IEEE Open Journal of
Intelligent Transportation
Systems

Received 29 November 2021; revised 21 May 2022 and 19 October 2022; accepted 30 December 2022. Date of publication 10 January 2023; date of current version 26 January 2023.

Digital Object Identifier 10.1109/OJITS.2023.3235898

Development of a Data-Driven On-Street Parking Information System Using Enhanced Parking Features

SYRUS GOMARI [1,2], ROHITH DOMAKUNTLA [1], CHRISTOPH KNOTH [3], AND CONSTANTINOS ANTONIOU [1]

[1] Chair of Transportation Systems Engineering, TUM School of Engineering and Design, Technical University of Munich, 80333 Munich, Germany

[2] Team Connected Parking, Technical Product Design Location Based Services, BMW Group, 80788 Munich, Germany

[3] Analog and RF Verification, Infineon Technologies AG, 81726 Munich, Germany

CORRESPONDING AUTHOR: S. GOMARI (e-mail: syrus.gomari@gmail.com).

ABSTRACT On-street parking information (OSPI) systems help reduce congestion in the city by lessening parking search time. However, current systems use features mainly relying on costly manual observations to maintain a high quality. In this paper, on top of traditional location-based features based on spatial, temporal and capacity attributes, vehicle parked-in and parked-out events are employed to fill the quality assurance gap. The parking events (PEs) are used to develop dynamic features to make the system adaptive to changes that impact on-street parking availability. Additionally, a parking behavior change detection (PBCD) model is developed as an OSPI supplementary component to trigger potential parking map updates. The evaluation shows that the developed OSPI availability prediction model is on par with state-of-the-art models, despite having simpler but more enhanced and adaptive features. The foundational temporal and aggregated spatial parking capacity features help the most, while the PE-based features capture variances better and enable adaptivity to disruptions. The PE-based features are advantageous as data are automatically gathered daily. For the PBCD model, impacts by construction events can be detected as validation. The methodology proves that it is possible to create a reliable OSPI system with predominantly PE-based features and aggregated parking capacity features.

INDEX TERMS Change detection, connected vehicles, geospatial analysis, intelligent transportation systems, machine learning, parking, vehicle navigation.

I. INTRODUCTION

A. BACKGROUND

Vehicles cruising for parking are estimated to contribute to 30% congestion within a transport network [1]. This causes noise, air pollution, and travel time delays. As a parking management measure, cities have invested in parking guidance signs to direct cars to primarily off-street parking lots and multi-story car parks. Comparable systems have also recently been developed for finding parking spots on the streets, denoted as on-street parking information (OSPI). One of the benefits of such services is reduction of traffic congestion caused by cruising for a parking space [2], [3], [4], [5].

Connected intelligent transport systems (C-ITS), such as OSPI, have the potential to efficiently and better distribute vehicles within a transport network as they search for parking. Reliability and quality of such information systems must be ensured to offer dependable services that contribute to helping people make better decisions on how to navigate inside the city or whether to even use a car or not.

The content of state-of-the-art OSPI systems are mostly developed using complex engineered features and machine learning techniques [2], [3], [5], [6], [7], [8], [9]. The main

The review of this article was arranged by Associate Editor Emmanouil Chaniotakis.

difference between the models available are the data gathered for training the models and the incorporated features in the models. The differences in input data play a major role in the reliability and quality. The quality of the information provided by such systems are validated by the comparison of observed on-site data against the prediction model estimates.

B. PROBLEM STATEMENT WITH THE CURRENT SYSTEMS

Continuous manual ground truth collection for parking information systems is costly. The level-of-service and reliability remains an on-going challenge within the industry. This is attributed to difficulties in gathering accurate yet scalable data with adequate spatial and temporal coverage relative to the localized information needed. Many researches have used sensor data to develop models, but as stated in [6], these incur high costs of installation. Further, maps are usually only updated every quarter [10] as it is likewise a costly process to do so. This can be problematic when there are mid- to long-term changes that last from a few weeks to permanently. This is especially true for the case of on-street parking, since searching in an area that has obstructed parking could considerably increase the parking search time. For a parking service the sooner the changes are known, the better a service can be and parking availability models can be updated as well. As such, the goal is to provide the same quality of a prediction model with a scalable set of features based on sound domain knowledge to engineer features that rely on smart systems and less on-site surveyors.

This issue is partially tackled in this study with the use of real-time and readily available parking events data, which can be used to engineer added-value features to an OSPI service. Additionally, the same dataset could be used to specifically help parking maps be adaptive with the use of parking behavior change detection trigger.

C. CONTRIBUTIONS AND MAIN OBJECTIVE

The contributions of this research are as follows:

- The value discovery in vehicle parking events as a source to extract a wide range of features to enhance an on-street parking information system. These features include variations of hourly to weekly moving averages of time-series parked-in and parked-out data. The proposed OSPI system also has a parking events-based adaptive feature with a supplementary parking behavior change detection (PBCD) feature that is more dynamic as it can detect mid- to long-term (i.e., more than 10 days) static anomalies, closures, and disruptions signaled by the drop of parking events caused by construction obstructions, rule changes, or significant infrastructural changes, among others. These detections, essentially, convert predictions to zero to indicate unavailability of parking on top of an alert trigger to drivers to flag and confirm potential changes relating to on-street parking provisions and as an alert for the evaluation of the OSPI system. To the best knowledge of the authors, currently, there are no systems in practice or in research that updates their maps and predictions using such a dataset.
- The domain knowledge of the authors enhanced engineering of parking features from the parking events data and spatial parking capacity data previously unknown. Engineered valuable features from simple spatial capacity features that are easy to collect and prepare as input for an on-street parking availability model. Simple spatial on-street parking capacity features become more valuable when aggregated on a higher neighborhood (i.e., quadkey) level. Rather than just having the capacity information on a street-level, aggregation on a higher level can capture variances that supplements the variances captured through the street-level capacity feature.
- This proposed OSPI system can replace a system which solely relies on a prediction model that depends on continuous expensive parking availability features to keep the information system up-to-date. Shifting away from such a system lessens the cost associated with manual ground truth collection and allows faster scaling.

As opposed to many researches that have been done using complex models to create parking prediction models, this study aims to use less time-intensive machine learning algorithms that are easier to comprehend, interpret, and implement. Thus, the focus is on utilizing domain knowledge to engineer features to improve an OSPI system while using a readily available machine learning algorithm that only needs to be trained and hyperparameters-tuned. Developing a new machine learning algorithm is out of scope.

The paper is organized as follows. Related literature is described in Section II. Section III covers the main discussions of this paper. The data and study area are introduced in Section III-A. The elaboration of the development methodology of the OSPI system is presented in Section III-A1. Section IV presents the supplemental OSPI feature developed with the parking behavior change detection methodology that represent the dynamicity of the proposed OSPI system. The specifics regarding the features, algorithm hyperparameters, and the evaluation of the models are described in Section V. Section VI gives concluding remarks and some recommendations.

II. RELATED LITERATURE

The proposed approach in this study focuses on developing a data-driven OSPI system focused on valuation generation from different data sources while using prominent machine learning algorithms as the different baseline models. The logic behind this is that domain knowledge in parking can enhance the model developed. The literature review in this section is subdivided to the ground truth data used for validation in parking studies, the supplementary data used to engineer features that are not dependent on ground truth data, the popular parking prediction machine learning models that have been used in research, and the usage of parking behavior change detection models in OSPI systems. The review here mainly focuses on on-street parking.

A. GROUND TRUTH DATA USED FOR VALIDATION OF PARKING STUDIES

Most state-of-the-art on-street parking availability models developed today use a diverse range of data sources. This can be classified to two: data only used for feature engineering and ground truth data primarily used for training, testing, and validating. The latter can also be used for feature engineering.

Different types of ground truth data exist for validation of on-street parking prediction models. Some have used parking sensors in researches [8], [9], [11], [12], [13], [14], [15]. Some [9], [16], [17] have also used parking meter payments or mobile payments [3], [18] as a type of sensor to infer parking availability. A study also used costly labor-based manual observations for validation [19]. Another line of research [2], [20] have used images and videos from the camera of a moving vehicle to identify on-street parking spaces by processing these through some machine learning image recognition algorithm. Some researchers also employed crowd-sensing information by equipping probe vehicles with on-board sensors, cameras, or ultrasonic sensors [20], [21]. There are also studies who have explored the usage of crowd-sourcing data from smartphones or Global Positioning System (GPS) devices [18], [20], [22], [23].

Most of the ground truth data sources abovementioned have been studied to replace the longstanding industry practice that is still primarily based on manual ground truth collection to the best knowledge of the authors. The main reason is, each alternative ground truth is either limited in scope in different cities, such as street parking sensors and meters, and/or is unscalable. If different ground truth sources are used for each model in each city, this can be problematic as it will increase development costs of a system. Hence, the dependence on reliable manual observation.

An apparent gap that exists in all studies is that they have not tested these other ground truth sources to instead support manual ground truth to reduce frequency of manual observations required in practice. That is, the training of a model can be based on the manual observations, and the coverage-limited data gathered can be used as updates to the system since it is automatically collected albeit being sparse in space and time. The focus of the studies has been to completely replace them without direct comparisons against models that rely completely on manually gathered ground truth data.

In this study, the authors propose to use the cheaply and automatically collected sparse parking events data as a source to support manual ground truth collection and reduce the frequency of collection.

B. FEATURES IN PARKING BEHAVIOR AND PARKING PREDICTION STUDIES

On-street parking behavior and prediction studies have used a variety of features for their models. Common practice is to use the data as is as a feature and do feature engineering in this data to possibly capture different variances to better predict the target value. Two common features in research are temporal and spatial features mainly taken from the ground truth parking availability data that inherently has a location and time component. This typically is the composition of a baseline model's feature set. A few studies incorporated traffic data in their parking prediction models [6], [15], [24], [25] – this can be in the form of speed or their own engineered features to get traffic congestion indices. Some studies also have used parking-specific influencing factors such as parking pricing to understand changes in parking occupancy [26], [27]. Such factors can be used on street-level features. Another study used on-street parked out events to classify legal and illegal parking spots in the city [28]. Floating car data is another indirect source to infer parking behavior [1], [29]. Weather data has been proven by many studies to either help make prediction models or understand parking behavior [6], [9], [25]. Some other features that are also incorporated include map-related features such as street length, landuse, and points-of-interest (POI) data regarding shops, parking facilities, [1], [5], [7], [15], [19]. A few studies also included special events [5], [6]. A particularly interesting approach was done using survey data by studies like that done by Google's research team, where they asked about the subjective difficulty of parking in one's search area [30].

All studies besides a few do not give details regarding the features engineered. Particularly, a gap was observed in further aggregating simple features such as street capacity. This is typically done on temporal features, where moving averages or aggregation on various intervals are incorporated, but spatial aggregation has not been explored much according based on the literature reviewed. Studies also primarily focus on developing better algorithms than focusing on the usage of domain knowledge for feature engineering to improve their parking prediction models.

C. POPULAR PARKING PREDICTION MODELS

Parking prediction modelling studies have become popular in the last years since the hype of big data. There is a wide range of machine learning models that have been employed by researchers in the last few years. The following models have been tested in the reviewed studies: clustering [15], [21], [31], different linear regression algorithm like Lasso, Ridge, or basic linear regression [32], vector spatio-temporal autoregression [13], ARIMA [25], Support Vector Machine classifier [33], decision tree [15], [28], random forest [7], Support Vector Regression [14], [25], and tree-based algorithms like Gradient Boosting Regression Tree (GBRT) [15], [34] among others. Despite longer run times and in the hopes that unsupervised learning can enhance models, many studies have utilized deep learning approaches using neural networks like multi-layer perceptron [15], [35], CNN, Hybrid CNN, Graph CNN, RNN, LSTM, [2], [3], [6], [8], [9], [25]. Another one used logistic probability distribution and aggregating over all the observations [16]. XGBoost [36], one of the currently popular algorithms in various fields that uses a type of gradient tree boosting system that resembles an ensemble tree model, was employed by several studies [3], [7], [24] that showed the most promise in the use case of our proposed system as well. Google's research team used a single layer regression and feed forward deep neural network [30] for estimating difficult of parking using mainly Google maps travel data.

