

Contents lists available at ScienceDirect



Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Improving parking availability prediction in smart cities with IoT and ensemble-based model

Stéphane Cédric Koumetio Tekouabou^a, El Arbi Abdellaoui Alaoui^{b,c,*}, Walid Cherif^d, Hassan Silkan^a^a Department of Computer Science, Laboratory LAROSERI, Faculty of Sciences, El Jadida, Morocco^b EIGSI, 282 Route of the Oasis, Mâarif, 20140 Casablanca, Morocco^c E3MI Research Team, Department of Computer Science, Faculty of Sciences and Techniques at Errachidia, University of Moulay Ismail, Route Meknes, 52000 Errachidia, Morocco^d Laboratory SI2M, National Institute of Statistics and Applied Economics, Rabat, Morocco

ARTICLE INFO

Article history:

Received 23 September 2019

Revised 10 January 2020

Accepted 17 January 2020

Available online xxx

Keywords:

Smart cities

Parking availability

IoT

Regression

Ensemble models

ABSTRACT

Smart cities are part of the ongoing advances in technology to provide a better life quality to its inhabitants. Urban mobility is one of the most important components of smart cities. Due to the growing number of vehicles in these cities, urban traffic congestion is becoming more common. In addition, finding places to park even in car parks is not easy for drivers who run in circles. Studies have shown that drivers looking for parking spaces contribute up to 30% to traffic congestion. In this context, it is necessary to predict the spaces available to drivers in parking lots where they want to park. We propose in this paper a new system that integrates the IoT and a predictive model based on ensemble methods to optimize the prediction of the availability of parking spaces in smart parking. The tests that we carried out on the Birmingham parking data set allowed to reach a Mean Absolute Error (MAE) of 0.06% on average with the algorithm of Bagging Regression (BR). This results have thus improved the best existing performance by over 6.6% while dramatically reducing system complexity.

© 2020 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The continuous growth of the urban population favored by the massive rural exodus pushed cities towards the optimization of their urban resources. In this regard, the will of cities actors and the progress in information and communication technologies (ITC) gave birth to the “smart cities” (Bélissent, 2010). The advent of smart cities is a growing global trend. They aim to integrate ICT solutions to improve the quality of life of its citizens and their interaction with government officials. Thus, traffic and urban mobility are one of the major problems of urban development. They face many challenges of sustainable mobility in face of increasing demand of parking spaces especially those related to the limitation of capacity of the city's transport, traffic and parking systems. One of the typical smart city illustrations is the use of public transport applications and the provision of custom information routines to users. For the design of these applications (usually with support for mobile devices), valuable information must be provided by users to optimize their movement. At the same time,

transport companies are forced to improve the quality of services provided to meet the challenges of smart urban mobility.

The urban mobility in smart cities highlights several issues that are anchored on sustainable development, which aims to make them more attractive, more ecological and more economical while strengthening the social link. Currently, some smart tools help drivers by reporting phenomena such as traffic jams, accidents or even road conditions (Tang and Gao, 2005; De Fabritiis et al., 2008). The rise in urban mobility is accentuated by the increasing number of journeys in personal means compared to the often precarious public transport. This growth in urban mobility is causing a huge number of vehicles that make traffic and even parking more tedious in the face of these sustainable development issues (Jin et al., 2014). This directly affects driver activity by the wasted time they spent to find available parking spaces, the disruption of urban traffic flow and the increased pollution in cities. According to (Zheng et al., 2015), 30% of traffic jams are caused by vehicles in search of parking. In this context, knowing in advance the available parking spaces would overcome this problem. Machine learning techniques would be the best tools for predicting this availability with great precision (Zantali et al., 2019; Camero et al., 2018).

Different machine learning applications have been proposed in the literature to predict parking spaces, these methods vary in the types of collected data and also in the used methods to analyze

* Corresponding author at: EIGSI, 282 Route of the Oasis, Mâarif, 20140 Casablanca, Morocco.

E-mail address: abdellaoui.e@gmail.com (E.A. Abdellaoui Alaoui).

<https://doi.org/10.1016/j.jksuci.2020.01.008>

1319–1578/© 2020 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

these data. Some works used as data the captured images in real time (Xiang et al., 2017) comparable to simple sensor systems. These images have always been processed in real time by algorithms such as deep learning (Amato, 2017; Almeida et al., 2015; Bachani et al., 2016; Amato et al., 2016) to give users real-time availability. These models based on real time already have several drawbacks namely: images are more complex to analyze. Our system overcomes this problem by introducing a data centric-IoTs for data collection, analysis and processing. Our predictive data analysis system must be strengthened from time to time by the most recent data to adjust and update the prediction. The authors (Stolfi et al., 2017; Camero et al., 2018) had to deal with this type of data by using the very complex algorithms often specialized in image processing (Convolutional Neural Network “CNN”). In short even if performance is improved, they remain to be optimized not only in terms of complexity but also in terms of performance and execution time. In this paper, we introduce a model based on ensemble methods which is able to overcome the above mentioned problem. Basically in this work, we propose a data-centric approach that aims to predict parking space availability for a given city parking. The proposed system uses machine learning techniques and also integrates different connected sources of information (IoT). The main sources of data used for the test of our model comes from car parks in Birmingham city. For the predictions, different regression techniques were used to predict a parking lot availability for a given time. Interesting results were obtained with predictors based on different regression techniques.

1.1. Research contributions

The key contributions of this work are:

- The integration of an IoT-based system in smart cities and particularly in the case of smart parking;
- The global system uses ensemble model to predict the availability of places in a smart parking;
- The optimization of different ensemble-based models for predicting availability rate in smart parking;

The rest of this paper is organized as follows: Section 2 explains the related work. Section 3 will present IoT and predictive analysis for smart cities challenges. Section 4 details the proposed methodology. In Section 5, the obtained results are analysed and discussed. Finally, Section 6 concludes this work.

2. Related work

This section is devoted to the literature review and aims to clarify through the readings, the main terms, concepts and currents of thought related to the field of parking. Parking is a very large subject; for this purpose, this section focuses on the elements related to the specific objectives of this paper.

In Seong-eun et al. (2008), the parking management system is broken down into two subsystems. A first vehicle detection system (VDS) and a second vehicle management system (VMS). The VDS detects the status of the parking spaces and sends the collected information to the VMS subsystem to provide to drivers. Another intelligent parking management system is proposed in Kumar et al. (2007). The authors of Kumar et al. (2007) compare the use of different types of sensors (acoustics, light sensors and magnetic) for parking management. Information from different types of sensors is sent to a central server in a finite time. Some authors such as (Gandhi and Rao, 2016) developed prototype using sensor circuit, RFID and IoT to detect the car details and then used IR sensor to find the presence of the car so that all details are accessed from

remotely through IoT. The RFID-based technique has also been used by Pawowicz et al. (2019) to improve traffic control management in smart city but the problem of prediction will still remains with this technology. In Giuffrè et al. (2012), the authors proposed a smart parking system, called Intelligent Parking Assistant (IPA). Their system relies mainly on the use of sensor networks in intelligent parking management. However, this work does not take into account machine learning algorithms and the advantages of IoT.

Alkheder et al. (2016) introduced a smart parking system for shopping malls in the city Abu Dhabi, especially on weekends, holidays and even weekdays. In this article (Rajabioun and Ioannou, 2015), the authors have developed a self-regressive model space-time vector that can be used to predict the evolution of the availability of parking for on-street and off-street parking at the driver's expected arrival time. The project takes into account temporal correlations in parking availability as well as spatial correlations. It is used to recommend the parking location having the highest probability. They used real-time parking data from the San Francisco area to evaluate the results and verify the model. Bachani et al. (2016) presented a comprehensive analysis of the crucial aspects of the design of an intelligent parking system, namely the selection of sensors and the optimal position of their deployment for accurate detection.

