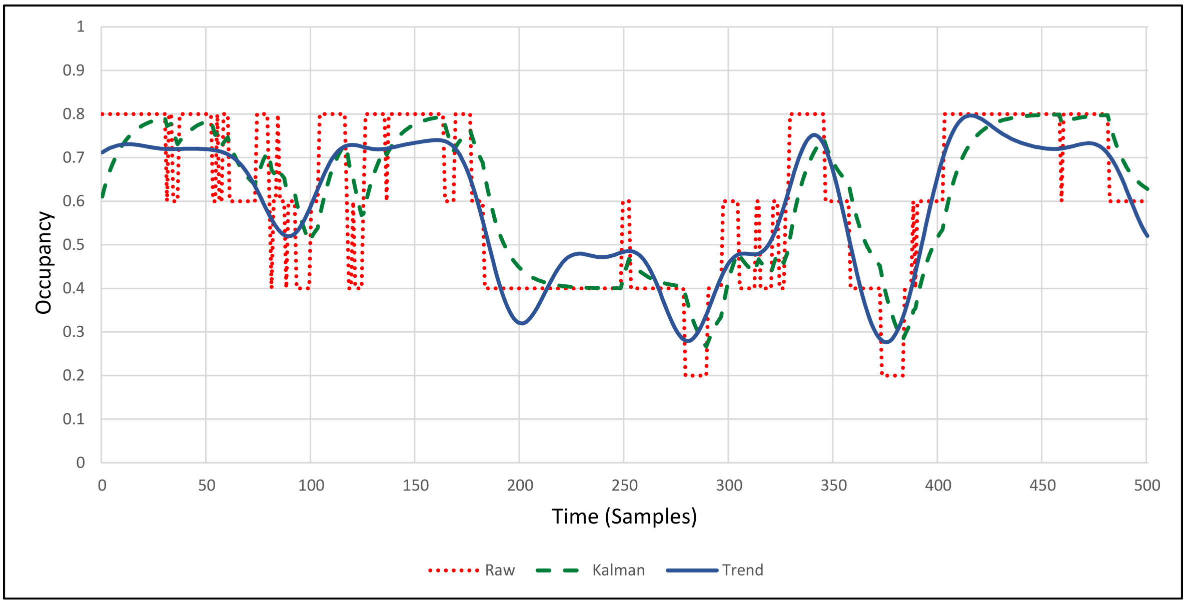

Figure 7. Comparison among raw data (red), SVR trend (blue) and Kalman online estimates (green)

implementation provided by the LibSVM library [42] to get the predictions, in combination with an ad-hoc technique to tune hyper-parameters. In particular, to find the optimal combination of C and γ parameters, we performed an inner cross validation on the training set, where 20% of the training set was used as development set. In this phase, we used a split validation protocol, so that the earlier part of the curve was considered to train the candidate models and the later part was used to evaluate the performance. The optimisation criterion we choose is the minimisation of the *Root Mean Square Error* (RMSE) on the development set and the ϵ parameter is fixed at the LibSVM default value (0.1). Once the optimal combination of the parameters was found, the final SVR model was trained using them on the full training set and evaluated on the test set.

We considered a combination of three different amounts of historical data (5, 30 and 60 minutes) to predict parking availability with an horizon of 30 minutes. Other than this historical parking availability data, we also associated the *TimeOfDay* feature to each data sample. By clustering together road segments using the similarity of their temporal dynamics, the hypothesis is that the number of examples needed to train a prediction model for each cluster is reduced. To evaluate if the expected effect is present and its strength, we considered different sample sizes for the Kennard-Stone algorithm. Specifically, results obtained using 1000, 4000 and 16000 samples per cluster are presented in the following. As for the baselines, we tested the performance obtained both with the raw and with the Kalman features.