D. PARKING BEHAVIOR CHANGE DETECTION MODELS

There are no known studies that specifically use parking events to determine potential changes in parking behavior associated with longer term static changes like in rules and restrictions, constructions, or infrastructural changes. There was one study by [37] that used sensor data as well for detection of unusual patterns and infer it to any possible disturbances to parking location or sensors. Reference [28] used park-out events to detect anomalies with regards to classifying legal and illegal parking spots in relation to their map.

Majority of the studies have relied on explicit usage data input from on-street parking sensors or apps, while implicit recognition of parking occupancy has not be widely used [6]. In our study, we employ user data from parked-in and parked-out events to partly infer parking availability in conjunction with other features. The aim is to combine these data with readily available machine learning algorithms that could compete on the same level as commercial OSPI models. Although we aim to provide real-time updates to the model through introducing parking events-based features, parking events cannot be used for validation as half of the picture is missing. Fully occupied streets (true negatives) and streets that were predicted to have parking but did not (false positives) also cannot be validated with parking events, hence, it was used primarily as a source to engineer features. Nonetheless, as an added component to an OSPI system, the parking events data is also utilized to provide map triggers about potential on-street parking behavior changes that are caused by long term external factors such as construction.

III. DEVELOPMENT METHODOLOGY OF A DATA-DRIVEN ON-STREET PARKING INFORMATION (OSPI) SYSTEM

A. DESCRIPTION OF DATA USED

This section describes the data that were used in this study for training and evaluation of the model. The data that were used to extract features from are also presented. BMW's OSPI service area for the city of Munich, Germany was the chosen city use case for this paper.

The data sources are only described on a high-level to not violate BMW data confidentiality policies. Absolute numbers and descriptive statistics cannot be elaborated upon. Nonetheless, details relevant for the development of an OSPI system are described here.

1) PARKING EVENTS

Feature extraction from parking events (PEs) is one of the main contributions of this paper. Parking events (PEs) data are gathered from the fleet of BMW vehicles and are collected at BMW's backend data center. Hence, there existence of the bias towards these users. All parking events adhere to anonymization according to EU defined data privacy standards. A PE is generated when a car engine switches off or on, corresponding to a parked-in event or parked-out event, respectively (see Fig. 2). The PE event was also post processed to contain only events within the proximity of a street. Further details about the nature of the parking events dataset are discussed in [38] and [39]. As opposed to studies reliant on ground [7], [8] which cover only certain parts of a city, this research aims to utilize parking events as floating sensors.

Hundreds of thousands of parking events data used was gathered between May 2019 and October 2020 with a gap between October 2019 and February 2020.

2) GROUND TRUTH OBSERVATIONS

The ground truth (GT) data used was collected between May 2019 and October 2020. The GT observations were used for training and testing the models developed. For this dataset, the sparse data collection strategy (i.e., where, when, and how much data) was beyond the control of the authors. In the validation phase and the final scoring phase, a prioritization-based quality assessment [42] is used to adjust the scores depending on the amount of parking events that occurred in each spatio-temporal cell. This helps eliminate unimportant hours. In this study, more than 10000 random walk observations were made within the central area of Munich, Germany. Each recorded observation was made on a street block (i.e., intersection to intersection) at the time of collection. When at least one legal parking spot is observed on a block, this was recorded as available. Regardless of the number of open spots, the observations were recorded as a binary outcome – available (1) or not (0). Most foundational and important features are extracted from these observations. Among others, this includes spatial and temporal features further described in Table 1. In Fig. 1, the average parking availability aggregated on quadkey level 14 over a period of 168 week-hours is illustrated. Since observations were mostly random, there is an uneven distribution of collection throughout the city. Fig. 3 represents the spatial distribution of each observation. The average parking availability in the entire study area is 0.56. Central busy areas such as neighborhoods 6, 8, 14, and 16 (see Fig. 1) are more difficult to predict compared to the periphery.

A time series split (i.e., temporally sorted) cross validation was implemented for training and testing. In this case, testing sets here are considered the evaluation sets as well. The data is split into three equal partitions to conduct two

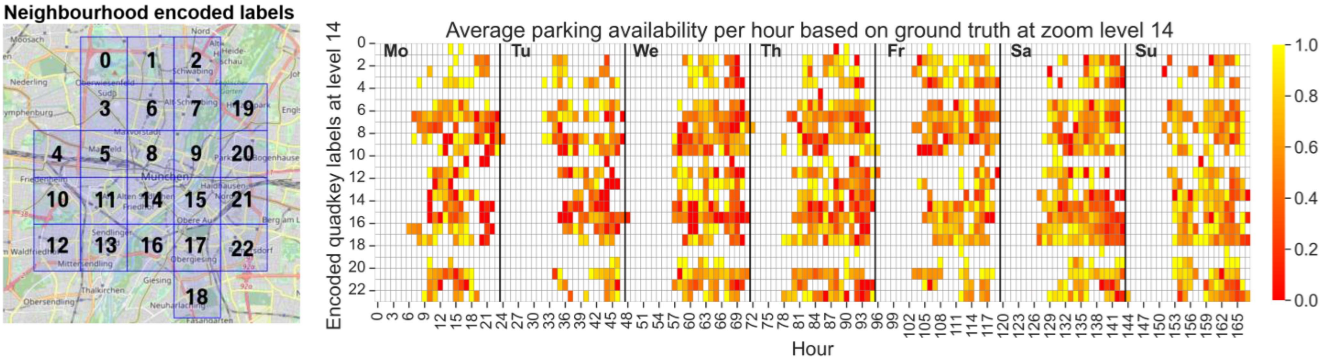


FIGURE 1. The average parking availability aggregated over 168 week-hours at zoom level 14 in Munich's study area.

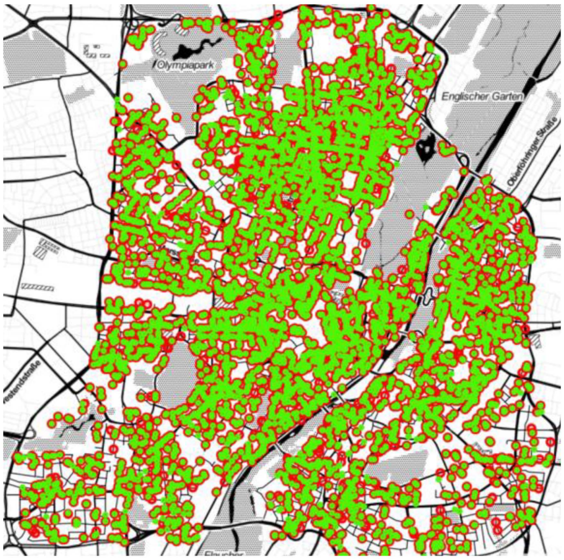


FIGURE 2. Paired parking events in Munich for one day. Green is for parked-out events and red is for parked-in events.

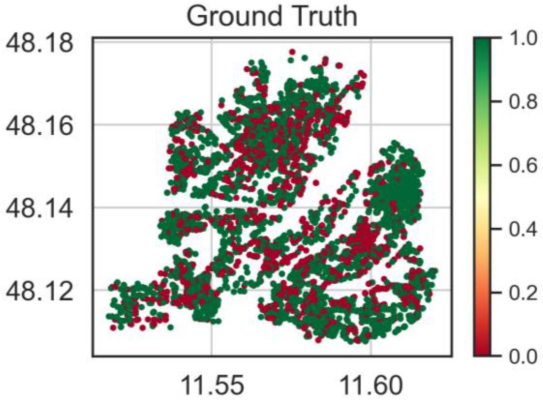


FIGURE 3. Spatial distribution of ground truth observations.

cross validation iterations. In the first iteration, the first 33% of the ground truth observations are used for training the model, and the next 33% used for evaluation. The second iteration takes the first 66% for training and the last 33% for evaluation.

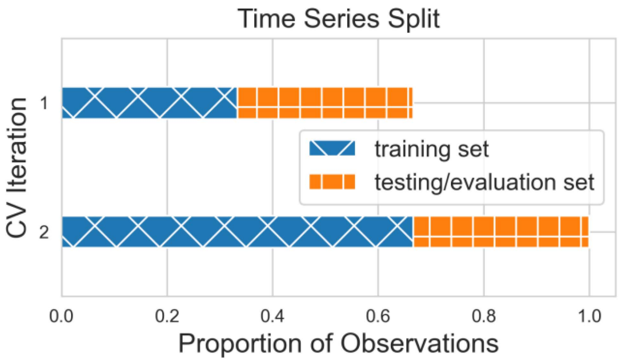


FIGURE 4. Time series split cross-validation (CV) train and test sets.

3) TRANSPORT NETWORK FOR ON-STREET PARKING

BMW's transport network consists of on-street blocks as defined above. The main feature used from here is the number of legal parking spots or on-street parking capacity of each block.

4) OTHER MAP DATA AND WEATHER DATA

To further enhance the features of the model, map data regarding construction were requested from HERE maps (2021). Furthermore, open weather data were downloaded from Deutscher Wetterdienst (2021). Only temperature and rainfall data were used in the models.

B. METHODOLOGICAL FRAMEWORK FOR OSPI DEVELOPMENT

The core feature of an OSPI system is the provision of an availability prediction to show the users the chances or difficulty of finding a parking spot in certain areas at given time periods. Particularly, the availability model that was developed in this study, as part of its novel contribution, uses mainly parking events-based features, which are dynamic in nature and uphold or improve the performance of a model. Despite the unbalanced nature of the PE dataset, the goal was to develop a model that is up to the level of commercial models. The PE dataset is unbalanced as it only provides information about open spots and occupied parking spots