Amato (2017) proposed an intelligent parking system based on computing vision, they used deep learning to determine the availability of spaces in a parking lot. Their system is compared with two methods that exist in the literature PKLot (Almeida et al., 2015) and CNRPark-EXT (Amato et al., 2016). Xiang et al. (2017) proposed an approach for detecting real-time parking occupancy at gas stations using Haar-AdaBoosting and CNN algorithms. Another case study of deep learning concerns (Shoeibi and Shoeibi, 2019) which introduced an automated valet parking based on hybrid robotic valets in smart parking and helps to optimize the usage of parking space with Deep Q-Learning which as a reinforcement learning method. The Paper (Mago and Kumar, 2018) proposed a model for design of an optimized parking management system based on advanced video processing techniques which will be useful for assigning vehicles to the outdoor parking number available at the entry point. His method allows a real-time dictation which is far from the objective of predictive analysis which aims to anticipate real time. In Camero et al. (2018), the authors presented a new technique based on deep learning with recurrent neural networks to process the parking occupancy rate forecast. The same authors in Stolfi et al. (2017) proposed a study of parking occupancy data in order to test several forecasting strategies such as: polynomial adjustment, fourier series, k -means grouping and time series. The models built by this different authors are still critical, don't take into account emerging technologies such as IoT and could be the subject of improvement; which led to rest of this work.

In this paper we propose a parking management architecture where we propose a new method for predicting available parking spaces using a combination between the trio of smart city, IoT and machine learning which are neglected parking management architectures proposed in the above works.

3. IoT and predictive analysis for providing the smart cities challenges

3.1. Smart cities challenges

The concept of smart city is quite new and represents a new approach to urban development. This one promotes the integration of new information and communications technology (ICT) in the management of cities to effectively meet the needs of citizens.

The idea of the intelligence of a city is therefore to improve the quality of services and urban interactivity while minimizing costs and resources. Smart cities, for example, help to improve relations between citizens, service providers and governing authorities. Generally, a smart city relies on the following three streams as illustrated in the Fig. 1:

- **Logistics flow:** Logistics flow in a “smart city” corresponds to urban traffic systems(UTS).
- **Energy flows:** In urban areas, they correspond to all energy transfers from the production sources to the distributors and then to the use entities (vehicles, homes, public lighting, charging stations, etc.).
- **Data flow:** With the arrival of new technologies in cities (smart phones, sensor networks, vehicles, demotic and building systems, etc), a lot of data from different applications is stored and can be transformed into knowledge.

A smart city can be improved with the mechanisms of IoT and intelligent transport system (ITS). Indeed, the concepts of IoT and ITS can offer valuable real-time information to smart city players. For example, in Fig. 2, we illustrate the various services provided by ITS in a smart city. Among these challenges, many are related to the problem of smart parking and thus challenge the Internet of Things and predictive analysis.

3.2. IoT and predictive analysis for smart parking challenges

The Internet of Things (IoT) is speeding up the pace of innovation in the transport sector, particularly with regard to smart parking and urban mobility. Today, many smart car parks are equipped with connected systems that allow drivers to view their smart-phone apps, set directions, use roadside assistance, open doors remotely, and locate free parking spaces. Thus, the Internet of Things will also bring to the automotive sector novelties of which we have not yet the slightest idea. However, any new technology involves new challenges including: technology complexity, security, privacy, data management and analysis (Arasteh et al., 2016; Mainetti et al., 2015; Kubler et al., 2016).

Are we not already drowned in the data? the IoT will produce even more data, adding additional complexity to our enterprise information management systems. Indeed, fully functional IoTs such as intelligent parking systems can generate a large amount of data. These so-called data-centric IoTs have focused on all aspects of data flow, including collection, processing, storage and visualization (Jin et al., 2014).

The massive volume of data poses challenges associated with the collection of these data, their processing, storage, management

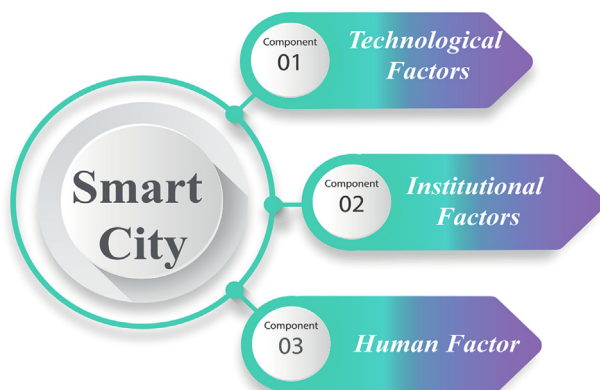


Fig. 1. Components of smart city.

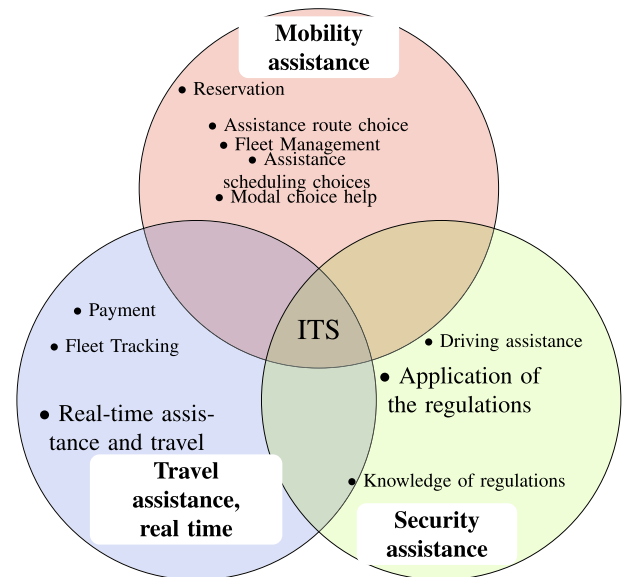


Fig. 2. Services types provided by ITS.

and manipulation. Advanced scanning technologies will be needed to provide relevant information from the data generated by the connected instruments. It will also open up new opportunities to optimize business processes, offering new features for e-government, supply chain and urban transport management. Data capture and analysis will be maximally effective if capture, analysis, and delivery is done from a cloud-based system. It is the case of the automatic management system of intelligent parking. In the next section, we will present our ensemble-based model to optimize the prediction of space availability in smart parking (Arasteh et al., 2016; Mainetti et al., 2015).

4. Methodology

4.1. System model of smart parking

The intelligent parking system is an intelligent parking system that uses a detection device to define the occupancy rate of the parking space. It helps the driver to park safely and informs him/her of the availability of parking spaces through appropriate vehicle management. Thanks to intelligent technologies, optimized parking can reach the city centre. A sensor system indicating to drivers where the nearest free parking space is located has already been successfully tested in multi-storey car parks. Street tests are currently underway. In San Francisco, CA, 6000 sensors have been embedded in the asphalt and are working in conjunction with an application and a GPS (Lin et al., 2017; Rodier and Shaheen, 2010). An advanced smart parking system architecture needs to have the following elements:

- Sensor,
 - Gateway hardware;
 - Server/Cloud;
 - Mobile application.
- as shown in the Fig. 3.

4.2. About ensemble-based models

Ensemble-based prediction methods combine several independent basic models that are in most cases decision trees or neural networks. Each of these basic models provides an alternative

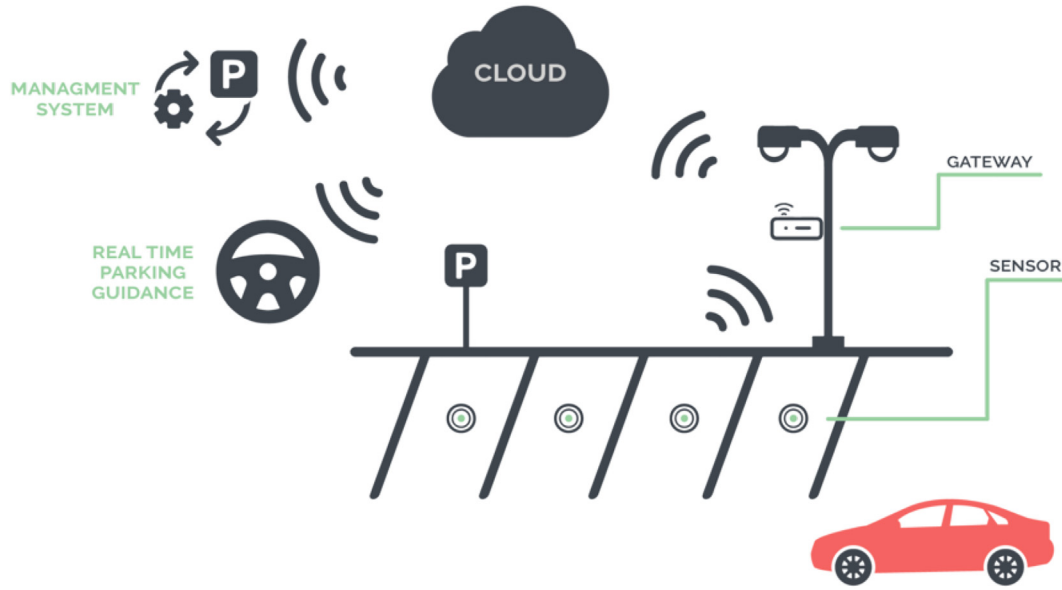


Fig. 3. Smart parking system.

prediction of the problem and the final prediction is a combination (usually by weighted or unweighted vote) of alternative predictions.