5.2. Metrics

The prediction quality for decision level systems is not influenced only by the estimated average error, but also by the expected stability of this error. When evaluating performances on the road segments included in the entire dataset, it is important to be able to assume that the predictor's performance on all segments is approximately the same, so that uniform management strategies can be developed in an informed way. In this paper, we introduce a specific measure designed to take into account, other than the expected prediction error, its stability, too. In this way, solutions leading to less skewed distributions in RMSE values are preferred.

More in detail, let's consider the [0-0.2] interval to represent the distribution of RMSE values obtained on each road segment in the dataset. This interval is discretised in 10 bins so that the probability of each bin corresponds to the fraction of road segments for which the RMSE value falls inside that bin. Formally, if x is vector of RMSE values and n_i the number of road segments showing an RMSE value falling inside the $i-th$ bin, the probability of the $i-th$ bin is computed as

$$p(bin_i) = \frac{n_i}{|x|} \quad (2)$$

The Normalised Entropy H_N is, then, defined as

$$H_N = \frac{-\sum_i (p(bin_i)\ln(p(bin_i))}{\ln(N)} \quad (3)$$

where N is the total number of bins. Then, a quality measure based on the entropy of the RMSE distribution is defined as

$$Q_H = 1 - H_N \quad (4)$$

Similarly, a quality measure based on RMSE is defined as

$$Q_{RMSE} = 1 - RMSE \quad (5)$$

The final quality measure F is defined as the harmonic mean of Q_H and Q_{RMSE}, to privilege solution offering the best balance between average RMSE and distribution compression.

$$F = \frac{2}{\frac{1}{Q_H} + \frac{1}{Q_{RMSE}}} \quad (6)$$

5.3. Results

The results of the two baseline prediction approaches are reported in Table 1, together with the RMSE, Entropy and F metrics, while the boxplots summarising the performance obtained with these configurations are shown in Figure 8. From these numbers, we can see that raw and smoothed solutions are very close in terms of RMSE. The introduction of Kalman filters reduces Entropy when using 5 minutes of historical data, and increase it at 60 minutes.

The application of the hierarchical clustering produces the dendrogram presented in Figure 10, with an automatically computed value of 5 as the number of recommended clusters, following the criterion described in Section 4. Figure 11 shows the clusters distribution over the map of San Francisco provided by OpenStreetMap: while spatial patterns can still be observed, as it is to be expected, the image shows that similar temporal dynamics can occur in different parts of the city, highlighting that the same trained model can be used to manage spatially distant road segments.

In the following we report the prediction performances of the proposal with the three considered values for the n parameter of the Kennard-Stone algorithm, namely 1000, 4000 and 16000. For the case of just 1000 training samples per cluster, a very minor fraction of the original dataset, both the tests with the raw and Kalman features are worse than the baselines: RMSE is just slightly higher than the reference values, but the entropy values highlight that the error distribution is larger, so that the reliability of the results is reduced for decision-level systems. The details of the results with a sample size of 1000 are shown in Table 2 while the corresponding boxplots are shown in Figure 9.

The configuration using 4000 training samples per cluster shows that, for the setup using raw features, the performance is still far from the reference one. The Kalman-based solution, on the other hand, is stable across the considered history configurations and very close to the reference one. As a matter of fact, the clustered configuration, using 60 minutes as history, performs better than the baseline. This may be explained by considering that with fewer samples of higher quality, less noise is introduced in the dataset when the highest number of input features is used, as the Kalman filter already compensates for it. The details of results with a sample size of 4000 are shown in Table 3 while the corresponding boxplots are shown in Figure 12.