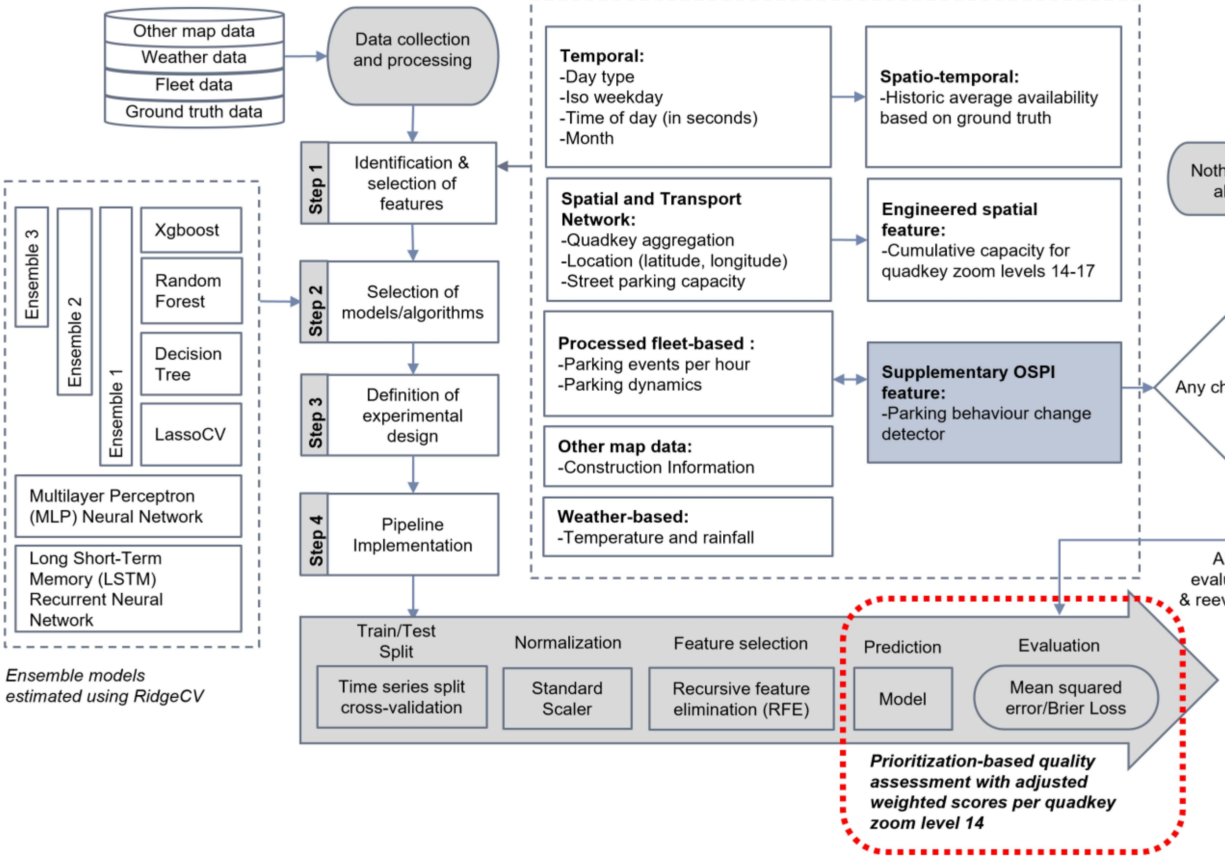

Other map data
Weather data
Fleet data
Ground truth data
Data collection and processing
Step 1
Identification & selection of features
Step 2
Selection of models/algorithms
Step 3
Definition of experimental design
Step 4
Pipeline Implementation
Ensemble 3
Ensemble 2
Ensemble 1
Xgboost
Random Forest
Decision Tree
LassoCV
Multilayer Perceptron (MLP) Neural Network
Long Short-Term Memory (LSTM) Recurrent Neural Network
Ensemble models estimated using RidgeCV
Temporal:
-Day type
-Iso weekday
-Time of day (in seconds)
-Month
Spatio-temporal:
-Historic average availability based on ground truth
Spatial and Transport Network:
-Quadkey aggregation
-Location (latitude, longitude)
-Street parking capacity
Engineered spatial feature:
-Cumulative capacity for quadkey zoom levels 14-17
Processed fleet-based :
-Parking events per hour
-Parking dynamics
Supplementary OSPI feature:
-Parking behaviour change detector
Other map data:
-Construction Information
Weather-based:
-Temperature and rainfall
Nothing to alert
No
Any changes?
Yes
Alert evaluation & reevaluate!
Train/Test Split
Time series split cross-validation
Normalization
Standard Scaler
Feature selection
Recursive feature elimination (RFE)
Prediction
Model
Evaluation
Mean squared error/Brier Loss
Prioritization-based quality assessment with adjusted weighted scores per quadkey zoom level 14

FIGURE 5. Development methodology workflow for an OSPI system.

cannot be directly inferred. Additionally, further aggregate features from basic attributes such as parking capacity were developed as described in Section V.

The OSPI availability prediction models were developed in four main steps (see Fig. 5). The overview of each step is described below. All machine learning implementation besides Xgboost was done using scikit learn [40] in Python.

1) IDENTIFICATION AND SELECTION OF FEATURES

The pre-requisite to start the development was raw data acquisition as described in Section III-A. As the first step, these datasets were used to engineer relevant on-street parking features that are identified based on related literature and domain knowledge. The descriptions of feature content are explained later in Table 1. The features were categorized as follows: temporal, spatial, weather, ground truth historic availability, fleet (parking events) data-based, and other map data.

2) SELECTION OF ALGORITHMS AND ENSEMBLE MODELS

There is a wide range of machine learning models that could be used for parking prediction. The most promising libraries shown in literature are: gradient boosting decision trees like XgBoost [36], Random Forest, and Decision Tree. Deep learning approaches with neural networks have also recently become widely popular butgiven similar performance scores in comparison with the increase in training and processing time [41] it did not seem to be promising. Furthermore, [8] mentions that neural networks perform well with high number of samples to train with like their 12 million records from Melbourne, but with smaller sizes, it may not be feasible. Also, [6] describes that neural networks are suitable when relationships are unknown and high volume of data is available. In this case, since many studies have explored which features could possibly influence the model, unknown relationships are not a big concern. Nonetheless, two neural network models namely Feed Forward Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) were implemented for baseline comparisons of all popular models used for parking studies. This is on top of the following four most popular models that were selected and tested amongst each other: Xgboost, Random Forest, Decision Tree, and LassoCV as the baseline linear regression model. Moreover, to get the best of all models, as done in [35], 3 ensemble models were created using RidgeCV as the final estimator that combines the four models (i.e., excluding neural networks) to avoid overfitting on one model.

TABLE 1. Defined feature categories.

Feature category	Description of feature content	Sample values
Temporal	Only time-related features considering aggregations into time intervals in different time scales and categorization of special days: months, weeks, days, hours, minutes, seconds, weekdays, weekends, holidays, etc.	Months: 1-12 Weeks: 1-52 Days: 1-31 Seconds in a day: 1-86400 Holidays: 0 or 1
Spatial and Transport Network	GPS location, on-street parking capacity features divided or aggregated on different spatial levels	Latitude: 48.138393 Longitude: 11.570882 Capacity per street segment: 20 Capacity aggregated on level 14: 74
Weather	Rain and temperature open data	Rainfall: 11.3mm Temperature: 13 deg Celsius
Historic parking availability	Aggregation of historic parking availability on different tile levels and time intervals (e.g. moving averages) in the past.	Available: 1 Occupied: 0 Moving average of availability on level 14: 0.61
Parking events-based	Automated aggregation in various time intervals of TTPD [38] that describe on-street parking activity on tile zoom level 14, and aggregation in various time intervals (e.g. real-time and moving averages) and tile levels of parked-in and parked-out events; anomalies detected based on the developed behavior change detection (see Section IV)	Parked-in volume aggregated on level 14 on 15-minute intervals: 13 Parked-out volume aggregated on level 14 on 15-minute intervals: 12

3) DEFINITION OF THE EXPERIMENTAL DESIGN (ED) SETUP

An experimental design setup was created to organize the process of evaluating the performance of each model by gradually adding feature categories and changing to different types of machine learning algorithms. The aim of the ED setup is to recreate and identify the best combination across algorithms and data types to allow comparison between the different setups that normally exist in the industry given the available dataset in this study. The industry replica model is developed to the best knowledge of the authors since the actual models cannot be used in publications. This ED also allows to identify if a certain setup mainly reliant on parking events-based features can be on par with an industry model and replace it or outperform the industry-level model. In total 54 ED setups were created as displayed and discussed later in Table 3. Further combinations of features for the experimental design were not necessary since even if more features are added to a prediction model, these are reduced in the feature selection step in the pipeline implementation described next.

4) MODEL PIPELINE IMPLEMENTATION

After setting up the input needed into each model, the next step was to create a pipeline implementation to maintain consistency from data transformation to evaluation. The implementation was done through the following pipeline (see Fig. 5): (1) defined the train and test strategy using the time series split cross-validation (see Fig. 4); (2) features were independently normalized using standard scaler from scikit learn; (3) since a large number of features were created, feature selection was employed using recursive feature elimination (RFE) to recursively reduce the number of features used in a model and eliminate irrelevant input features that either do not help the prediction or are redundant; (4) once the optimal features are selected to make the best predictions, these are passed on to a selected model algorithm, and the hyperparameters are tuned. The parking availability predictions are made to the resolution of a second based on the time of request. When the results are integrated into a system, they conform to the user interface (UI), e.g., to be stable, not change frequently, and update every 5 minutes for example, similar to traffic variable message signs (VMS). (5) The last step is to do the evaluation using a metric. Hours that have no ground truth data are excluded from evaluation and are a limitation of this study. Nonetheless, these hours are also considered unimportant hours in Munich based on the study of Gomari et al. [42]. The selected metric for analysis in this study was the Mean Squared Error (MSE) as described below, which is also called the Brier Loss for cases with binary outcomes:

$$MSE = \frac{1}{N} = \sum_{t=1}^{N}(p_t - 0_t)^2 \tag{1}$$

where p is the predicted probability outcome, o is the observation at instance t (0 means there was no available parking spot, 1 means there was at least one available spot), and N is the total number of instances.

MSE is used here as it can punish probability predictions that are farther away from the binary observed ground truth. For further insights, additional metric scores are calculated using the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) which can be found in the Appendix. Additionally, given the BMW user-centric system in this study, the proposed prioritization-based quality assessment of [42] is implemented. This method essentially adjusts the scores by taking the weighted sum of the scores of each quadkey at zoom level 14, denoted as KPI_p. The importance weights are based on the total volume of parking events recorded per quadkey over last 3 months of the study period.

$$KPI_P = \sum_{s=1}^{N} KPI_q \times w_q \tag{2}$$

$$w_q = \frac{PEVolume_q}{\sum_{q=1}^{N} PEVolume_q} \tag{3}$$

where KPI_q is the KPI of a quadkey, w is the importance weight assigned to a quadkey, and $PEVolume_q$ is the parking events volume in a quadkey.

All data science tasks carried out in this paper were performed in the Python scripting language. The main packages used were as follows: ADTK, xgboost, Pandas, GeoPandas, Numpy, OSMnx, Matplotlib, Seaborn, Statsmodel, PySal, Scikit-learn, and PyTorch.

IV. SUPPLEMENTAL DYNAMIC OSPI SYSTEM FEATURE: PARKING BEHAVIOR CHANGE DETECTION (PBCD)

An on-street parking availability prediction model is the core component of an OSPI system. This section presents an added-value component and feature to an OSPI system (see Fig. 5) that provides additional dynamicity external of a prediction algorithm, but still part of the OSPI system. The availability of parking events data provided the opportunity to develop a parking behavior change detection (PBCD) model to enhance a user's experience. The PBCD model described here was mainly developed to detect static longer-term changes. Long term is defined as changes that remain in place for at least some defined duration of days ranging from 3 days to 2 weeks. The idea is that the detector allows flagging of potential anomalies due to parking behavioral changes in a city's neighborhood. This then allows an update in the availability predictions made and change the values to zero to represent unavailable spots. Mainly detected are street parking capacity changes or parking rule changes that impact an OSPI system's performance. Such an automatic fleet-based change detection system aims to keep on-street parking maps up-to-date. Early detection of impactful changes helps keep the parking map reliable, accurate, and reduce costs. Furthermore, a PBCD system can alert evaluators to assess the quality of their OSPI models in identified areas by the detector.

The following sections describe the development process and the evaluation carried out for partial validation of the detector. An extensive analysis of the PBCD model is not within the scope of this study. In this paper, only the current status and potential of a PBCD model as an added component within an OSPI system is discussed.

A. METHODOLOGICAL FRAMEWORK FOR THE PARKING BEHAVIOR CHANGE DETECTION (PBCD) MODEL DEVELOPMENT

The complete workflow for the PBCD is illustrated in Fig. 6. The first step after importing parking events fleet data and the on-street parking network was to filter out and process the data. Minimum spatial level and data volume requirements were set to enable behavior change detection. Initially, the spatial requirement heuristically was set to a sub-street quadkey level 17 (approximately 306m x 306m). Each sub-street could contain more than one street block (i.e., intersection to intersection). A sub-street level analysis was chosen instead of street or block level since it was observed that disruptions only occur in small portions of a street affecting only a few parking spots. To minimize noise in the change detection, only sub-street quadkeys at level 17 with parking events greater than 100 for the whole duration of study are chosen for analysis to lessen ambiguity in results. Next, after processing the data is converted into a time series for each sub-street.