The prediction technique by combining the predictions of a set of individual base models generally allows for more stable and accurate output prediction because the error is much smaller than that provided by one of the individual base models which form the overall model. Indeed, the final ensemble-based model corrects the errors made individually by the basic models to drastically reduce the total error. To be effective, the basic models should be forced to fulfill two conditions namely to be independent and to be weak models.

The initial idea was to divide the training data D into n basic data to train n models m_1, m_2, \dots, m_n . But this technique was quickly exceeded because it promotes under-fitting when n becomes high. To overcome this limit, there are methods of re-sampling the training data into n independent and larger data sub-samples to generate weak models. To do this, various techniques among which the most used known are: bagging and boosting.

Thus, the most favorable algorithms for these conditions are unstable algorithms such as decision trees and neural networks, a slight modification of which in the data set makes it possible to obtain a different model. The Fig. 5 illustrates the flow chart of a prediction system from a set-based model. In the rest of this part we will present our basic model which are the decision trees, then the two techniques of sampling (bagging and boosting) and finally our three models of set namely Random Forest regressor, Gradient boosting regressor and Adaboost regressors.

4.3. KNN for regression

KNN is a lazy learning algorithm that uses a non-parametric method to solve a classification or regression problem (Pow et al., 2014). It assigns the class or regression value by averaging the nearest k neighbor values, for numeric instances, or applying the majority vote for k neighbors, if the instance values are categorical. After selecting the k nearest neighbors, the value can be predicted either by averaging the k neighbor outputs (uniform weighting) or a weighted sum defined by a function (Sinta et al., 2014; Imandoust and Bolandraftar, 2013).

4.4. Decision trees

The prediction tree technique relies on the use of the tree to represent the recursive partition of the initial total space. Each of the terminal nodes, or leaves, of the tree represents a cell of the partition and is associated with a simple model that applies only to that cell. To better understand, consider a regression problem with a continuous output variable Y and two independent variables X_1 and X_2 . We are inspired by the fact that we have to divide to better reign, to first divide our space into two regions and model the Y response (mean of Y) individually in each region. Then we continue to split each individual region into two more regions and continue the process until a stop rule is met (Faris et al., 2019). To generalize the previous case: consider a regression problem consisting of p entries with a single output variable. For example, we have k instances, each instance being composed of $(y_i, x_{i1}, x_{i2}, \dots, x_{ip})$ for $i = 1, 2, \dots, k; j = 1, 2, \dots, p$. In terms of predicting the availability rate of places in a smart car park, y_i can be the availability rate at parking i , $(x_{i1}, x_{i2}, \dots, x_{ip})$ are relevant variables for the rate time of predicted availability, such as time, date, capacity, identifier, or other factors. The characteristic space is partitioned into M regions E_1, E_2, \dots, E_m . In other words, the regression tree divides the traffic conditions into categories based on the value of the input parameters and models the response (dependent variable) individually for each category. The answer for each region is often treated as a C_m constant.

$$\text{minimize}(SSE) = \sum_{k=1}^{E_m} \sum_{i=1}^n (y_i - C_k)^2 \quad (1)$$

$$\text{minimize } SSE + \alpha|T| \quad (2)$$

Since the optimization criterion consists in minimizing the sum of squares, the best C_m is only the mean of y_i in the R_m region (Zhang and Haghani, 2015). To find the optimal region we use a cost complexity parameter (α) which penalizes our objective function in Eq. (1) for the number of terminal nodes of the tree (T) as in Eq. (2). To choose the best division variable and the dividing point, a greedy algorithm was implemented. For each division variable, the best division point can be determined by scanning all possible values, this should be done very quickly. By browsing all the input

variables, it is possible to find the best pair of split variable and split points. A single regression tree is the basic model for the powerful Random Forest and Gradient Boosting Regression trees methods.

4.5. Ensemble-based models for regression

The general idea of the whole is summarized in Fig. 4 which shows that these models are based on three main stages namely: bootstrapping, intermediate modeling and aggregation. The bootstrapping consists in dividing all data D into n data D_1, D_2, \dots, D_n . From each data set D_i we will construct an intermediary regressor R_i and the final regressor will be an aggregation of the intermediary regressors R_i . From this general idea will be born several methods among which the most powerful are the bagging used in the Random Forest algorithm and the Boosting used by Gradient Boosting and Adaptive Boosting.

4.6. Bagging method

4.6.1. Bagging Regression (BR)

The term bagging comes from the contraction of Bootstrap Aggregating. We present this family of methods in a regression context, but they also extend very easily to supervised classification. We denote by (X, Y) a random vector representing the learning data where X takes its values in R_p and Y in R . We denote $D_n = (X_1, Y_1), \dots, (X_n, Y_n)$ a sample independent and equally distributed, and with the same law that (X, Y) and $\hat{m}(x) = E[Y|X = x]$ the regression function. For $x \in R_p$, we consider the mean squared error of an estimator m and its bias-variance decomposition as follows:

$$(\hat{m}(x) - m(x))^2 = (E\hat{m}(x) - m(x))^2 + V(\hat{m}(x)) \quad (3)$$

They consist in aggregating a number B of models $\hat{m}_1, \hat{m}_2, \dots, \hat{m}_B$ such that:

$$\hat{m}(x) = \frac{1}{B} \sum_{k=1}^B (\hat{m}_k(x)) \quad (4)$$

and we have:

$$E\hat{m}(x) = Em_1(x) \quad (5)$$

and

$$V(\hat{m}(x)) = \frac{1}{B} V(m_1(x)) \quad (6)$$

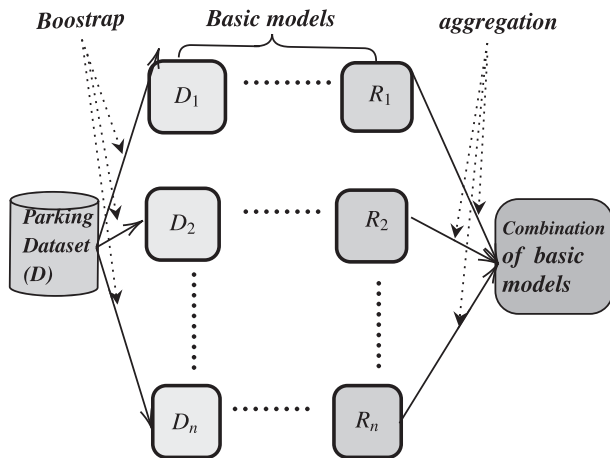


Fig. 4. Ensemble-based Model.

The bias of the aggregate model is therefore the same as that of the \hat{m}_k but the variance decreases. Of course, in practice it is almost impossible to consider independent models \hat{m}_k insofar as they all depend on the same sample D_n . The bagging approach is to try to mitigate the dependency between the models that are aggregated by building them on bootstrap samples.

Algorithm 1 Bagging Regressor (BR)

Input:

- x the observation to predict;
- d_n the observation;
- B the number of Trees;
- $m \in N$ the number of candidate variables to cut a node.

Output: the estimator $\hat{m}(x) = \frac{1}{B} \sum_{k=1}^B \hat{m}_k(x)$

1: for $k = 1$ to B do

2: Draw a bootstrap sample d_n^k in d_n

3: Adjust the regressor on this bootstrap sample $\hat{m}(x)$.

4: end for

4.6.2. Random forest regression (RFR)

Random forest is nothing more than a particular bagging method consisting of an aggregation of trees based on random variables. Most often, trees are built with the classification and regression tree (CART) algorithm whose principle is to recursively partition the space generated by the explanatory variables in a dyadic way. More precisely, at each stage of the partitioning, a part of the space is cut into two sub-parts according to a variable X_j (Zhang and Haghani, 2015).