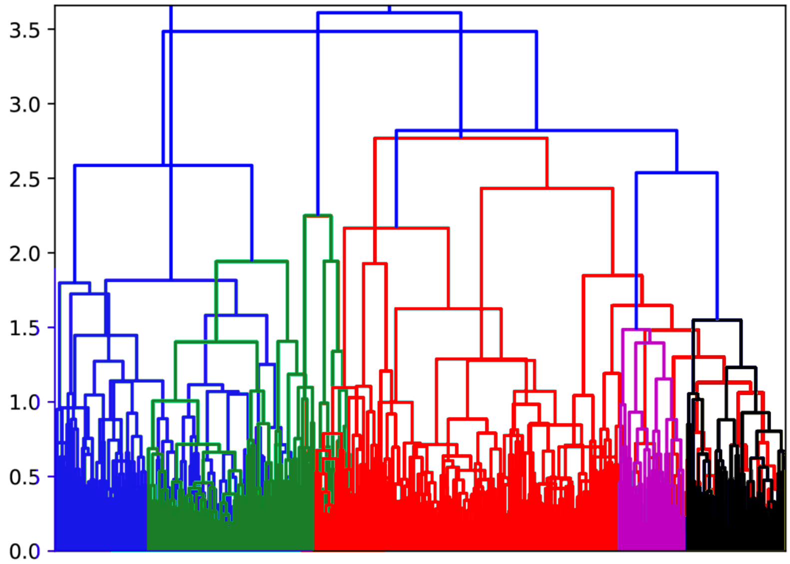

Figure 10. Dendrogram provided by the hierarchical clustering algorithm

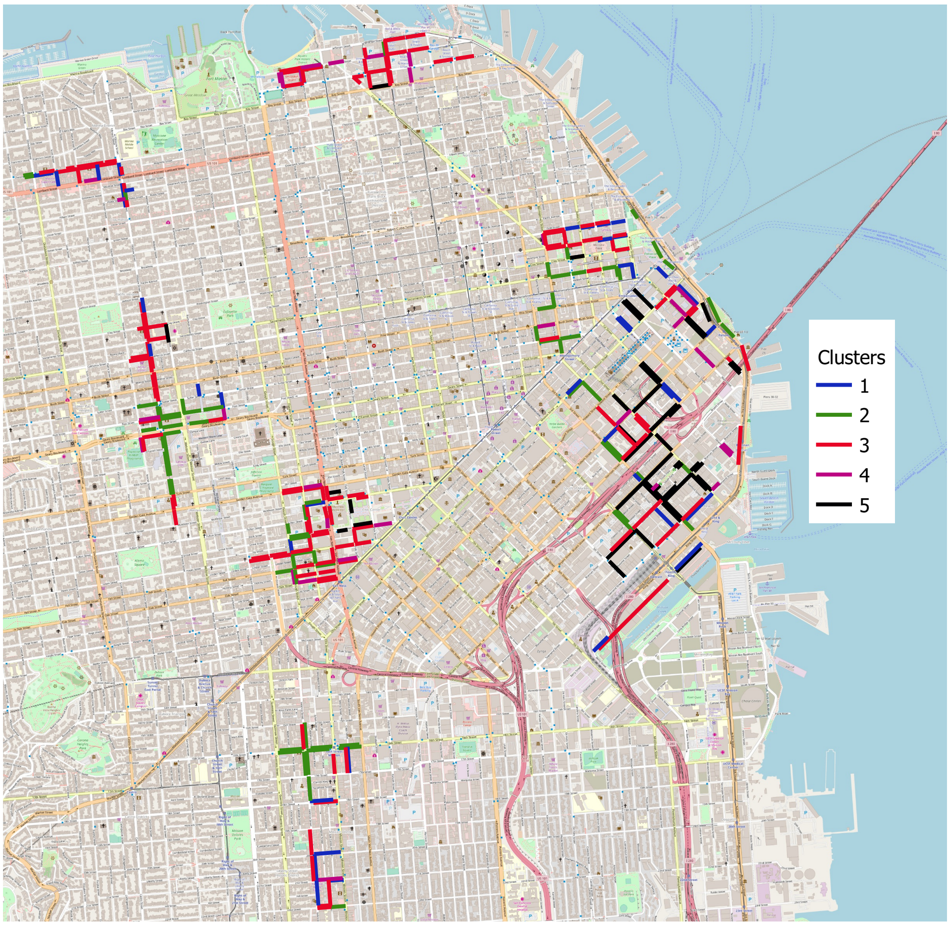

Figure 11. Geographic distribution of clusters among the considered road segments in the SFPark dataset