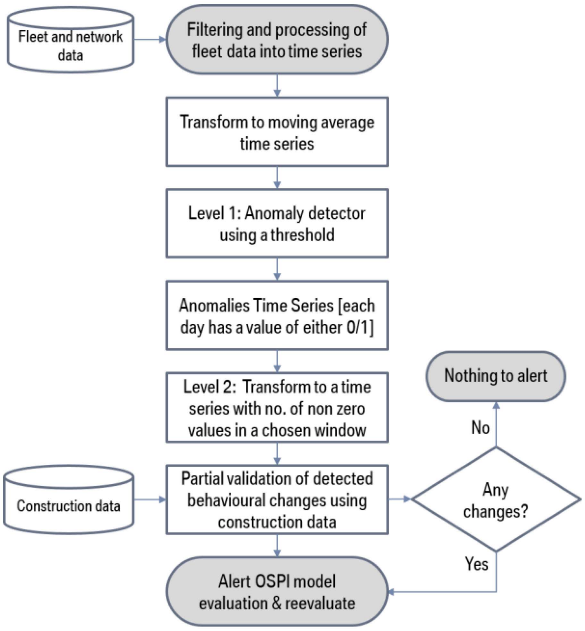

Fleet and network data
Filtering and processing of fleet data into time series
Transform to moving average time series
Level 1: Anomaly detector using a threshold
Anomalies Time Series [each day has a value of either 0/1]
Level 2: Transform to a time series with no. of non zero values in a chosen window
Construction data
Partial validation of detected behavioural changes using construction data
Any changes?
No
Nothing to alert
Yes
Alert OSPI model evaluation & reevaluate

FIGURE 6. Methodology workflow for developing a parking behavior change detection model.

The rule-based anomaly detection model developed was executed as a **two-level model** shown in Fig. 6 below. A rule-based approach was chosen heuristically based on the known disruptions in the city. For level 1, a threshold was set, and each day with a daily on-street parking volume below this was considered as anomaly and labelled as 1 (with anomaly) or 0 (no anomaly). For an anomaly to be qualified, it must satisfy the level 2 condition, which was done using a rolling aggregator that sums up anomalies and behaves consistently over a defined window number of days based on an experimental design.

The **level 1** detection: a moving average with a window size of 7 days was chosen heuristically for smoothing and transforming the time series. This transformed time series was then used to identify the first level behavioral anomalies. To further eliminate ambiguity, the removal of holidays and weekends before level 1 detection was done, to remove drops on these days, but nonetheless, nothing changed in terms of anomaly detection, indicating that these days do not impact the model. The main factor in the level 1 detection is the testing of different threshold values as cut off values. All the days in the time series which had fewer parking events than the respective threshold value were considered as anomalies. For instance, given the set threshold at 10%, all the days in the time series with parking events less than 10% quantile value are anomalies. This method ensures that all the days with comparatively fewer activities reported are identified as potential longer-term anomalies and can be marked for further analysis.

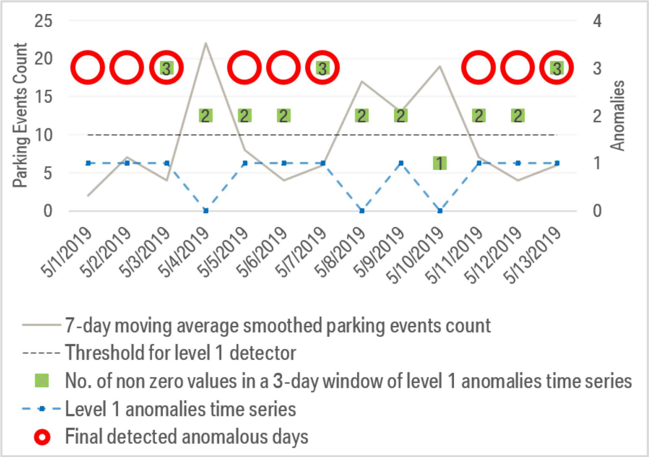


FIGURE 7. Anomaly detection model instance for time window of 3 days.

As an input to *level 2* after the threshold detector, each day in the generated anomalies time series was classified as either having a value of 0 (not an anomaly) or 1 (anomaly). Thereafter, the level 2 detector transformed the time series by performing a rolling aggregate to identify the number of non-zero values, i.e., number of anomalies of level 1 for a defined window size in days. If all the values in the considered window are 1, then all are considered as second level anomalies. Even if one of the values in the window is 0, which is not an anomaly, all the remaining values are also considered as 0. For instance, when a window of 5 days is considered, if all the days in that window are first stage anomalies, then all of them are also second stage anomalies; however, if even one day of the 5-day window is not a first stage anomaly, then all the 5 days are dropped as potential anomalous behavior. If both levels are satisfied, a warning can be triggered to change the availability status after 5 days regarding drop in the parking activity of the sub-street, which can be flagged due to a disruptive activity, such as construction, rule change, or some special event.

Fig. 7 illustrates an abstract example of the level 1 and level 2 detections from the PBCD model. The solid line represents the imaginary sample of parking events time series data after performing a 7-day moving average. Now, considering 10 parking events counts as the threshold value (dotted black line), all the days with park event values less than 10 are anomalies after level 1. This new time series with values 0 or 1 is plotted as the dashed blue line. Considering a time window of 3 days for level 2, the green squares show the values (count of number of 1's in the 3 days window) after level 2, which can be 0 or 1 or 2 or 3 and the red circles are the final anomalies after level 2, which are considered as final potential parking behavior changes. These are obtained by considering the green circles with count equal to 3 and two respective previous days as final anomalies.

The percent anomalies omitted after level 2, left-over anomalies, i.e., the days which turned out to be anomalies after level 1 but are rejected in level 2 (the day 2019-05-09 in the Fig. 7) are considered as omitted anomalies and these could also be due to construction (see Fig. 8). For instance, if the considered window in the level 2 is 15 days and all the 14 days in a window are anomalies after level 1. After level 2, none of the dates in that window are considered as anomalies as they do not satisfy the criteria of level 2. But still, they could be due to construction and therefore it is important to capture the percent of omitted anomalies which could be potential anomalies. Percent omitted anomalies within the days where there is a construction event reported could have more chances of becoming an anomaly and therefore these are also evaluated separately.

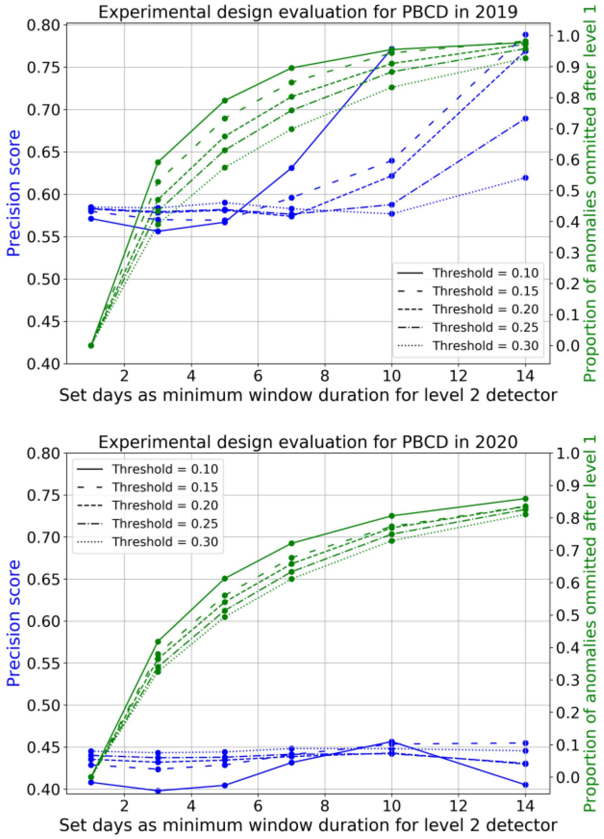


FIGURE 8. Evaluation precision scores (left y-axis) of each experimental design setup for the parking behavior change detection model including the percentage of anomalies filtered after level 1 detection (on the right y-axis).

B. EVALUATION EXPERIMENTAL DESIGN OF PBCD MODEL

To evaluate the capabilities of the defined PBCD model, the following experimental design was defined: basically, there were 5 threshold values from below 10% to 30% at 5% intervals, and 6 minimum duration values namely 1, 3, 5, 7, 10, 14, resulting to 30 experimental design setups to check the precision scores. The calculated precision score is a partial validation that presents the percentage of detected parking behavior anomalies that coincide with construction activities, although there may be other reasons for anomalies.

Construction events dataset from HERE Maps was used for the partial cross-validation of the PBCD model. It must be noted that the construction dataset may have a few shortcomings as well, such as: latency in updates and lack of information on impact on parking. The only construction information used were the period of construction and the location or street, where the construction works were observed. Each on-street parking behavior change detected by the model on quadkey level 17 was validated against the existence of construction on street level. If there is a construction on a particular day, then that day is considered as an anomaly. These days with construction events are considered as known anomalies. The precision is defined as:

$$precision = \%\ anomalies\ within\ construction = \frac{TP}{TP + FP} \tag{4}$$

where the observed value is the construction report by HERE maps and detected is an on-street parking behavior change detection. True Positive (TP) is any day which is a model anomaly and a known construction anomaly, and False Positive (FP) is defined as any day which is a detected anomaly but an unknown anomaly.

Based on field inspection, a construction observation does not necessarily mean the on-street parking segment was closed, thus, not all days with construction coincides with an anomaly. It was more often the case that when the road was open, then a parking lane was taken for this, hence, the parking segment was obstructed. Based on the sanity check of construction precision score, which means detecting that for at least one day, an anomaly is recorded within the construction period, we were able to detect at least one disruption in on-street parking for each construction event. The construction sanity precision score of 1.0 for all construction events means that all were detected at some point during their reported period of construction on a specific street. However, the overall anomaly precision scores are lower (see Fig. 8) given that there were identified changes that were not within any construction period. Hence, an anomaly detected by the model is not always caused by construction. Other detections could be other longer-term changes due to parking rules changes, an event occurring at that place for a certain period, or other potential unknown anomalies. It is also possible that, the model anomalies estimated are false change detections. This means not all model anomalies are actual changes but could be because of model inaccuracies.

C. MAIN FINDINGS FROM THE PBCD MODEL

The precision scores corresponding to the various combinations of threshold values for level 1 and the minimum window duration in days for level 2 are presented in Fig. 8. Both the scores for the 2019 and 2020 parking events data are presented. In most cases, it is observed that, higher minimum window duration values for level 2 correlates with a higher precision score. Concurrently, many level 1 anomalies are filtered out as seen with the green lines in the figure. The scores for the 2019 experimental design range from 0.55 to 0.79, while the spread is from 0.39 to 0.46 for the year 2020. The reason for the big difference in precision scores between 2019 and 2020 is the range of data used. In 2019 only 5 months of data from May to September was available, while for 2020 it was 9 months from February until October. Henceforth, the possibilities of detecting more anomalies throughout the year. Another reason for the difference is that anomalies detected in 2020 may not be due to construction; an example could be anomalies from varying restrictions due to the COVID-19 pandemic that started in March 2020 – although this is not tackled here. For both 2019 and 2020, it can also be observed from Fig. 8, that the percent proportion of anomalies omitted after level 1 increases as the minimum window duration is increased. For 2019, the precision improves as more level 1 anomalies are omitted, meaning they are unlikely to be an actual anomaly. However, for 2020, the precision score remained on the same level throughout the different experimental designs as seen in the graph. Similarly, this is attributed to other possible anomalies not related to construction.