Algorithm 2 Random Forest Regressor (RFR)

Input:

- x the observation to predict;
- d_n the observation;
- B the number of Trees;
- $m \in N$ the number of candidate variables to cut a node.

Output: $h(x) = \frac{1}{B} \sum_{k=1}^B h(x, y)$

1: for $k = 1$ to B do

2: Draw a bootstrap sample in d_n

3: Construct a CART tree on this bootstrap sample, each cutoff is selected by minimizing the cost function of CART over a set of m randomly selected variables among the p . We note $h(., k)$ the built tree.

4: end for

4.7. Boosting method

In the state of the art of machine learning, boosting is proving to be one of the most effective ideas of the past three decades and continues to be the subject of abundant literature (Freund et al., 1999; Freund and Schapire, 1996). As the name implies, the general principle of boosting is to build a family of models that are then aggregated by a weighted average of estimates (in regression) or a majority vote (in classification). Unlike the above bagging method, in boosting the classical increase in subset creation is not random. The general idea of most boosting methods is to train predictors sequentially, each trying to correct its predecessor (Géron, 2017). The performance depends on the performance of previous models as each new subset is iterated on the previous contained elements that could have been misclassified by previous models. AdaBoost and Gradient Boosting are among the most popular boosting methods.

4.7.1. Adaptive boosting regression (ABR)

The adaptive boosting method is based on the fact that a new predictor to correct the error of its predecessor simply pays a little more attention to the training instances under which the predecessor has adapted. The result is new predictors that focus more and more on difficult cases. For example, to create an AdaBoost classifier, consider a first classifier that is nothing more than a decision tree. This primitive tree formed is used to make predictions on the set of formations. The weight corresponding to the misclassified training instances is then increased. A second classifier is then formed based on these updated weights (Mishra et al., 2017). The second classifier again makes predictions about the training game. The weights are then updated, and so on. Once all the predictors have been formed, the set makes predictions very similar to bagging or pasting operations. The only difference is that the resulting predictors have different weights based on their overall accuracy over the weighted training set (Géron, 2017). The process of Adaptive Boosting is presented in the Algorithm 3.

Algorithm 3 AdaBoosting Regressor (ABR)

Input:

- x the observation to predict;
- $d_n = (x_1, y_1), \dots, (x_n, y_n)$ the sample;
- the weak rule;
- M the number of iterations;

Output: the estimator $\hat{g}_M(x)$

1: *Initialization:*

$$g_0(\cdot) = \operatorname{argmin}_c f(x) = \frac{1}{n} \sum_{m=1}^M L(y_i, c)$$

2: **for** $m = 1$ to M **do**

3: Calculate the opposite of the gradient $\frac{\partial}{\partial y} L(y, g)$ - and evaluate it at points $g_{m-1}(x_i)$:

$$U_i = - \left[\frac{\partial L(y_i, g_m(x_i))}{\partial g(x_i)} \right]_{g(x_i)=g_{m-1}(x_i)}, i = 1, \dots, n.$$

4: Adjust the weak rule on the sample $(x_1, U_1), \dots, (x_n, U_n)$, we notice h_m the rule thus defined.

5: Update: $g_m(x) = g_{m-1}(x) + \lambda h_m(x)$

6: **end for**

4.7.2. Gradient Boosting Regression (GBR)

Another very popular reinforcement algorithm is gradient enhancement. Gradient Boosting works similarly to AdaBoost by sequentially adding predictors to a set, so that everyone tries to correct the errors of its predecessor. However, instead of adjusting the instance weights at each iteration, as AdaBoost does, this method tries to fit the new predictor to the residual errors committed by the previous one (Géron, 2017).

Algorithm 4 Gradient Boosting Regression (GBR)

Input:

- x the observation to predict;
- h a weak rule,
- $d_n = (x_1, y_1), \dots, (x_n, y_n)$ the sample;
- λ a regularization parameter such as $0 < \lambda < 1$;
- M the number of iterations;

Output: the estimator $\hat{g}_M(x) = \sum_{m=1}^M \alpha_m g_m(x)$

1: *Initialization:* Initialize the weight distribution of training data by $w_i = \frac{1}{N}, i = 1, 2, \dots, N$

Algorithm 4 (continued)

2: **for** $m = 1$ to M **do**

3: Adjust the weak rule on the sample d_n weighted by the weights w_1, \dots, w_n , we note $g_m(x)$ the estimator resulting from this adjustment.

4: Compute $e_m = \frac{\sum_{i=1}^N w_i * 1_{y_i \neq g_m(x)}}{\sum_{i=1}^N n w_i}$

5: Compute $\alpha_m = \log(\frac{1-e_m}{e_m})$.

6: Readjust the weights:

$$w_i = w_i * \exp(w_i * 1_{y_i \neq g_m(x)}), i = 1, \dots, N$$

7: **end for**

5. Experimental results and discussion

In this subsection, we will detail the experiments of the process of parking availability prediction based on ensemble method.

5.1. Dataset

As shown in Fig. 5, our global predictive system consists of several phases. The first step consists of collecting data from the sensors installed in the different smart parkings. At this level, the data is collected in parking database as a csv file. The data analyzed in this paper comes from the Birmingham car park and was first used in Camero et al. (2018) and Stolfi et al. (2017) comprising valid occupancy rates of 29 car parks operated by NCP (National Car Parks) in the city of Birmingham in the U.K. Birmingham, which is a major city in the West Midlands of England, standing on the small River Rea. It is the largest and most populous British city outside London, with an estimated population of 1,124,569 as of 2016 (Camero et al., 2018). Several cities in the U.K. have been publishing their open data to be used, not only by researchers and companies, but also for citizens to better know the place where they live. The Birmingham data set is licensed under the Open Government License v3.0 and it is updated every 15 min from 8:00 AM to 4:30 PM (18 occupancy values per car park and day). In our study, we worked with data collected from Oct 4, 2016 to Dec 19, 2016 (11 weeks) which is available on UCI machine Learning Repository (Camero et al., 2018; Stolfi et al., 2017).

The selection of relevant data consists in eliminating irrelevant and redundant information (Khurana and Saxena, 2018; Sharma and Mir, 2019). For the Birmingham Parking Database, the features considered relevant to the problem are: *SystemCodeNumber* is an alphanumeric code that identifies a car park. *LastUpdated*: contains the date and time of the last update for occupancy data for each parking block. Schedules are recorded between 8:30 am and 6:30 pm. *Capacity*: contains the capabilities of each car park. *Occupancy*: contains the occupations of each car park which are updated every 30 min. Other features such as the occupancy rate and the exit rate of each parking lot were not considered in this work.

From these features we have generated a specific feature called the availability rate that we have noted *AVR* which is the ratio of the capacity minus the occupation at time t of the date d , on the capacity of the parking lot. In our case, it is calculated by the following formula:

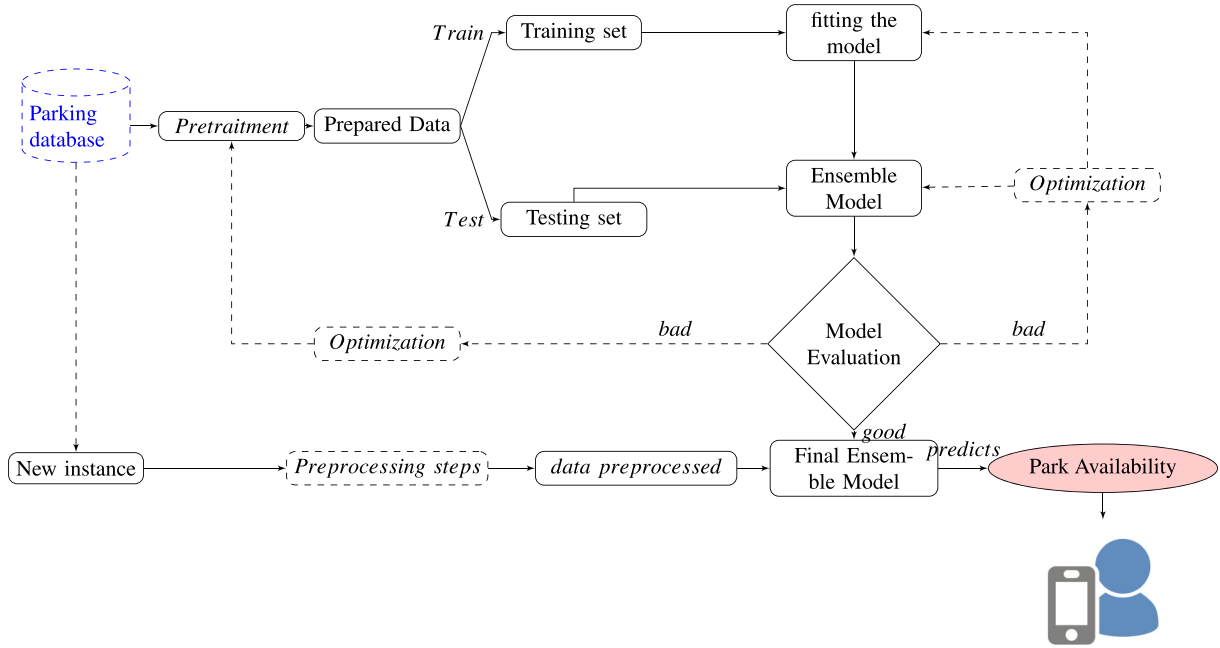


Fig. 5. Global ensemble-based System for Real-time Parking availability Prediction.