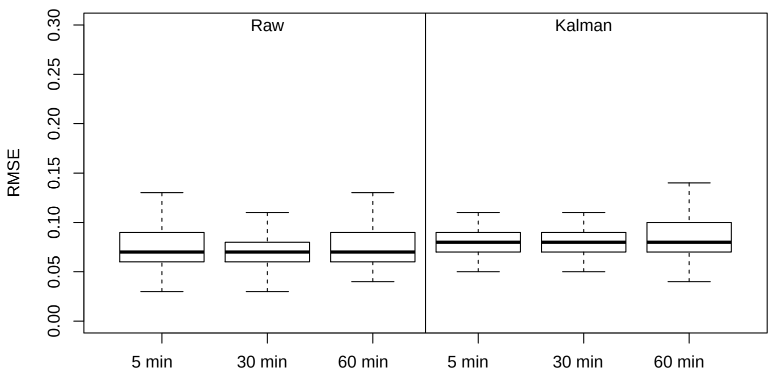

Figure 8. Boxplots of the baselines, considering one model for each road segment.

Feature Type	*History*	*Entropy*	*Mean RMSE*	*F*
Raw	5 min	0.56	0.07	0.59
	30 min	0.53	0.07	0.62
	60 min	0.56	0.07	0.59
Kalman	5 min	0.53	0.08	0.63
	30 min	0.53	0.08	0.62
	60 min	0.62	0.08	0.53

Table 1. Performance details of the baseline approaches, considering one model per road segment.

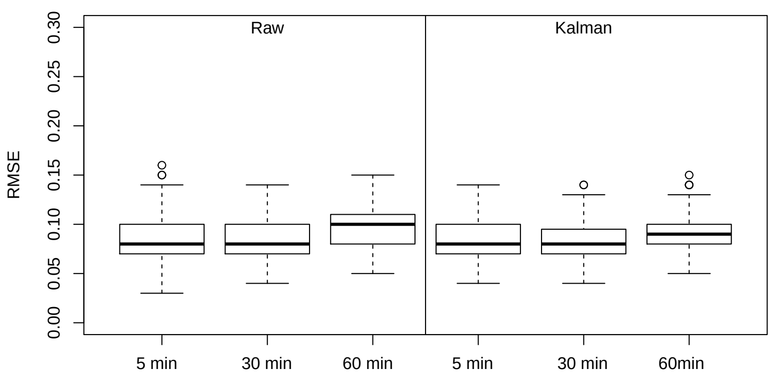

Figure 9. Performance boxplots with clustered segments and 1000 sample size.

Feature Type	*History*	*Entropy*	*Mean RMSE*	*F*
Raw	5 min	0.66	0.08	0.50
	30 min	0.62	0.08	0.53
	60 min	0.68	0.10	0.47
Kalman	5 min	0.64	0.08	0.52
	30 min	0.58	0.08	0.57
	60 min	0.61	0.09	0.53

Table 2. Performance details with clustered segments and 1000 sample size.

In the final configuration, considering 16000 training samples per cluster selected by the Kennard-Stone algorithm, provides almost always the best performances. When using raw features, in the case of 30 minutes of history, it provides basically the same results of the baseline. In the two other cases, results are close but worse that the raw baseline. On the other hand, when considering the Kalman features, this configuration is able to provide exactly the same performances of the baseline, while using an amount of training data being two orders of magnitude smaller than the baseline. The details of the experiments considering a sample size of 16000 are shown in Table 4 while the corresponding boxplots are shown in Figure 13.