Nonetheless, these precision scores are acceptable as it can detect some behavioral changes for which more than 55% and 40% precisions were achieved that are attributed to construction for 2019 and 2020, respectively. This is sufficient as far as the goal to use the PBCD model only as an additive component on top of the availability prediction model (see Section III-A).

Considering all the setups, the most optimal parameters are 0.20 as the threshold for the level 1 detector and the minimum window of 10 days for the rolling aggregator at level 2. With this setup, the parking behavior change detection (PBCD) model developed can detect long term disruptions which last for at least 10 days - anything below this period is neglected. The aim of the developed model was to detect long term static anomalies signaled by the drop of parking events caused by construction, rule change, or a significant infrastructural change, among others. Anomalous activities that increase the number of parking events were not part of this study. In summary, the developed model is valuable and can be used as a trigger functionality in a navigation app to flag potential changes to on-street parking provisions and as an alert for the evaluation of OSPI systems. Furthermore, the feature can be incorporated in the proposed OSPI system described in the next chapter by changing predictions to 0 for unavailability of on-street parking spots.

V. DEVELOPMENT OF A DATA-DRIVEN OSPI SYSTEM

This section presents a comprehensive comparison of different OSPI availability models based on the pipeline implementation discussed in Section III-B4) that can be used as part of the proposed OSPI system. The specific features engineered, elaboration on the usage of each feature category, and the model evaluation are discussed here as well.

A. FEATURES ENGINEERED

A relevant parking prediction study in Munich was carried out by [6] in 2016, wherein they discovered that weekday, location, temperature, and time of the day significantly improve their model performance, while information regarding traffic, holidays and rainfall only had a secondary influence. Hence, apart from traffic information, all the other features were also created and enhanced in this study. In total 102 features were extracted from the raw data available. The breakdown is as follows: 15 time-related, 7 space and location-related, 2 weather-related, 9 based on historic parking availability, 54 features related to parked-in and parked-out events, 12 related to aggregated parking events data called temporal trend of parking dynamics (TTPD) as defined in [38], and 3 related to parking behavior change detection (see Section IV). The description in Table 1 provides more information.

In summary, to create more generalized features, all the data except weather, were aggregated on different quadkey zoom levels; this is a standardized partitioning of the world map into tiles provided by Microsoft's Azure Maps [43]. Aggregation was done from zoom levels 14 corresponding to a tile size of 2446m x 2446m to smaller sizes up to level 17 of 306m x 306m. For the parking events-related and historic parking availability features, different horizons of moving averages slices were tested. A *slice* is a spatio-temporal boundary consisting of a specific quadkey and hour within the 168 hours of the week. These moving averages include taking the average value over the last 2, 4, 6, 8 hours or looking at the same week-hour and quadkey (i.e., slice) over the last 2, 4, 6, 8 calendar weeks. Another averaged value was, for example, taking the average number of parking events at each slice from the last month.

TABLE 2. **Models and tuned hyperparameters.**

Xgboost		**Random Forest**	
Parameter	Value	Parameter	Value
learning_rate	0.04	n_estimators	800
n_estimators	135	min_samples_split	10
max_depth	7	max_depth	110
min_child_weight	5	min_samples_leaf	4
gamma	0	max_features	sqrt'
subsample	0.45	**Decision Tree**	
colsample_bytree	0.65	max_depth	4
objective:	‘binary:logistic'	min_samples_leaf	57
scale_pos_weight	1	**LassoCV**	
reg_alpha	0	alpha	0.022
Multilayer Perceptron (MLP) Feed Forward Neural Network		**Long Short-Term Memory (LSTM)**	
hidden_layer_sizes	(21,)	n_epochs	50
activation	‘logistic’	num_layers	1
random_state	1	input_size	number of input features
early_stopping	True	hidden_size	21
learning_rate	‘adaptive’	learning_rate	0.001

B. MODELS AND TUNED HYPER-PARAMETERS

The optimal hyperparameters of the models change depending on the feature and the nature of the problem tackled. It was observed within all the experimental design setups, the tuned hyperparameters only marginally helped to improve the models relative to the improvements brought by features included in a model. The tuned values displayed in Table 2 are those of experimental design setup 6, which is chosen as the sample setup of the analysis.

The optimal parameters were determined using exhaustive grid search (i.e., GridSearchCV), when it was feasible, and randomized parameter optimization (i.e., RandomizedSearchCV) [40] when model runs take much more time, like in the case of the Random Forest models. For model parameters not listed in Table 2, the default values were taken [36], [40]. The 3 ensemble models created within this paper combines the different standalone models using RidgeCV, which is a linear regression model. The default alpha parameter was taken for the ensemble models.

On average, Xgboost [36] was the best standalone machine learning algorithm tested in this paper. Xgboost is a type of gradient tree boosting system, which is a tree ensemble model on its own, wherein the final prediction is based on the prediction values calculated from an aggregation of each tree. The objective function to minimize was set to binary logistic, since the problem dealt with is a logistic regression for binary classification that gives a probability output between 0 and 1. The most important parameter to tune was learning rate; the lower value, the better the predictions had become. After setting a learning rate, the number of trees (n_estimators) is determined. After a certain number of trees, the score does not improve anymore, and it plateaus. For Random Forest, the number of estimators made the most difference, but the scores did not change much in comparison with the default hyperparameters. The biggest difference observed in tuning parameters was with the Decision Tree model. After changing the minimum number of samples to be at a leaf node (min_samples_leaf) from default of 1 to 57, and updating the maximum depth from none to 4, the MSE score improved by 31%. As a baseline example for a linear regression model, LassoCV was used. LassoCV is usually used in regularization in machine learning to avoid overfitting and for feature selection. The only relevant factor to tune here was the complexity parameter alpha which was set to 0.022. For the Multilayer Perceptron (MLP) the hidden layer size was the most relevant. The optimum value for this was the desired number of selected input features after feature selection in the pipeline. For the Long Short-Term Memory baseline model, the number of epochs was the most crucial

TABLE 3. Experimental design and Prioritization-based scores for prediction models.

		Experimental design setup						
	Feature category	**1**	**2**	**3**	**4**	**5**	**6**	
1	**Temporal**	x	x	x	x	x	x	
2	**Spatial**		x	x	x	x	x	
3	**Weather**			x	x	x	x	
4	**Historic parking availability**				x		x	
5	**Parking events-based**					x	x	
	Model (M)	Model Mean Squared Error (MSE) Scores						**Average score**
1	**Xgboost**	0.2468	0.2160	0.2161	0.2166	0.2168	0.2159	0.2214
2	**Random Forest**	0.2467	0.2170	0.2161	0.2174	**0.2152**	0.2165	0.2215
3	**Decision Tree**	0.2398	0.2412	0.2400	0.2418	0.2354	0.2352	0.2389
4	**LassoCV**	0.2408	0.2316	0.2291	0.2251	0.2294	0.2253	0.2302
5	**Ensemble 1 = M1+M2+M3+M4**	0.2387	0.2154	0.2144	0.2152	0.2148	0.2157	0.2190
6	**Ensemble 2 = M1+M2+M3**	0.2423	0.2163	0.2150	0.2174	0.2148	0.2165	0.2204
7	**Ensemble 3 = M1+M2**	0.2447	0.2154	0.2148	0.2167	**0.2146**	0.2162	0.2204
8	**MLP Neural Network**	0.2330	0.2245	0.2239	0.2288	0.2288	0.2245	0.2273
9	**LSTM RNN**	0.2385	0.2392	0.2402	0.2448	0.2422	0.2437	0.2414
	Average score	0.2413	0.2241	0.2233	0.2249	0.2236	0.2233	
	Legend for each MSE score	Low	High					
	Legend for average MSE score	Low	High					

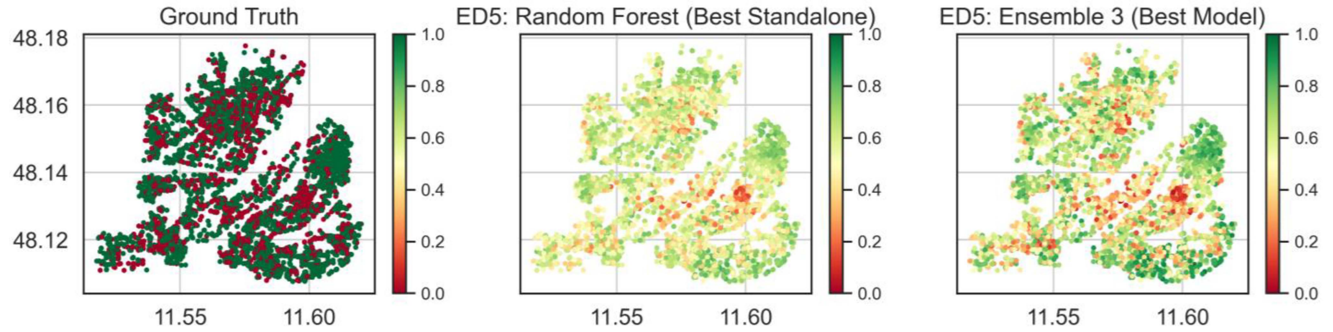

Ground Truth
ED5: Random Forest (Best Standalone)
ED5: Ensemble 3 (Best Model)
48.18
48.16
48.14
48.12
11.55
11.60
1.0
0.8
0.6
0.4
0.2
0.0

FIGURE 9. Ground truth observations (left), test set predicted probability maps of the best standalone model (middle) and the best overall model (right).

to optimize training time. After 50 epochs, the score was not improving anymore.

The main finding in the hyperparameter tuning task for the prediction model of the OSPI system is that the features selected and passed on to a model are more important compared to hyperparameter optimization unless a new algorithm is to be developed. However, for a simpler model like Decision Tree, the parameter values have a larger impact on the evaluation score. Nonetheless, tuning is vital in maximizing the performance of prediction algorithms used.

C. EXPERIMENTAL DESIGN AND EVALUATION OF OSPI AVAILABILITY MODELS

The comparative analysis of the various models based on the experimental designs is discussed in this section: the mean MSE scores, the features that help a model, the features that can replace other ones, geographical analysis, and the performance of different algorithms. The systematic process of evaluation was defined through several experimental design (ED) setups as described in Table 3. Different feature categories were gradually added as part of the experiment. For each of the 6 EDs, 9 models were used, totaling to 54 setups.