$$AVR_p(t) = \frac{Capacity_p - Occupancy_p(t)}{Capacity_p} \quad (7)$$

As an example, car park “BHMBCCMKT01” has a capacity of 577 parking space, the 04/10/2016 at 07:59:42 it has an occupancy of 61, so we can deduce are occupancy rate at this time which is 61/577 and is 10.57%. The availability rate in this car park at this time of this date is 89.43% and shows that the user can easily park in this parking at this time without harming the traffic. Fig. 6 illustrates the distribution of park availability rate according to the day time Fig. 6a, then by day of week Fig. 6b and finally according to each park lot Fig. 6c. Knowing these availability rates in real time, we are now looking to predict future availability rates without knowing occupation using machine learning algorithms. The availability rate of parking spaces is the fifth variable of our problem which will be our label.

5.2. Performance measures

In order to build an optimal approach, we compared the performance of different models using three main measures: the mean absolute error (MAE), the r-square (R^2) and the root mean square error (RMSE). The three terms can judge the difference between

the real and predicted parking availability rates in different aspects. They are calculated as follows: (Xu and Ying, 2017)

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (AVR_{i,p} - AVR_i)^2}{N-1}} \quad (8)$$

$$MAE = \frac{\sum_{i=1}^N |AVR_{i,p} - AVR_i|}{N} \quad (9)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (AVR_{i,p} - AVR_i)^2}{\sum_{i=1}^N (AVR_i - \bar{AVR})^2} \quad (10)$$

where N is the total number of instances, $AVR_{i,p}$ is the predicted availability rate of the instance i and AVR_i is the real availability rate of this instance. The choice of a single measure may not always allow to separate the models. If the RMSE will show the error characterized by the variance and mean between the predicted and the real by favoring the effects of the high deviations, the absolute error may reflect the effect of the precision in the prediction of the waiting time and R^2 will show us the proportion of the actual waiting time that has been correctly predicted. The optimal model will result from the homogeneity between these three measures.

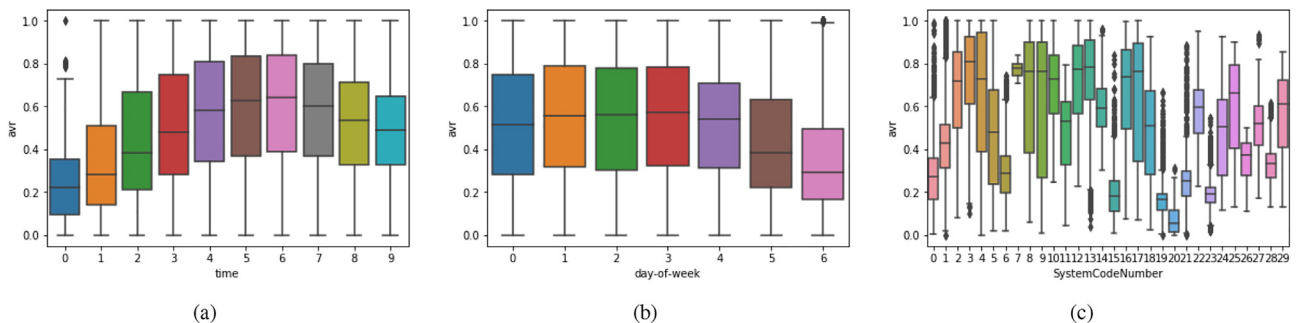


Fig. 6. Distribution of the park availability.

5.3. Optimizing simulation parameters

We want to compare the performances of the different learning methods discussed in the Section 4 to the prediction of the availability rate of parking spaces. We have the database exposed in Section 5.1 part. We divided this data into a training set (80%) to build the models and a testing set (20%) to evaluate the constructed models and then compare their performance. First, we looked for parameters to build stable and optimal models. For bagging methods, the most important are the number of iterations, the out of bag error (E_{OOB}) and the best weak learner. For the boosting method, it was necessary to find the number of iteration, the learning rate and the best weak learner.

The Fig. 7 illustrates the tests that were performed to select these parameters using a simple decision tree as weak learner. Figs. 7a and b provide the optimal parameters for the boosting method. From Fig. 7a, we found that unlike default value learning rate (0.1), the optimal rate is rather around 0.5 for Gradient Boosting. This same rate could be adopted for AdaBoost. Fig. 7b shows that the optimal number of estimators for Gradient Boosting is at least 300, whereas for Adaboost it is not very important (around 40 estimators). Fig. 7c shows that, starting from the estimator, the stable optimal classifier based on the bagging is constructed. The out-of-bag error is almost equal to 0. In Fig. 8, we illustrate the shape of the regression distribution of 200 first instances of the database for the four algorithms GBR, ABR, RFR and BR. From this figure, we see that the distribution of BR then RFR shows that it would be more favorable to better performance unlike GBR and ABR which would be relatively less efficient. Analysis of the results would lighten this trend.

5.4. Results analysis and discussion

We performed two main tests to evaluate the chosen algorithms: first on the entire database of all lots of car parks and second on each parking lot data.

5.4.1. Results analysis for global parking

At first, we tested our approach on the entire database of all lots of car parks. We considered the variable “SysteCodeNumber” as an id that was coded by numbers from 0 to 28 for the 29 lots to predict overall parking availability. The tests carried out consisted of analyzing and comparing the performances of four methods. We illustrated two algorithms which use the bagging optimization method (BR and RFR) and two others which use boosting (ABR and GBR). The results obtained are shown in Table 1 from which we generated Fig. 9 which clearly illustrates these comparative performances. Overall for the three measurements, the bagging methods (BR and RFR) are given the best results compared to the boosting method.

With 0.001823 and 0.000673 for RMSE and MAE respectively, BR gave the optimal performance for these 2 measures. In terms of R^2 , BR and RFR gave the same almost perfect performance reaching 0.999954. these performances are very slightly above RFR with an improvement not exceeding 0.00007 for these two measures.

GBR has given 0.034405, 0.025432 and 0.983772 peer-ratings for MAE, RMSE and R^2 respectively. These values were better than ABR which gave 0.088783, 0.106124 and 0.845595 respectively for these three measures.

Among the two boosting methods tested (GBR and ABR), the results were very different compared to bagging methods where they were similar as shown by the trend of Fig. 7. Compared to bagging methods, the boosting methods were a little less effective for the tests performed. The discrepancies between the optimal

Table 1
Performance comparison.