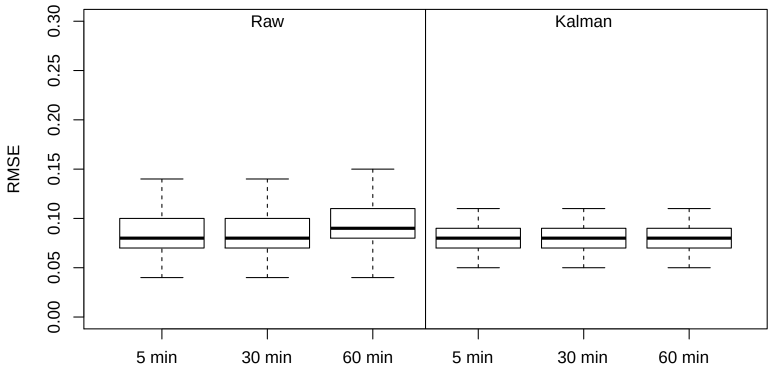

Figure 12. Performance boxplots with clustered segments and 4000 sample size.

Feature Type	*History*	*Entropy*	*Mean RMSE*	*F*
Raw	5 min	0.64	0.08	0.51
	30 min	0.62	0.08	0.53
	60 min	0.65	0.09	0.49
Kalman	5 min	0.53	0.08	0.62
	30 min	0.54	0.08	0.62
	60 min	0.54	0.08	0.62

Table 3. Performance details with clustered segments and 4000 sample size.

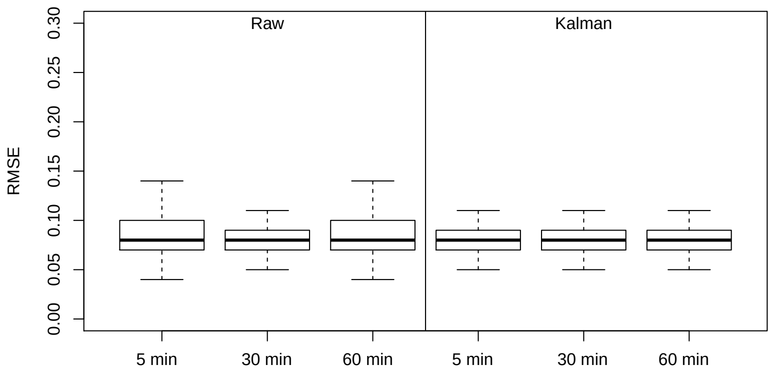

Figure 13. Performance boxplots with clustered segments and 16000 sample size.

Feature Type	*History*	*Entropy*	*Mean RMSE*	*F*
Raw	5 min	0.65	0.08	0.51
	30 min	0.54	0.08	0.62
	60 min	0.63	0.08	0.52
Kalman	5 min	0.53	0.08	0.62
	30 min	0.53	0.08	0.62
	60 min	0.53	0.08	0.62

Table 4. Performance details with clustered segments and 16000 sample size.

6. Discussion and Conclusions

Improving the effectiveness of on-street parking availability predictions is a key issue for Parking Guidance and Information (PGI) systems. The most of on-street parking availability prediction solutions presented in the literature are characterised by significant computational requirements, by learning a model for each road segment offering parking spaces, with considerable scalability issues.

The investigation we presented in this paper aims at evaluating if and how recurrent temporal patterns may be exploited to reduce the computational requirements of predictive approaches for on-street parking availability. Firstly, we have provided a quantitative and qualitative analysis of recurring patterns in the data collected from stationary sensors employed in a large experimental project in the Municipality of San Francisco (USA). This analysis highlighted that there are notable temporal recurrences in on-street parking availability dynamics, with an evident recurring pattern at 7 days lags. Anyhow, a *naive* replication strategy, where the parking availability prediction is obtained by repeating the situation sensed 7 days before, is not sufficient to obtain an adequate quality of the predictions.