The calculated prioritization-based MSE scores [42] are illustrated in Table 3. The worst performing model scores are achieved at ED1 when only temporal features are considered. In this scenario, it can be observed that the neural network models outperform the other models as they are more capable of finding latent variances that the other algorithms cannot determine without more features. The best performing model among the 54 setups was Ensemble 3 at ED5 with a 0.2146 score, which combines Xgboost and Random Forest while taking all features except historic parking availability-based features. The best standalone model is also at ED5: Random Forest with a score of 0.2152. Each predicted probability from the test set of around 7000 observations is mapped in Fig. 9. To demonstrate the sensitivity to time in terms of average parking availability, see Fig. 10. This is the average parking availability versus the average parking probability prediction based on the best model per week-hour. And as seen, the model can predict in line with the availability patterns. If there are discrepancies, these are

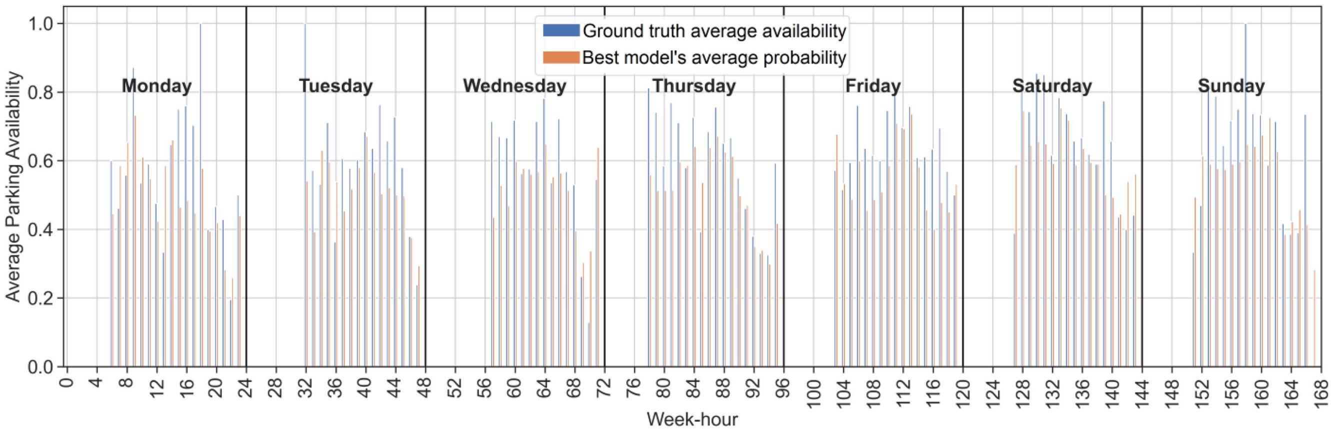
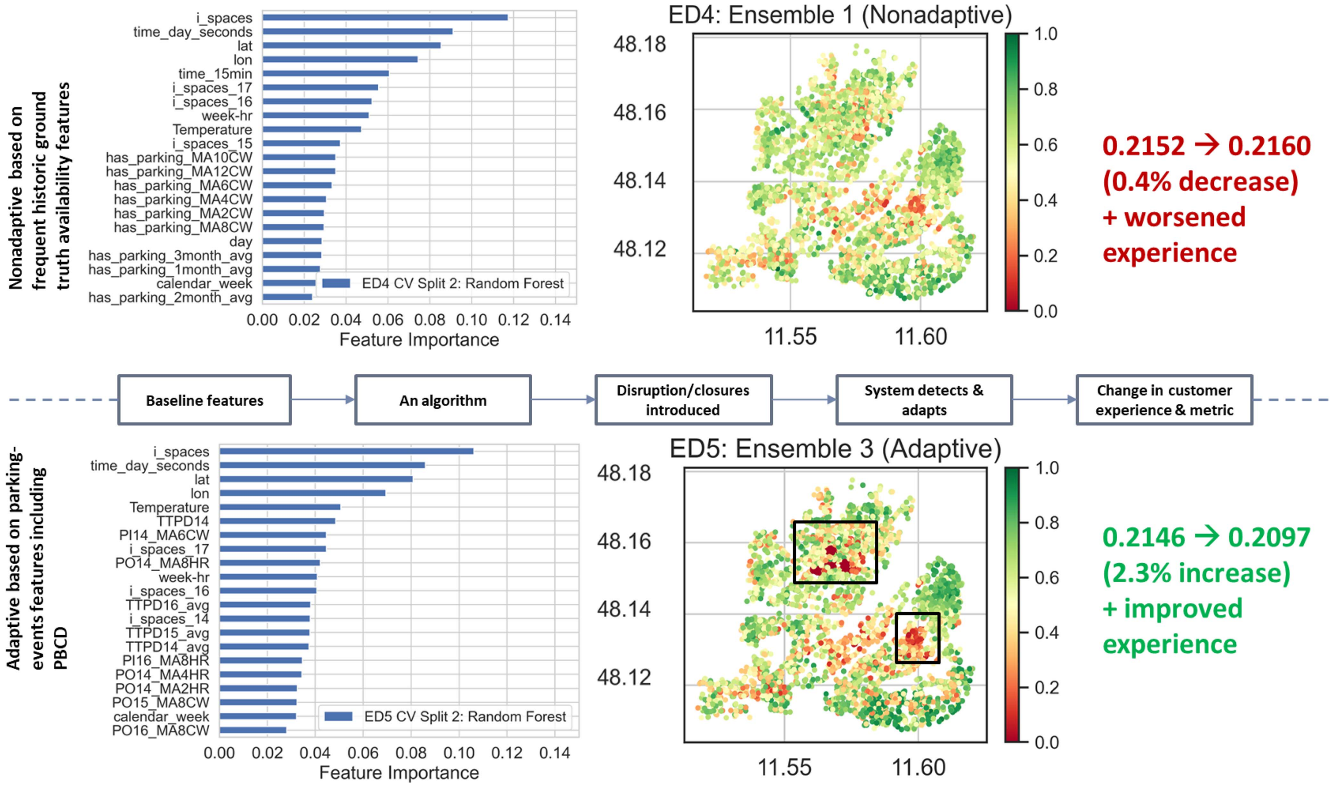


FIGURE 10. Average parking availability based on ground truth versus the best prediction model's average parking probability.



0.2152 → 0.2160 (0.4% decrease) + worsened experience

0.2146 → 0.2097 (2.3% increase) + improved experience

FIGURE 11. The stages of comparison evaluation and the differences between a nonadaptive (top) and an adaptive (bottom) OSPI system.

considered in the prioritization-based scores, which adjusts in accordance with spatio-temporal importance [42].

Fig. 9 illustrates the difference between the predictions made between the two models. ED5: Ensemble 3 has a wider spread of prediction, meaning the spread is farther from the mean. This can be observed in the maps by the larger contrast in color in the best model's predicted probability map. This translates to the model making more confident prediction.

An objective of this study was to reduce reliability on ground truth data collection and have a more dynamic data-driven OSPI that does not rely on continuous ground truth collection to reduce costs. There are **two main stages** to assess this: (1) see if an alternative model, in this case, the parking events-based model (ED5) is on par with existing industry-level models as represented by ED4; and (2) illustrate the dynamicity and advantage of the alternative model. The stages of comparison are demonstrated in Fig. 11.

Stage 1: In Table 3, it is shown that the performance of ED5 across the different algorithms implemented is in most cases outperforming the ED4 models. This makes it clear that ED5 can be a feasible alternative to an industry model that focuses on historic parking availability features for its dynamicity (see Fig. 11). To compare features, specifically, the industry-level model at ED4 using Random Forest can

TABLE 4. Prioritization-based scores after introducing disruption/closure in 5 out of 772 street segments.

	Model (M)	Model Mean Squared Error (MSE) Scores						Average score
	Experimental design (ED) setup	**1**	**2**	**3**	**4**	**5**	**6**	
1	**Xgboost**	0.2483	0.2172	0.2170	0.2180	0.2120	0.2107	0.2205
2	**Random Forest**	0.2480.	0.2181	0.2168	0.2187	**0.2103**	0.2116	0.2206
3	**Decision Tree**	0.2418	0.2425	0.2415	0.2431	0.2301	0.2299	0.2382
4	**LassoCV**	0.2410	0.2302	0.2277	0.2244	0.2242	0.2201	0.2279
5	**Ensemble 1 = M1+M2+M3+M4**	0.2395	0.2163	0.2150	0.2160	0.2099	0.2106	0.2179
6	**Ensemble 2 = M1+M2+M3**	0.2426	0.2173	0.2157	0.2184	0.2099	0.2114	0.2192
7	**Ensemble 3 = M1+M2**	0.2448	0.2164	0.2155	0.2177	**0.2097**	0.2112	0.2192
8	**MLP Neural Network**	0.2349	0.2232	0.2227	0.2295	0.2237	0.2195	0.2256
9	**LSTM RNN**	0.2408	0.2411	0.2421	0.2466	0.2366	0.2382	0.2409
	Average score	0.2424	0.2247	0.2238	0.2258	0.2185	0.2181	
	Legend for each MSE score	Low	High					
	Legend for average MSE score	Low	High					

be compared to the best standalone model ED5: Random Forest. The reason standalone models are compared is that feature importance can be directly extracted as opposed to an Ensemble model. This is obtained from the built-in feature importance attribute, determined by the proportion of the number of times a feature appeared in a tree by a model. The optimal number of features selected through various trials was 21. Thus, whenever more features were available, the 21 best features that best generalize the parking prediction were selected. The differences in features used and the respective importance factors are shown in Fig. 11. The most important features are the primary spatial and temporal features: parking spaces or capacity, time of day in seconds, and GPS location. Looking at Table 3, the primary features support each other. There are variances only captured by spatial features, that significantly improve the performance that are not captured by temporal features as seen in scores of ED1. In the ED5: Random Forest feature importance graph (lower left in Fig. 11), it can be seen that 11 out of 21 features are parking event-based. Looking at Table 3, ED5: Random Forest attains a score of 0.2152, while ED4: Random Forest attains 0.2174. This presents a 1% difference in score and can be concluded that ED5: Random Forest after replacing historic parking availability-based features with parking events-based features does not impact the performance. Thus, for stage 1 of the assessment, it can be an alternative to an industry model.

Furthermore, from the comparison of features it was discovered that aggregated spatial features appear to capture variances previously unknown. This is beneficial to further reduce reliance on historic parking availability features. On top of on-street parking capacity on a street-level, denoted as i_spaces in Fig. 11, aggregation of capacity on level 14, 16, and 17, labelled as i_spaces_14, i_spaces_16, and i_spaces_17, respectively, are capable of capturing variances and better generalize. To the best knowledge of the authors, this is a new finding that has not been discussed in research, as majority focus on directly using street parking capacity on a street-level, when this data is available. This static feature can also be updated with a dynamic feature such as PBCD that detected disruptions as discussed next.

Stage 2: For the next stage, the dynamicity is important, hence, as shown in Fig. 11, the best models are used for score comparison, and these are ED4: Ensemble 1 and ED5: Ensemble 3, respectively. To explicitly demonstrate the dynamicity of the parking events-based models at ED5 with the integration of a PBCD, on-street parking disruption or closures were artificially introduced to 5 of 772 street segments in the study area. For the entire study period, the ground truth availability is then changed to zero. This was to illustrate the difference in the performance scores for models that detect these anomalies and adapt. As seen in Fig. 11 in the two predicted probability maps, the adaptive OSPI system using parking events features and PBCD can detect the closures that are denoted with the boxes. This is visibly not detected in the nonadaptive model of ED4: Ensemble 1. Before disruption, the scores are quite similar with the nonadaptive model scoring 0.2152, and the adaptive model scoring 0.2146. However, after the closure, only the adaptive one improves its score as it is able to change its predictions based on a trigger from its PBCD. In large cities, these disruptions are difficult to detect. And often in a city like Munich, a parking closure that is left unnoticed and not updated in the system causes a compounding effect on parking search that can lead to a worsened experience of the OSPI system. Thus, a system that relies on parking events and its added PBCD feature does not only lessen the dependence on manual ground truth observations to check for disruptions, it also automatically improves user experience of the proposed OSPI system.

The complete changes in scores are shown in Table 4 with the updated scores after the introduction of disruption and Table 5 Shows the percentage difference in comparison with the prioritization-based MSE scores in Table 3. To summarize, the spread of scores based on feature category experimental design and by model used is shown in Fig. 12 Based on the average scores per feature category ED, ED 6 scores the best with an average MSE of 0.2181 after introduction of disruption, followed by ED 5, 3, 2, 4, and 1 (see Fig. 12). It is also apparent that the adaptive models in ED5 and ED6 outperform the other 4 EDs proving their advantage over models that need manual ground truth

TABLE 5. Score percentage difference between after and before disruption per model and experimental design.