		RMSE	MAE	R^2
Bagging	RFR	0,001888	0,000682	0,999951
	BR	0,001823	0,000673	0,999954
Boosting	GBR	0,034405	0,025432	0,983772
	ABR	0,106124	0,088783	0,845595

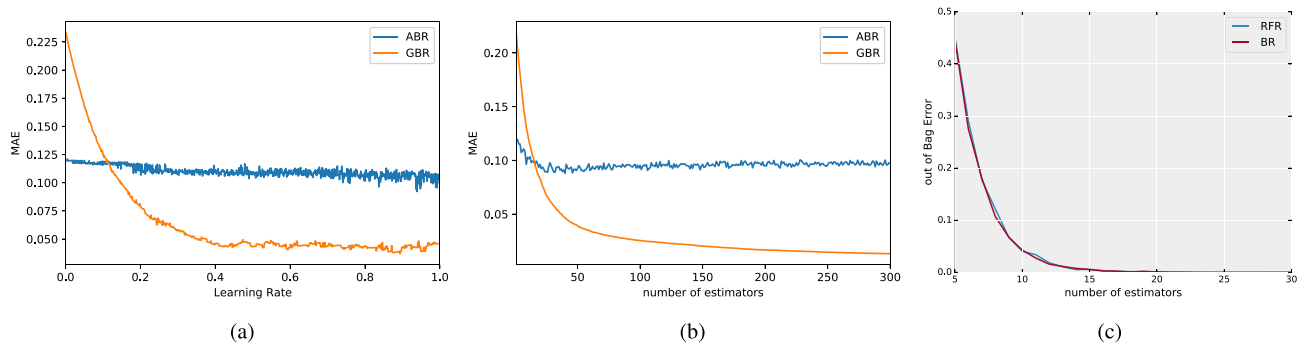


Fig. 7. Optimizing parameters of boosted methods.

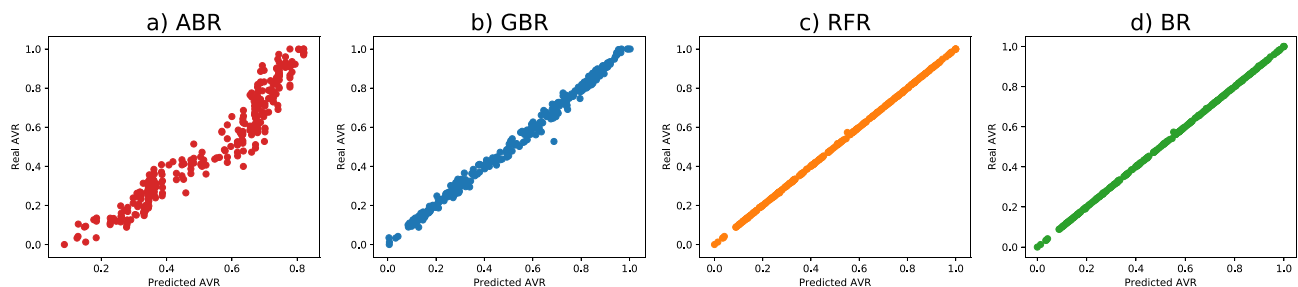


Fig. 8. Comparison of the regression distribution look for 200 first instances by the different algorithms.

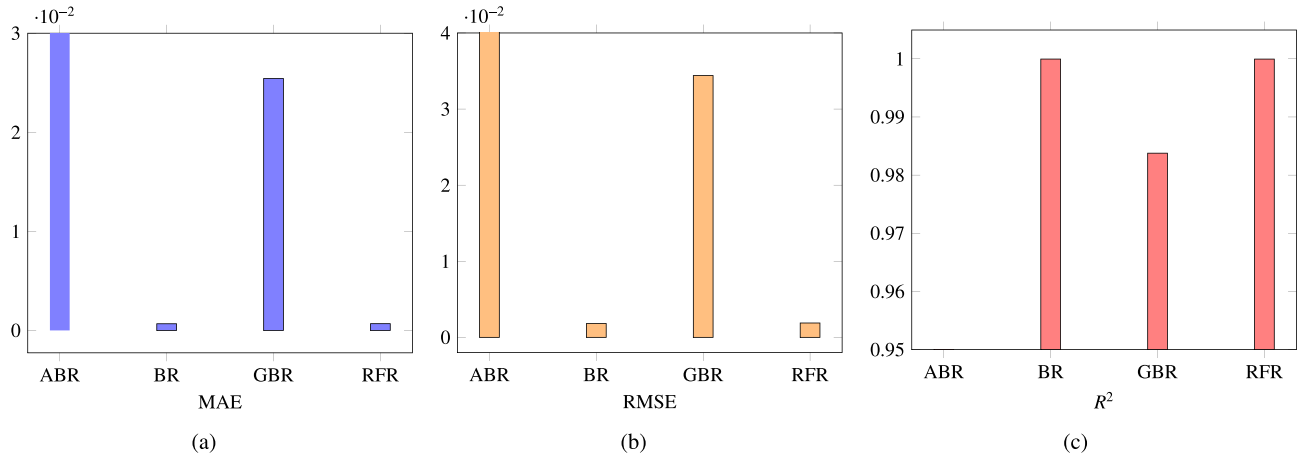


Fig. 9. Comparison of the performances of the used regression algorithms.

method (BR) and GBR were varied between 0.015 and 0.033. While they were between 0.088 and 0.155 between BR and ABR.

5.4.2. Results analysis for each parking lot

We tried our methods to predict the availability on each individual parking lot. Table 2 shows the results obtained for the three loss functions (i.e: MAE, RMSE and R^2) in contrast to previous work where only one loss function was used. In the same table, we compared the average, minimum and maximum performance (i.e: MAE, RMSE and R^2) of each method. In terms of MAE, the RFR method gave the lowest average value 0.00057 compared to the

other methods. For RMSE and R^2 , we found that BR gave the optimum mean values (0.0012 and 0.99981 respectively) compared to the other methods. Still in Table 2, we were interested in the extreme values (Min and Max), then in the standard deviations of each method. We found that BR followed by RFR gave the lowest standard deviations for all measures. Thus, we concluded that optimizing by bagging and especially BR was the most efficient method for predicting the availability of space in each car parks lot.

5.4.3. Comparison with previous works

Given that the main objective of the prediction is to predict values as close as possible to reality, we have compared the perfor-

Table 2
MAE, RMSE and R^2 of the predicted availability in each car park lot.