We have, therefore, presented a processing pipeline to predict parking availability, meant to exploit these recurrences to lower computational requirements, by including clustering and training set reduction techniques. In particular, the clustering step is designed to group together segments with the similar temporal dynamics so that a shared model could be trained to predict parking availability for all the segments in the cluster. This implies that, in comparison with the strategy employed in similar works, training one model for each road segment, the number of models needed to cover the area of interest does not increase linearly with the number of segments, reaching volumes that may become hard to manage when large cities are considered. This provides important advantages from the scalability point of view: indeed, using temporal clustering allows to group together road segments that, although possibly far from a spatial point of view, exhibit similar dynamic occupancy patterns. This may be caused, for example, by qualitatively similar contextual situations, like the presence of residential or commercial areas.

Grouping road segments having similar (recurrent) occupancy patterns has the consequence that, when considering the windowed samples from all the segments included in a cluster, to form a single training set, many of these samples will be very similar to each other. To reduce the computational complexity of the training step, given a large dataset with redundant information, we applied a data reduction approach, using the Kennard-Stone algorithm, and investigated at which size the considered configurations of our system reach comparable performances with the ones obtained with the baseline approach.

The Kennard-Stone algorithm and the prediction quality can be significantly influenced by the amount of noise in the features. For this reason, we introduced an online Kalman filter to smooth the raw curve and reduce the influence of random events causing strong changes in the raw curve. The results we presented show that, with the Kalman-filtered data, the number of samples to be selected with the Kennard-Stone algorithm to reach the performance of the baseline is lower than the number needed using the raw features. This combination of temporal clustering, online filtering and data reduction techniques, therefore, allows to reach performances comparable to the ones obtained with the baseline approach while using a significantly lower number of models.

We conducted an experimental evaluation on a real on-street parking availability dataset from 321 road segments, in San Francisco, comparing our pipeline against a baseline where we trained a SVR model for each segment over 6,048 time frames, both for raw and filtered data. We had 5 clusters, and thus 5 models *vs* 321 of the baseline. Result shown that performances comparable with the baseline approach can be reached, when raw features are used, by selecting, using the Kennard-Stone algorithm, 16000 examples from the dataset obtained by merging all the data samples from all the roads included in a single cluster. Baseline performances, using Kalman-filtered features, can be reached by selecting 4000 examples, suggesting that a limited number of samples that are less affected by noise is sufficient to train supervised models when recurring patterns are present and shared among road segments.

This means that we had 4,000 x 5= 20,000 training examples *vs* 6,048 x 321= 1,941,408 of the original dataset, thus significantly reducing the computational complexity.

The limitations of this study are related to the dataset representing the specific situation of the San Francisco urban area, which may exhibit characteristics not found in other cities. The preliminary step of the procedure we followed here, using the autocorrelation operator to check the presence of recurring temporal dynamics in the considered road segments, remains necessary to deploy the approach in other situations. Potential differences may emerge due to different extensions of the considered urban areas or to specific geographical characteristics, as well as to the socio-economical background of the considered city, which may cause non-periodic recurrences that would not be detected through autocorrelation. Also, the temporal extension of the data available through the SFPark project is relatively limited and does not allow us to take into account possible changes due to seasonal variations through the whole year. Future work will, therefore, consist of re-applying the procedure to datasets collected from different cities and covering longer time periods, in order to evaluate, for example, if new clusters and/or new models should be trained to cover different times of the year, how long these time spans should be, and for how long a recurring pattern is present in the series.

We believe that the results of this work can be exploited for further replications/evolutions of the proposed pipeline. Indeed, as future work, we foresee the possibility that the obtained results can be improved by employing more advanced machine learning techniques, like CNN or LSTM on top of the proposed pipeline. Moreover, it would be interesting to replicate the experiment on other parking availability datasets, to understand if and how these recurrent patterns are common in other urban areas.

Acknowledgments: We gratefully thank Yuri Attanasio for the help in the conduction of the experiments.