	Model (M)	Model Mean Squared Error (MSE) Scores						Average score
1	**Xgboost**	-0.6%	-0.6%	-0.4%	-0.6%	2.2%	2.4%	0.4%
2	**Random Forest**	-0.5%	-0.5%	-0.3%	-0.6%	2.3%	2.3%	0.4%
3	**Decision Tree**	-0.8%	-0.5%	-0.6%	-0.5%	2.3%	2.3%	0.3%
4	**LassoCV**	-0.1%	0.6%	0.6%	0.3%	2.3%	2.3%	1.0%
5	**Ensemble 1 = M1+M2+M3+M4**	-0.3%	-0.4%	-0.3%	-0.4%	2.3%	2.4%	0.5%
6	**Ensemble 2 = M1+M2+M3**	-0.1%	-0.5%	-0.3%	-0.5%	2.3%	2.4%	0.5%
7	**Ensemble 3 = M1+M2**	0.0%	-0.5%	-0.3%	-0.5%	2.3%	2.3%	0.6%
8	**MLP Neural Network**	-0.8%	0.6%	0.5%	-0.3%	2.2%	2.2%	0.7%
9	**LSTM RNN**	-1.0%	-0.8%	-0.8%	-0.7%	2.3%	2.3%	0.2%
	Average score	-0.4%	-0.3%	-0.2%	-0.4%	2.3%	2.3%	

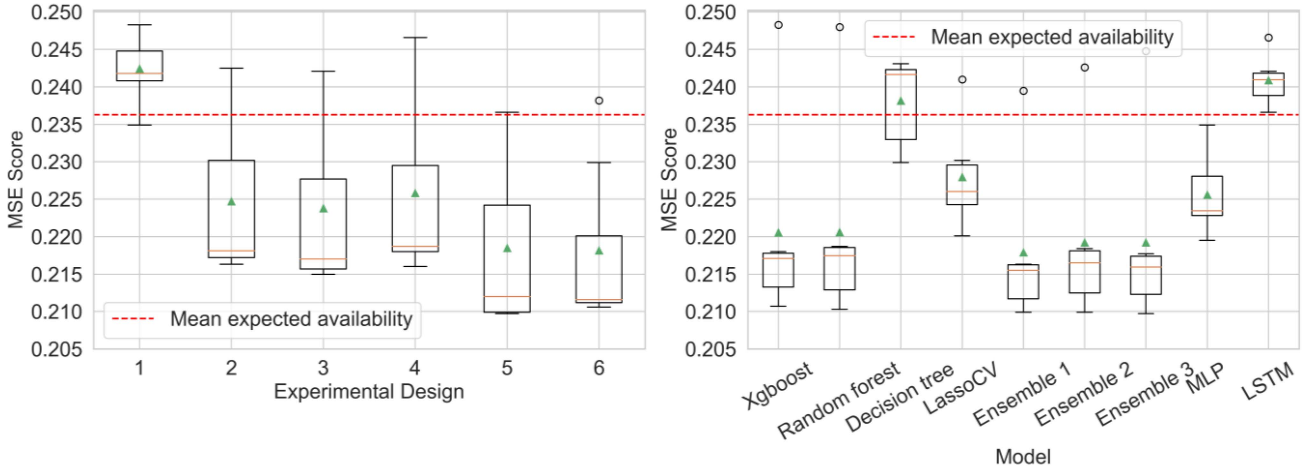


FIGURE 12. Spread of scores after disruption. Average cross-validation MSE score box plot by feature category experimental design (ED) and average cross-validation MSE score box plot by machine learning algorithm/model.

observations to identify disruptions. Meanwhile, the worst score is recorded when using Random Forest with only temporal features (ED 1). However, when spatial features are added, the Random Forest model significantly improves its performance. The same is also observed with Xgboost and the ensemble models. The Decision Tree and LassoCV models only improve with the gradual increase in features. For MLP the trend is unclear as a decline in performance was observed at ED4. For LSTM, since it is a model that primarily relies on organized time-series data, there is no significant improvements after the temporal features introduced at ED1. This proves that a domain knowledge driven models that rely on feature engineering can outperform baseline neural networks as those presented here.

ED4 includes historic parking availability features, but comparatively, it performed worse than the previous step on average. This changes at ED6 when parking events-based features are added, resulting to the best average MSE. Comparing ED5 against ED4, it can be concluded that in most models, the parking events-based features help more than the historic parking availability features. The more features provided, on average, models can capture more variances to make adjustments necessary to improve predictions – this is even more apparent with LassoCV, a simple linear model. The boxplot for the average MSE for LassoCV in Fig. 12 shows the shift from 0.2410 at ED 1 to 0.2201 at ED6 that is tabulated in Table 3.

D. DISCUSSION ON THE LATENCY OF THE PROPOSED OSPI SYSTEM

Fig. 4 shows that training only needs to take place every three months as conducted in this study. This means, the estimator factors in the machine learning algorithms employed remain the same. These factors are calibrated and adjusted based on the input values in the training sets. Table 2 presents the feature categories that contain different engineered features for each. Each feature takes a different value input. The temporal features get input in relation to a timestamp of a request. The spatial features are static based on the parking map but can be updated when the parking behavior change detection (PBCD) feature detects long-term closures or disruptions in capacity. However, for dynamic feature categories such as weather, historic parking availability and parking events-based features the system relies on ingested data. The feed or ingestion rate is different for each. For weather, hourly temperature and rainfall data can be captured. For ground truth historic parking availability data, this can only be fed into the system at random intervals depending on when data collection is scheduled with onsite observers. Thus, the system takes the historic averages

TABLE 6. Experimental design and prioritization-based root mean squared error scores for prediction models.

		Experimental design setup						
	Feature category	**1**	**2**	**3**	**4**	**5**	**6**	
1	**Temporal**	x	x	x	x	x	x	
2	**Spatial**		x	x	x	x	x	
3	**Weather**			x	x	x	x	
4	**Historic parking availability**				x		x	
5	**Parking events-based**					x	x	
	Model (M)	Model Root Mean Squared Error (RMSE) Scores						**Average score**
1	**Xgboost**	0.4965	0.4640	0.4642	0.4649	0.4649	0.4639	0.4697
2	**Random Forest**	0.4964	0.4654	0.4644	0.4660	**0.4634**	0.4648	0.4701
3	**Decision Tree**	0.4894	0.4905	0.4894	0.4913	0.4847	0.4845	0.4883
4	**LassoCV**	0.4907	0.4811	0.4785	0.4742	0.4789	0.4744	0.4796
5	**Ensemble 1 = M1+M2+M3+M4**	0.4885	0.4635	0.4624	0.4635	0.4626	0.4637	0.4674
6	**Ensemble 2 = M1+M2+M3**	0.4922	0.4644	0.4631	0.4658	0.4627	0.4646	0.4688
7	**Ensemble 3 = M1+M2**	0.4946	0.4635	0.4629	0.4652	**0.4624**	0.4643	0.4688
8	**MLP Neural Network**	0.4826	0.4734	0.4727	0.4776	0.4779	0.4735	0.4763
9	**LSTM RNN**	0.4879	0.4887	0.4898	0.4945	0.4919	0.4935	0.4911
	Average score	0.4910	0.4727	0.4719	0.4737	0.4722	0.4719	

Legend for each MSE score	Low	High
Legend for average MSE score	Low	High

available from the last collection period. This is also the reason that manual collection is not deemed feasible. Parking events-based features are engineered to aggregate values for several intervals with the shortest being 15 minutes. This means, the OSPI system predicts parking with a 15-minute latency or 15 minutes into the future. Thus, if information is requested now, the parking events-based features feed the aggregated value in the last 15-minute interval.

In the occurrence of feed failure errors, the system reverts to the last historic averages to fill in the missing data. Detailed feed failures with regards to parking events cannot be covered in this study due to the lack of access to relevant data at the point of collection.

E. MAIN FINDINGS AND DEPLOYMENT OF A DATA-DRIVEN OSPI SYSTEM

The best model based on the analysis is ED5: Ensemble 3 using temporal, spatial, weather, and parking events-based features. The ensemble model is a combination of Random Forest and Xgboost. The ED5: Random Forest standalone model was the best performing. Once combined with Xgboost, the resulting model was able to take the best of the two algorithms by learning from the weak predictions and replacing them with the advantages of the other. This is illustrated by the starker difference in predicted probabilities of the ED5: Ensemble 3 model as shown in Fig. 9. This is also interpreted as a more confident prediction model since the values are closer to a binary outcome, while improving the prioritization-based MSE score performance.

Even though an industrially accepted model such as ED4: Ensemble 1, which mainly relies on manual ground truth for updates and disruption information, model ED5: Ensemble 3 is a better model of choice for companies or institutions that have direct access to reliable incoming fleet data. This is because the best model employs features that rely on continuously available parking events data capable of capturing real-time and up-to-date variances that are needed to adjust the parking availability model. Furthermore, a parking behavior change detection (PBCD) feature based on the parking events improves the performance of the system by detecting disruptions and closures of on-street parking spaces. Such a system reduces the need to send manual observers to collect data to update the system and its relevant associated parking maps.

VI. CONCLUSION AND RECOMMENDATIONS

In the industry, manual data collection is still prevalent to ensure quality. The authors have proposed an on-street parking information system with a parking availability prediction model and a supplementary additive component that provides on-street parking behavior change detection (PBCD) using the parking events dataset. The parking availability prediction model utilizes parking events-based features and enhanced spatial features that have a better capability to generalize on-street parking capacity on different spatial aggregation quadkey zoom levels. The developed parking availability prediction model and methodology can be a competitive alternative to existing models which mainly rely on historic ground truth observations converting it to parking availability features and do not have many adaptive and dynamic features such as the parking events-based ones introduced in this paper. A wide range of feature categories and machine learning algorithms were tested as part of an experimental design to identify the best configuration of features engineered based on domain knowledge and existing algorithms.

One main advantage of the presented approach for a city like Munich, where there is abundant parking events data, is the opportunity to reduce the frequency of ground truth collection since the model can rely on incoming parking events

TABLE 7. Experimental design and prioritization-based mean absolute error scores for prediction models.

		Experimental design setup						
	Feature category	**1**	**2**	**3**	**4**	**5**	**6**	
1	**Temporal**	x	x	x	x	x	x	
2	**Spatial**		x	x	x	x	x	
3	**Weather**			x	x	x	x	
4	**Historic parking availability**				x		x	
5	**Parking events-based**					x	x	
	Model (M)	Model Absolute Error (MAE) Scores						**Average score**
1	**Xgboost**	0.4740	0.4243	0.4283	0.4288	0.4257	0.4248	0.4343
2	**Random Forest**	0.4723	0.4414	0.4437	0.4472	**0.4424**	0.4436	0.4484
3	**Decision Tree**	0.4653	0.4584	0.4602	0.4610	0.4567	0.4558	0.4596
4	**LassoCV**	0.4862	0.4707	0.4687	0.4607	0.4690	0.4613	0.4694
5	**Ensemble 1 = M1+M2+M3+M4**	0.4799	0.4308	0.4296	0.4359	0.4242	0.4274	0.4380
6	**Ensemble 2 = M1+M2+M3**	0.4872	0.4308	0.4299	0.4389	0.4242	0.4284	0.4399
7	**Ensemble 3 = M1+M2**	0.4900	0.4306	0.4304	0.4388	**0.4240**	0.4285	0.4404
8	**MLP Neural Network**	0.4689	0.4494	0.4476	0.4365	0.4376	0.4310	0.4452
9	**LSTM RNN**	0.4703	0.4759	0.4756	0.4785	0.4817	0.4884	0.4784
	Average score	0.4771	0.4458	0.4460	0.4474	0.4428	0.4432	
	Legend for each MSE score	Low	High					
	Legend for average MSE score	Low	High					

data from vehicles. This was proven by the performance ED 5: Ensemble 3 model. Although the model performs well and adapts to disruptions and closures, normal routine ground truth checks are still necessary at intermittent periods. The introduced methodology in this paper however is also limited based on the accessibility of institutions to reliable fleet data that can be used.