	MAE				RMSE				R^2			
	BR	ABR	GBR	RFR	BR	ABR	GBR	RFR	BR	ABR	GBR	RFR
BHMBCCMKT01	0,0005	0,0091	0,0014	0,0006	0,0014	0,0121	0,0021	0,0013	0,9999	0,9952	0,9999	0,9999
BHMBCCPST01	0,0006	0,0114	0,0013	0,0005	0,0018	0,0148	0,0020	0,0015	0,9999	0,9946	0,9999	0,9999
BHMBCCSNH01	0,0006	0,0093	0,0014	0,0006	0,0009	0,0123	0,0020	0,0010	0,9999	0,9973	0,9999	0,9999
BHMBCCTHL01	0,0005	0,0101	0,0011	0,0005	0,0013	0,0122	0,0017	0,0012	0,9999	0,9973	0,9999	0,9999
BHMBRCBRG01	0,0007	0,0143	0,0014	0,0007	0,0012	0,0170	0,0020	0,0011	0,9999	0,9971	0,9999	0,9999
BHMBRCBRG02	0,0007	0,0107	0,0016	0,0007	0,0011	0,0138	0,0023	0,0011	0,9999	0,9974	0,9999	0,9999
BHMBRCBRG03	0,0004	0,0061	0,0009	0,0004	0,0009	0,0080	0,0014	0,0010	0,9998	0,9965	0,9999	0,9999
BHMEURBRD01	0,0013	0,0021	0,0010	0,0015	0,0016	0,0030	0,0015	0,0018	0,9971	0,9908	0,9975	0,9964
BHMEURBRD02	0,0005	0,0145	0,0014	0,0005	0,0010	0,0182	0,0020	0,0010	0,9999	0,9963	0,9999	0,9999
BHMMBMMBX01	0,0004	0,0146	0,0010	0,0004	0,0013	0,0195	0,0018	0,0013	0,9999	0,9964	0,9999	0,9999
BHMNCPHST01	0,0004	0,0151	0,0010	0,0004	0,0008	0,0171	0,0015	0,0007	0,9999	0,9902	0,9999	0,9999
BHMNCPLDH01	0,0007	0,0091	0,0011	0,0006	0,0024	0,0124	0,0025	0,0024	0,9998	0,9956	0,9998	0,9998
BHMNCPNHS01	0,0005	0,0081	0,0012	0,0005	0,0008	0,0211	0,0017	0,0008	0,9999	0,9999	0,9999	0,9999
BHMNCPNST01	0,0007	0,0100	0,0014	0,0006	0,0015	0,0128	0,0021	0,0018	0,9999	0,9901	0,9999	0,9999
BHMNCPPLS01	0,0003	0,0060	0,0009	0,0004	0,0014	0,0078	0,0015	0,0014	0,9998	0,9972	0,9998	0,9998
BHMNCPRAN01	0,0003	0,0104	0,0009	0,0003	0,0013	0,0130	0,0017	0,0015	0,9998	0,9957	0,9998	0,9998
Broad Street	0,0007	0,0154	0,0014	0,0007	0,0014	0,191	0,0020	0,0015	0,9999	0,9993	0,9999	0,9999
Bull Ring	0,0005	0,0172	0,0013	0,0005	0,0009	0,0207	0,0018	0,0010	0,9999	0,9946	0,9999	0,9999
NIA Car Parks	0,0007	0,0172	0,0014	0,0006	0,0009	0,0102	0,0019	0,0008	0,9999	0,9954	0,9999	0,9999
NIA North	0,0005	0,0081	0,0009	0,0006	0,0015	0,0109	0,0019	0,0035	0,9997	0,9982	0,9995	0,9985
NIA South	0,0014	0,0084	0,0010	0,0009	0,0022	0,0054	0,0014	0,0024	0,9989	0,9956	0,9996	0,9987
Others-CCCP5105a	0,0005	0,0045	0,0011	0,0005	0,0014	0,0104	0,0018	0,0012	0,9999	0,9940	0,9998	0,9999
Others-CCCP5119a	0,0005	0,0078	0,0010	0,0005	0,0019	0,0069	0,0021	0,0013	0,9998	0,9939	0,9997	0,9999
Others-CCCP5133	0,0004	0,0054	0,0007	0,0004	0,0011	0,0075	0,0013	0,0016	0,9998	0,9973	0,9997	0,9996
Others-CCCP5135a	0,0006	0,0057	0,0012	0,0006	0,0011	0,0127	0,0016	0,0010	0,9999	0,9920	0,9999	0,9999
Others-CCCP5202	0,0005	0,0091	0,0012	0,0005	0,0008	0,0188	0,0017	0,0008	0,9999	0,9958	0,9999	0,9999
Others-CCCP58	0,0003	0,0157	0,0005	0,0003	0,0004	0,0053	0,0007	0,0004	0,9999	0,9933	0,9999	0,9999
Others-CCCP598	0,0004	0,0043	0,0009	0,0004	0,0008	0,0073	0,0014	0,0009	0,9999	0,9973	0,9999	0,9999
Shopping	0,0003	0,0004	0,0059	0,0003	0,0006	0,0047	0,0009	0,0006	0,9999	0,9969	0,9998	0,9999
Mean	0,00058	0,00974	0,00112	0,00057	0,0012	0,0018	0,0017	0,0013	0,99981	0,99478	0,99980	0,99973
Max	0,00144	0,01811	0,00163	0,00152	0,00239	0,00495	0,00250	0,00349	0,99999	0,99820	0,99997	0,99999
Min	0,00030	0,00209	0,00056	0,00029	0,00042	0,00038	0,00073	0,00043	0,99719	0,98564	0,99759	0,99643
Std	0,00022	0,00429	0,00026	0,00023	0,00051	0,00507	0,00037	0,00064	0,00053	0,00299	0,00043	0,00052

Table 3

Comparing MAE of the predicted availability in each car park lot with previous works.

	P	F	KM	KP	SP	TS	RNN	BR	ABR	GBR	RFR
BHMBCCMKT01	0.041	0.053	0.087	0.086	0.059	0.067	0.063	0.0005	0.0091	0.0014	0.0006
BHMBCCPST01	0.076	0.072	0.148	0.149	0.083	0.111	0.137	0.0006	0.0144	0.0013	0.0005
BHMBCCSNH01	0.132	0.141	0.15	0.148	0.139	0.069	0.117	0.0006	0.0093	0.0011	0.0006
BHMBCCTHL01	0.122	0.142	0.134	0.131	0.123	0.08	0.103	0.0005	0.0101	0.0014	0.0005
BHMBRCBRG01	0.101	0.148	0.148	0.149	0.133	0.095	0.123	0.0007	0.0143	0.0016	0.0007
BHMBRCBRG02	0.087	0.116	0.122	0.122	0.097	0.088	0.112	0.0007	0.0107	0.0009	0.0007
BHMBRCBRG03	0.068	0.085	0.113	0.112	0.074	0.059	0.076	0.0004	0.0061	0.0010	0.0004
BHMEURBRD01	0.044	0.057	0.087	0.085	0.036	0.042	0.077	0.0013	0.0021	0.0014	0.0015
BHMEURBRD02	0.072	0.078	0.064	0.063	0.067	0.068	0.062	0.0005	0.0145	0.0010	0.0005
BHMMBMMBX01	0.063	0.067	0.074	0.072	0.084	0.129	0.084	0.0004	0.0156	0.0010	0.0004
BHMNCPHST01	0.060	0.079	0.13	0.127	0.073	0.034	0.050	0.0004	0.0141	0.0011	0.0004
BHMNCPLDH01	0.030	0.034	0.087	0.084	0.036	0.072	0.072	0.0007	0.0093	0.0012	0.0006
BHMNCPNHS01	0.072	0.084	0.082	0.078	0.06	0.082	0.102	0.0005	0.0181	0.0014	0.0005
BHMNCPNST01	0.085	0.083	0.15	0.154	0.117	0.074	0.124	0.0007	0.0100	0.0009	0.0006
BHMNCPPLS01	0.078	0.088	0.067	0.066	0.08	0.058	0.076	0.0003	0.0060	0.0009	0.0004
BHMNCPRAN01	0.083	0.094	0.143	0.14	0.084	0.055	0.089	0.0003	0.0104	0.0014	0.0003
Broad Street	0.047	0.057	0.073	0.071	0.041	0.034	0.064	0.0007	0.0154	0.0013	0.0007
Bull Ring	0.088	0.119	0.113	0.112	0.101	0.074	0.100	0.0005	0.0172	0.0013	0.0005
NIA Car Parks	0.033	0.033	0.048	0.049	0.028	0.054	0.033	0.0007	0.0081	0.0014	0.0006
NIA North	0.067	0.079	0.102	0.101	0.073	0.067	0.079	0.0005	0.0084	0.0009	0.0006
NIA South	0.040	0.036	0.064	0.064	0.031	0.078	0.053	0.0014	0.0045	0.0010	0.0009
Others-CCCPs105a	0.032	0.050	0.119	0.121	0.05	0.072	0.065	0.0005	0.0078	0.0011	0.0005
Others-CCCPs119a	0.090	0.092	0.081	0.081	0.089	0.095	0.091	0.0005	0.0054	0.0010	0.0005
Others-CCCPs133	0.083	0.108	0.093	0.092	0.087	0.061	0.091	0.0004	0.0057	0.0007	0.0004
Others-CCCPs135a	0.057	0.075	0.078	0.076	0.058	0.029	0.049	0.0006	0.0091	0.0012	0.0006
Others-CCCPs202	0.016	0.024	0.074	0.075	0.025	0.023	0.033	0.0005	0.0157	0.0012	0.0005
Others-CCCPs8	0.038	0.055	0.081	0.079	0.04	0.047	0.061	0.0003	0.0043	0.0005	0.0003
Others-CCCPs98	0.092	0.089	0.177	0.179	0.101	0.097	0.092	0.0004	0.0059	0.0009	0.0004
Shopping	0.035	0.054	0.065	0.066	0.032	0.032	0.037	0.0003	0.0036	0.0006	0.0003
Mean	0.067	0.079	0.102	0.101	0.073	0.067	0.079	0.00058	0.00974	0.00112	0.00057
Max	0.132	0.148	0.177	0.179	0.139	0.129	0.137	0.00144	0.01811	0.00163	0.00152
Min	0.016	0.024	0.048	0.049	0.025	0.023	0.033	0.00030	0.00209	0.00056	0.00029
Std	0.029	0.033	0.035	0.035	0.033	0.026	0.028	0.00022	0.00429	0.00026	0.00024

mance of our method to those obtained in previous works as shown in Table 3. The comparison was made in terms of MAE which is the only metric used in previous works. This same table shows that on average our best method (BR) has reduced the MAE by 7.8% compared to RNN (Camero et al., 2018), then by more than 6.7% compared to (Stolfi et al., 2017) which used and compared polynomials (P), Fourier series (F), k -means clustering (KM), polynomials fitted to the k -means- centroids (KP), shift and phase modifications to KP polynomials (SP), and time series (TS). In addition BR gave a very low standard deviation (0.00016) compared to that of previous work whose minimum was at least 0.026. Better still, BR turns out to be faster compared to the algorithms of previous works.