References

1. Shoup, D. Cruising for parking. *Transport Policy* **2006**, *13*, 479–486. doi:10.1016/j.tranpol.2006.05.005.
2. Lin, T.; Rivano, H.; Mouël, F.L. A Survey of Smart Parking Solutions. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *PP*, 1–25. doi:10.1109/TITS.2017.2685143.
3. Bush, K.; Chavis, C. SAFETY ANALYSIS OF ON-STREET PARKING ON AN URBAN PRINCIPAL ARTERIAL 2. Technical report, 2017.
4. Xu, B.; Wolfson, O.; Yang, J.; Stenneth, L.; Yu, P.S.; Nelson, P.C. Real-time street parking availability estimation. Mobile Data Management (MDM), 2013 IEEE 14th International Conference on. IEEE, 2013, Vol. 1, pp. 16–25.
5. Ma, S.; Wolfson, O.; Xu, B. UPDetector: sensing parking/unparking activities using smartphones. Proceedings of the 7th ACM SIGSPATIAL International Workshop on Computational Transportation Science. ACM, 2014, pp. 76–85.
6. Mathur, S.; Kaul, S.; Gruteser, M.; Trappe, W. ParkNet: a mobile sensor network for harvesting real time vehicular parking information. Proc. 2009 MobiHoc S 3 Workshop, 2009, pp. 25–28.
7. Bock, F.; Di Martino, S.; Sester, M. What Are the Potentialities of Crowdsourcing for Dynamic Maps of On-street Parking Spaces? Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science; ACM: New York, NY, USA, 2016; IWCTS '16, pp. 19–24. doi:10.1145/3003965.3003973.
8. Bock, F.; Di Martino, S. How many probe vehicles do we need to collect on-street parking information? Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on. IEEE, 2017, pp. 538–543.
9. Monteiro, F.V.; Ioannou, P. On-Street Parking Prediction Using Real-Time Data. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 2478–2483.
10. Rajabioun, T.; Ioannou, P. On-street and off-street parking availability prediction using multivariate spatiotemporal models. *Intelligent Transportation Systems, IEEE Transactions on* **2015**, *16*, 2913–2924.