It is known that many special events, construction activities, rule changes occur unannounced and undocumented for, hence, the PBCD model presented in this paper can be recommended as an automated flagging component in future OSPI services, that would request for user feedback and confirmation on parking availability. This in return enables faster update of parking maps, while enhancing user experience. It would be interesting to further validate the parking behavior change detection with other data sources such as special events, and rule and infrastructure change data, among others.

The model of interest parking availability prediction model developed in this paper used the following features: temporal, spatial (location and parking capacity spatial aggregates), weather, and parking events-based. Reference [5] demonstrated the value of using their Baidu maps with refined POI data for example. However, in this research it was difficult to obtain reliable POI data without much categorization. Existing open-source POI data are unbalanced and skewed towards restaurants. Future researchers can work on OpenStreetMap POI data with an extensive category definition and cleansing that could be useful for comprehensive development of models in specific cities. The level of OSM POI data coverage is different for each city. Another recommendation is to investigate and evaluate the scores based on priority or important areas in a city [42].

In the future fast processing of videos and images will change the game, but for the meantime, the data volume of parking events is much smaller, and it will remain as a possible source for validating future researches.

APPENDIX

See Tables 6 and 7.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of INRIX in gathering ground truth data and HERE Maps for the provision of construction data, which were both fundamental assets used in this research.

REFERENCES

[1] T. Friedrich et al., "Routing for on-street parking search using probabilistic data," *AI Commun.*, vol. 32, no. 2, pp. 113–124, 2019, doi: 10.3233/AIC-180574.

[2] K. Gkolias and E. I. Vlahogianni, "Convolutional neural networks for on-street parking space detection in urban networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4318–4327, Dec. 2019, doi: 10.1109/TITS.2018.2882439.

[3] H. S. Jomaa, J. Grabocka, and L. Schmidt-Thieme, "A hybrid convolutional approach for parking availability prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852400.

[4] A. J. Pel and E. Chaniotakis, "Stochastic user equilibrium traffic assignment with equilibrated parking search routes," *Transp. Res. B, Methodol.*, vol. 101, pp. 123–139, Jul. 2017, doi: 10.1016/j.trb.2017.03.015.

[5] Y. Rong, Z. Xu, R. Yan, and X. Ma, "Du-parking: Spatio-temporal big data tells you realtime parking availability," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2018, pp. 646–654.

[6] C. Pflügler, T. Köhn, M. Schreieck, M. Wiesche, and H. Krcmar, "Predicting the availability of parking spaces with publicly available data," in *Proc. INFORMATIK*, 2016, pp. 361–374.

[7] C. Dosch, M. Farghal, M. Kapsecker, C. Lorenz, and A. Mosharafa, *Parking Prediction in the City of Melbourne*, Technische Universitaet Muenchen, Münich, Germany, 2017.

[8] W. Shao, Y. Zhang, B. Guo, K. Qin, J. Chan, and D. Flora, "Parking availability prediction with long short term memory model," in *Proc. Int. Conf. Green, Pervasive, Cloud Comput. (GPC)*, 2018, pp. 124–137.

[9] S. Yang, W. Ma, X. Pi, and S. Qian, "A deep learning approach to real-time parking occupancy prediction in spatio-temporal networks incorporating multiple spatio-temporal data sources," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 248–265, Oct. 2019.

[10] "Maps & map updates." Tomtom Website. Accessed: Aug. 27, 2021. [Online]. Available: https://www.tomtom.com/en_gb/sat-nav/maps-services/map-updates/#mydriveconnect-tab

[11] F. V. Monteiro and P. Ioannou, "On-street parking prediction using real-time data," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2478–2483.

[12] F. Shi, D. Wu, Q. Liu, Q. Han, and J. A. Mccann, "ParkCrowd: Reliable crowdsensing for aggregation and dissemination of parking space information," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4032–4044, Nov. 2019, doi: 10.1109/TITS.2018.2879036.

[13] T. Rajabioun and P. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2913–2924, Oct. 2015, doi: 10.1109/TITS.2015.2428705.

[14] A. Origlia, S. D. Martino, and Y. Attanasio, "On-line filtering of on-street parking data to improve availability predictions," in *Proc. 6th Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, 2019, pp. 1–7, doi: 10.1109/MTITS.2019.8883375.

[15] A. Ionita, A. Pomp, M. Cochez, T. Meisen, and S. Decker, "Where to park? Predicting free parking spots in unmonitored city areas," in *Proc. 8th Int. Conf. Web Intell. Min. Semant.*, 2018, pp. 1–12, doi: 10.1145/3227609.3227648.

[16] S. Yang and Z. Qian, "Turning meter transactions data into occupancy and payment behavioral information for on-street parking," *Transp. Res. C, Emerg. Technol.*, vol. 78, pp. 165–182, May 2017, doi: 10.1016/j.trc.2017.02.022.

[17] F. Bock, K. Xia, and M. Sester, "Mapping similarities in temporal parking occupancy behavior based on city-wide parking meter data," in *Proc. Int. Cartograph. Assoc. (ICA)*, vol. 1, 2018, pp. 1–5, doi: 10.5194/ica-proc-1-12-2018.

[18] B. Xu, O. Wolfson, J. Yang, L. Stenneth, P. S. Yu, and P. C. Nelson, "Real-time street parking availability estimation," in *Proc. IEEE Int. Conf. Mobile Data Manage.*, vol. 1, 2013, pp. 16–25, doi: 10.1109/MDM.2013.12.

[19] C. Ajeng and T. H. T. Gim, "Analyzing on-street parking duration and demand in a Metropolitan City of a developing country: A case study of Yogyakarta City, Indonesia," *Sustainability*, vol. 10, no. 3, p. 591, 2018, doi: 10.3390/su10030591.

[20] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," in *Proc. IEEE Int. Conf. Distrib. Comput. Sens. Syst.*, 2013, pp. 115–122, doi: 10.1109/DCOSS.2013.15.

[21] F. Bock and S. Di Martino, "How many probe vehicles do we need to collect on-street parking information?" in *Proc. 5th IEEE Int. Conf. Models Technol. Intell. Transp. Syst.*, 2017, pp. 538–543, doi: 10.1109/MTITS.2017.8005731.

[22] L. Stenneth, O. Wolfson, B. Xu, and P. S. Yu, "PhonePark: Street parking using mobile phones," in *Proc. IEEE 13th Int. Conf. Mobile Data Manage.*, 2012, pp. 278–279, doi: 10.1109/MDM.2012.76.

[23] M. Arab and T. Nadeem, "MagnoPark—Locating on-street parking spaces using magnetometer-based pedestrians' smartphones," in *Proc. 14th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2017, pp. 1–9, doi: 10.1109/SAHCN.2017.7964915.

[24] K. Ivan, T. Marta, M. Kostadin, and T. Dimitar, "Parking availability prediction using traffic data services," in *Proc. Conf. ICT Innovat.*, 2020, pp. 1–15.

[25] C. Badii, P. Nesi, and I. Paoli, "Predicting available parking slots on critical and regular services by exploiting a range of open data," *IEEE Access*, vol. 6, pp. 44059–44071, 2018, doi: 10.1109/ACCESS.2018.2864157.

[26] Z. Pu, Z. Li, J. Ash, W. Zhu, and Y. Wang, "Evaluation of spatial heterogeneity in the sensitivity of on-street parking occupancy to price change," *Transp. Res. C, Emerg. Technol.*, vol. 77, pp. 67–79, Apr. 2017, doi: 10.1016/j.trc.2017.01.008.

[27] H. Wang, R. Li, X. Wang, and P. Shang, "Effect of on-street parking pricing policies on parking characteristics: A case study of nanning," *Transp. Res. A, Policy Pract.*, vol. 137, pp. 65–78, Jul. 2020. [Online]. Available: https://doi.org/10.1016/j.tra.2020.04.003

[28] J.-E. Navarro-B, M. Gebert, and R. Bielig, "On automatic extraction of on-street parking spaces using park-out events data," in *Proc. IEEE Int. Conf. Omni Layer Intell. Syst. (COINS)*, 2021, pp. 1–7, doi: 10.1109/COINS51742.2021.9524119.

[29] O. Cats, C. Zhang, and A. Nissan, "Survey methodology for measuring parking occupancy: Impacts of an on-street parking pricing scheme in an urban center," *Transp. Policy*, vol. 47, pp. 55–63, Apr. 2016, doi: 10.1016/j.tranpol.2015.12.008.

[30] N. Arora et al., "Hard to park? Estimating parking difficulty at scale," in *Proc. 25th ACM SIGKDD Conf. Knowl. Disc. Data Min. (KDD)*, 2019, pp. 2296–2304.

[31] F. Richter, S. Di Martino, and D. C. Mattfeld, "Temporal and spatial clustering for a parking prediction service," in *Proc. IEEE 26th Int. Conf. Tools Artif. Intell.*, 2014, pp. 278–282, doi: 10.1109/ICTAI.2014.49.

[32] N. Feng, F. Zhang, J. Lin, J. Zhai, and X. Du, "Statistical analysis and prediction of parking behavior," in *Network and Parallel Computing*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-30709-7_8.

[33] A. Bhattacharyya, W. Wang, C. Tsang, and C. Amza, "Semantic-aware anomaly detection in real time parking data," in *Proc. IEEE 15th Int. Conf. Ind. Informat.*, 2017, pp. 486–491, doi: 10.1109/INDIN.2017.8104820.

[34] W. Alajali, S. Wen, and W. Zhou, "On-street car parking prediction in smart city: A multi-source data analysis in sensor-cloud environment," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage* (Lecture Notes in Computer Science 10658). Cham, Switzerland: Springer, 2017, pp. 641–652, doi: 10.1007/978-3-319-72395-2_58.

[35] F. M. Awan, Y. Saleem, R. Minerva, and N. Crespi, "A comparative analysis of machine/deep learning models for parking space availability prediction," *Sensors*, vol. 20, no. 1, p. 322, 2020, doi: 10.3390/s20010322.

[36] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[37] Y. Zheng, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Smart car parking: Temporal clustering and anomaly detection in urban car parking," in *Proc. IEEE 9th Int. Conf. Intell. Sens., Sens. Netw. Inf. Process.*, 2014, pp. 21–24, doi: 10.1109/ISSNIP.2014.6827618.

[38] S. Gomari, C. Knoth, and C. Antoniou, "Cluster analysis of parking behaviour: A case study in Munich," *Transp. Res. Procedia*, vol. 52, pp. 485–492, Jan. 2021, doi: 10.1016/j.trpro.2021.01.057.

[39] S. Gomari, C. Knoth, and C. Antoniou, "Spatio-temporal analysis of parked-in and parked-out events: A case study of Munich," in *Proc. Virtual ITS Eur. Congr.*, 2020, pp. 1–12.

[40] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011, doi: arXiv:1201.0490v4.

[41] S. Gutmann, C. Maget, M. Spangler, and K. Bogenberger, "Truck parking occupancy prediction: XGBoost-LSTM model fusion," *Front. Future Transp.*, vol. 2, pp. 1–17, Jul. 2021, doi: 10.3389/ffutr.2021.693708.

[42] S. Gomari, C. Knoth, and C. Antoniou, "Prioritization-based subsampling quality assessment methodology for mobility-related information systems," *IET Intell. Transp. Syst.*, vol. 16, no. 5, pp. 602–615, 2021. [Online]. Available: https://doi.org/10.1049/itr2.12160

[43] "Zoom levels and tile grid." Microsoft. 2020. Accessed: Jun. 20, 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/azure-maps/zoom-levels-and-tile-grid?tabs=csharp#quadkey-indic%5C