6. Conclusion

Urban mobility is one of the most important components of smart cities or even one that directly benefits citizens on a daily basis. This urban mobility is enormously affected by congestion, which is accentuated by at least 30% by the search for free places for parking (Zheng et al., 2015). The prediction of the availability of places to urban drivers considerably reduces this congestion and therefore urban pollution. Several authors have proposed approaches and models that were not optimal for this prediction. To overcome this, in this paper, we proposed a system integrating IoT and set-based regression models. The data-centric IoT that we proposed should make it possible to exploit all the connected objects in connection with the smart car parks in order to collect the data, analyze them and share the results to the drivers. Our ensemble-based model for predictive analysis optimized the prediction of the availability of parking spaces in smart car parks.

The tests that we carried out on the data of the parking lot of Birmingham made it possible for example to reach an average absolute error (MAE) of 0.06% on average with the algorithm of Bagging Regression (BR). Our best performance has improved the existing best performance of Camero et al. (2018) and Stolfi et al. (2017) by more than 6.6%, while dramatically reducing complexity. Our future work will focus on predictions of optimal travel meshes to urban taxis using regression models and IoT to increase urban eco-transport and sustainable development.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Alkheder, Sharaf A., Al Rajab, Murad M., Alzoubi, Khalid, 2016. Parking problems in Abu Dhabi, UAE toward an intelligent parking management system 'ADIP: Abu Dhabi Intelligent Parking'. *Alexandria Eng. J.* 55 (3), 2679–2687.
- Almeida, De, Paulo, R.L., et al., 2015. PKLot-A robust dataset for parking lot classification. *Expert Syst. Appl.* 42 (11), 4937–4949.
- Amato, Giuseppe et al., 2016. Car parking occupancy detection using smart camera networks and deep learning. In: 2016 IEEE Symposium on Computers and Communication (ISCC). IEEE.
- Amato, Giuseppe et al., 2017. Deep learning for decentralized parking lot occupancy detection. *Expert Syst. Appl.* 72, 327–334.
- Arasteh, H. et al., 2016. IoT-based smart cities: a survey. In: 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC). IEEE.
- Bachani, Mamta, Qureshi, Umair Mujtaba, Shaikh, Faisal Karim, 2016. Performance analysis of proximity and light sensors for smart parking. *Proc. Comput. Sci.* 83, 385–392.
- Bélissent, Jennifer, 2010. *Getting Clever About Smart Cities: New Opportunities Require New Business Models*. Cambridge, Massachusetts, USA.

- Camero, A., Toutouh, J., Stolfi, D.H., Alba, E., 2018. Evolutionary deep learning for car park occupancy prediction in smart cities. In: International Conference on Learning and Intelligent Optimization. Springer, Cham, pp. 386–401.
- De Fabritiis, Corrado, Ragone, Roberto, Valenti, Gaetano, 2008. Traffic estimation and prediction based on real time floating car data, Intelligent Transportation Systems. ITSC 2008. 11th International IEEE Conference on, 2008..
- Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A.M., Castillo, P.A., Aljarah, I., 2019. Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market. *Progr. Artif. Intell.*, 1–23
- Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In *icml* vol. 96, pp. 148–156..
- Freund, Y., Schapire, R., Abe, N., 1999. A short introduction to boosting. *J.-Jpn. Soc. Artif. Intell.* 14 (771–780), 1612.
- Gandhi, B.K., Rao, M.K., 2016. A prototype for IoT based car parking management system for smart cities. *Indian J. Sci. Technol.* 9 (17), 1–6.
- Géron, A., 2017. Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media Inc..
- Giuffrè, Tullio, Siniscalchi, Sabato Marco, Tesoriere, Giovanni, 2012. A novel architecture of parking management for smart cities. *Procedia-Soc. Behav. Sci.* 53, 16–28.
- Imandoust, S.B., Bolandraftar, M., 2013. Application of k-nearest neighbor (KNN) approach for predicting economic events: theoretical background. *Int. J. Eng. Res. Appl.* 3 (5), 605–610.
- Jin, J., Gubbi, J., Marusic, S., Palaniswami, M., 2014. An information framework for creating a smart city through internet of things. *IEEE Internet Things J.* 1 (2), 112–121.
- Khurana, M., Saxena, V., 2018. Green cover change detection using a modified adaptive ensemble of extreme learning machines for North-Western India. *J. King Saud Univ.-Comput. Inf. Sci.*
- Kubler, Sylvain et al., 2016. IoT-based smart parking system for sporting event management. In: Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. ACM.
- Kumar, Rakesh, Chilamkurti, Naveen K., Soh, Ben, 2007. A comparative study of different sensors for smart car park management. In: 2007 International Conference on Intelligent Pervasive Computing (IPC 2007). IEEE.
- Lin, Trista, Rivano, Hervé, Le Moul, Frédéric, 2017. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.*, 3229–3253
- Mago, N., Kumar, S., 2018. A machine learning technique for detecting outdoor parking. *Int. J. Eng. Technol.* 7 (2.30), 39–43.
- Mainetti, Luca et al., 2015. A Smart Parking System based on IoT protocols and emerging enabling technologies. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE.
- Mishra, S., Mishra, D., Santra, G.H., 2017. Adaptive boosting of weak regressors for forecasting of crop production considering climatic variability: an empirical assessment. *J. King Saud Univ.-Comput. Inf. Sci.*
- Pawowicz, B., Salach, M., Trybus, B., 2019. Infrastructure of RFID-based smart city traffic control system. In: Conference on Automation. Springer, Cham, pp. 186–198.
- Pow, N., Janulewicz, E., Liu, L., 2014. Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal..
- Rajabioun, Tooraj, Ioannou, Petros A., 2015. On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Trans. Intell. Transp. Syst.* 16 (5), 2913–2924.
- Rodier, Caroline J., Shaheen, Susan A., 2010. Transit-based smart parking: an evaluation of the San Francisco bay area field test. *Transp. Res. Part C: Emerging Technol.* 18 (2), 225–233.
- Sharma, V., Mir, R.N., 2019. An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm. *J. King Saud Univ.-Comput. Inf. Sci.*
- Shoeibi, N., Shoeibi, N., 2019. Future of smart parking: automated valet parking using deep Q-Learning. In: International Symposium on Distributed Computing and Artificial Intelligence. Springer, Cham, pp. 177–182.
- Sinta, D., Wijayanto, H., Sartono, B., 2014. Ensemble k-nearest neighbors method to predict rice price in Indonesia. *Appl. Math. Sci.* 8 (160), 7993–8005.
- Stolfi, D.H., Alba, E., Yao, X., 2017. Predicting car park occupancy rates in smart cities. In: International Conference on Smart Cities. Springer, Cham, pp. 107–117.
- Tang, S., Gao, H., 2005. Traffic-incident detection algorithm based on nonparametric regression. *IEEE Trans. Intell. Transp. Syst.* 6, 38–42.
- Xiang, Xuezhong et al., 2017. Real-time parking occupancy detection for gas stations based on Haar-AdaBoosting and CNN. *IEEE Sens. J.* 17 (19), 6360–6367.
- Xu, H., Ying, J., 2017. Bus arrival time prediction with real-time and historic data. *Cluster Comput.* 20 (4), 3099–3106.
- Yoo, Seong-eun, Chong, Poh Kit, Kim, Taehong, et al., 2008. PGS: Parking Guidance System based on wireless sensor network. In: 2008 3rd International Symposium on Wireless Pervasive Computing. IEEE, pp. 218–222.
- Zantalis, F., Koulouras, G., Karabetsos, S., Kandris, D., 2019. A review of machine learning and IoT in smart transportation. *Fut. Internet* 11 (4), 94.
- Zhang, Y., Haghani, A., 2015. A gradient boosting method to improve travel time prediction. *Transp. Res. Part C: Emerging Technol.* 58, 308–324.
- Zheng, Yanxu, Rajasegarar, Sutharshan, Leckie, Christopher, 2015. Parking availability prediction for sensor-enabled car parks in smart cities. In: 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). IEEE.