11. Awan, F.M.; Saleem, Y.; Minerva, R.; Crespi, N. A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction. *Sensors* **2020**, *20*, 322.
12. Bock, F.; Di Martino, S.; Origlia, A. A 2-Step Approach to Improve Data-driven Parking Availability Predictions. Procs of the 10th ACM SIGSPATIAL International Workshop on Computational Transportation Science; ACM: New York, NY, USA, 2017; IWCTS '17.
13. Zheng, Y.; Rajasegarar, S.; Leckie, C. Parking availability prediction for sensor-enabled car parks in smart cities. Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on. IEEE, 2015, pp. 1–6.
14. Richter, F.; Di Martino, S.; Mattfeld, D.C. Temporal and spatial clustering for a parking prediction service. Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on. IEEE, 2014, pp. 278–282.
15. Kennard, R.W.; Stone, L.A. Computer aided design of experiments. *Technometrics* **1969**, *11*, 137–148.
16. SFMTA. SFPark: Parking sensor technology performance evaluation, 2014. [Online; accessed 22-January-2019].
17. Grassi, G.; Bahl, P.; Jamieson, K.; Pau, G. ParkMaster: An in–vehicle, edge–based video analytics service for detecting open parking spaces in urban environments. The Second ACM/IEEE Symposium on Edge Computing (SEC 2017), 2017.
18. Mathur, S.; Jin, T.; Kasturirangan, N.; Chandrasekaran, J.; Xue, W.; Gruteser, M.; Trappe, W. Parknet: drive-by sensing of road-side parking statistics. Procs of the 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 123–136.
19. Kalman, R.E.; others. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* **1960**, *82*, 35–45.
20. SFMTA. SFPark Project Analysis. http://sfpark.org/, 2014. [Online; accessed 27-June-2016].
21. Bock, F.; Di Martino, S. On-street Parking Availaibilty Data in San Francisco, from Stationary Sensors and High-Mileage Probe Vehicles. *Data in Brief* **2019**, *25*, 104039.
22. Cherian, J.; Luo, J.; Guo, H.; Ho, S.S.; Wisbrun, R. ParkGauge: Gauging the Occupancy of Parking Garages with Crowdsensed Parking Characteristics. Mobile Data Management (MDM), 2016 17th IEEE International Conference on. IEEE, 2016, Vol. 1, pp. 92–101.
23. Camero, A.; Toutouh, J.; Stolfi, D.H.; Alba, E. Evolutionary deep learning for car park occupancy prediction in smart cities. International Conference on Learning and Intelligent Optimization. Springer, 2018, pp. 386–401.
24. Stolfi, D.H.; Alba, E.; Yao, X. Predicting car park occupancy rates in smart cities. International Conference on Smart Cities. Springer, 2017, pp. 107–117.
25. Jossé, G.; Schubert, M.; Kriegel, H.P. Probabilistic parking queries using aging functions. Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2013, pp. 452–455.
26. Caicedo, F.; Blazquez, C.; Miranda, P. Prediction of parking space availability in real time. *Expert Systems with Applications* **2012**, *39*, 7281–7290.
27. SFMTA. SFpark: Putting Theory Into Practice. Pilot project summary and lessons learned. http://sfpark.org/resources/docspilotsummary/, 2014. [Online; accessed 24-June-2016].
28. Fabian, T. An Algorithm for Parking Lot Occupation Detection. *2008 7th Computer Information Systems and Industrial Management Applications* **2008**, pp. 165–170. doi:10.1109/CISIM.2008.53.
29. Kotb, A.O.; Shen, Y.; Huang, Y. Smart Parking Guidance, Monitoring and Reservations: A Review. *IEEE Intelligent Transportation Systems Magazine* **2017**, *9*, 6–16.
30. Ganti, R.K.; Ye, F.; Lei, H. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* **2011**, *49*, 32–39. doi:10.1109/MCOM.2011.6069707.
31. Rinne, M.; Törmä, S.; Kratinov, D. Mobile crowdsensing of parking space using geofencing and activity recognition. 10th ITS European Congress, Helsinki, Finland, 2014, pp. 16–19.
32. Wu, J.; Feng, Y.; Sun, P. Sensor fusion for recognition of activities of daily living. *Sensors* **2018**, *18*, 4029.
33. Satonaka, H.; Okuda, M.; Hayasaka, S.; Endo, T.; Tanaka, Y.; Yoshida, T. Development of parking space detection using an ultrasonic sensor. Proceedings of the 13th ITS World Congress, 2006.

34. Houben, S.; Komar, M.; Hohm, A.; Luke, S.; Neuhausen, M.; Schlipsing, M. On-vehicle video-based parking lot recognition with fisheye optics. Proc. IEEE Int. Conf. on Intelligent Transportation Systems, 2013, pp. 7–12. doi:10.1109/ITSC.2013.6728595.
35. Liu, K.S.; Gao, J.; Wu, X.; Lin, S. On-Street Parking Guidance with Real-Time Sensing Data for Smart Cities. 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2018, pp. 1–9.
36. Origlia, A.; Di Martino, S.; Attanasio, Y. On-Line Filtering of On-Street Parking Data to Improve Availability Predictions. 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, 2019, pp. 1–7.
37. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies* **2014**, *43*, 3–19.
38. Hou, Z.; Li, X. Repeatability and similarity of freeway traffic flow and long-term prediction under big data. *IEEE Transactions on Intelligent Transportation Systems* **2016**, *17*, 1786–1796.
39. Gutiérrez, V.; Galache, J.A.; Sánchez, L.; Muñoz, L.; Hernández-Muñoz, J.M.; Fernandes, J.; Presser, M. SmartSantander: Internet of things research and innovation through citizen participation. The future internet assembly. Springer, 2013, pp. 173–186.
40. Murtagh, F.; Legendre, P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of classification* **2014**, *31*, 274–295.
41. Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*; Springer Science & Business Media, 2009.
42. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* **2011**, *2*, 1–27.