sustainability

MDPI

Article

Multisource Data Integration and Comparative Analysis of Machine Learning Models for On-Street Parking Prediction

Saba Inam [1], Azhar Mahmood [2], Shaheen Khatoon [3,*], Majed Alshamari [4] and Nazia Nawaz [1]

[1] Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad 44000, Pakistan; saba.techinsight@gmail.com (S.I.); nznawaz@gmail.com (N.N.)
[2] Faculty of Computing, Capital University of Science & Technology (CUST), Islamabad 44000, Pakistan; azhar.mahmood@cust.edu.pk
[3] School of AI and Advanced Computing, Xi'an Jiaotong Liverpool University, Suzhou 215000, China
[4] College of Computer Science and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia; smajed@kfu.edu.sa
* Correspondence: shaheen.khatoon@xjtlu.edu.cn

Abstract: Searching for a free parking space can lead to traffic congestion, increasing fuel consumption, and greenhouse gas pollution in urban areas. With an efficient parking infrastructure, the cities can reduce carbon emissions caused by additional fuel combustion, waiting time, and traffic congestion while looking for a free parking slot. A potential solution to mitigating parking search is the provision of parking-related data and prediction. Previously many external data sources have been considered in prediction models; however, the underlying impact of contextual data points and prediction has not received due attention. In this work, we integrated parking occupancy, pedestrian, weather, and traffic data to analyze the impact of external factors on on-street parking prediction. A comparative analysis of well-known Machine (ML) Learning and Deep Learning (DL) techniques, including Multilayer Perceptron (MLP), Random Forest (RF), Decision Trees (DT), K-Nearest Neighbors (KNN), Gradient Boosting (GA), Adaptive Boosting (AB), and linear SVC for the prediction of OnStreet parking space availability has been conducted. The results show that RF outperformed other techniques evaluated with an average accuracy of 81% and an AUC of 0.18. The comparative analysis shows that less complex algorithms like RF, DT, and KNN outperform complex algorithms like MLP in terms of prediction accuracy. All four data sources have positively impacted the prediction, and the proposed solution can determine the best possible parking slot based on weather conditions, traffic flow, and pedestrian volume. The experiments on live prediction showed an ingest rate of 0.1 and throughput of 0.3 events per second, demonstrating a fast and reliable prediction approach for available slots within a 5–10 min time frame. The study is scalable for larger time frames and faster predictions that can be implemented for IoT-based big data-driven environments for on-street and off-street parking.

Keywords: smart city applications; Internet of Things; predictive analytics; on-street parking prediction

Citation: Inam, S.; Mahmood, A.; Khatoon, S.; Alshamari, M.; Nawaz, N. Multisource Data Integration and Comparative Analysis of Machine Learning Models for On-Street Parking Prediction. *Sustainability* **2022**, *14*, 7317. https://doi.org/10.3390/su14127317

Academic Editors: Zubair Baig and Mark Anthony Camilleri

Received: 26 March 2022
Accepted: 13 June 2022
Published: 15 June 2022

1. Introduction

Due to massive urbanization, traffic volume in urban areas has grown, making urban life very congested and polluted, leading to many negative impacts on human life, such as higher energy consumption, global warming, and airborne diseases [1]. According to World Resource Institute [2], 74% of CO_2 is produced by greenhouse gas emissions, and 93% of it results from fossil fuel usage, transportation, manufacturing, and consumption. In fact, 2020 has been recorded as the hottest year per NASA analysis. For the sustainable development of cities, the efficient use of resources and the adoption of effective measures have become crucial for survival. We have witnessed the COVID-19 effects in different areas of life, making the internet and information the heart of modern and sustainable cities. To reap the benefits of the internet and Information Communication Technologies (ICT),

many city governments have initiated the concept of a "Smart City" with the deployment of advanced ICT aiming to provide a better living experience to its citizens [3]. At the heart of a smart city is the Internet of Things (IoT) which enables different devices to interact and draws upon various underlying operations of a smart city for sustainable living such as smart services, smart health, smart transportation, smart agriculture, smart energy to name a few [4].

The goal of sustainable transport in smart cities is to ensure efficient traffic movement while minimizing a negative impact on the environment and public health [5]. The most discussed area in smart cities is intelligent transportation highlighting its impact on intelligent mobility, the environment, and the economy. For example, Cisco Barcelona Jurisdiction Profile 2014 [6] reveals an annual increase of $50 million through parking fee revenues using smart parking technology. The main goal of smart parking is finding and providing appropriate parking for each user. However, the problem of finding a parking area is still challenging due to increased traffic flow in urban areas. For example, some studies reveal that an average of 30–40% increase in traffic is caused by drivers looking for vacant parking spots, and on average, a New York driver spends 107 h a year searching for a parking spot [7]. This phenomenon has increased air pollution and has had a negative environmental effect. With an efficient parking infrastructure, the cities can reduce carbon emissions caused by additional fuel combustion and avoid delays and traffic congestion while looking for a free parking slot.

In previous research, smart parking solutions are mainly categorized as off-street and on-street [8,9]. Off-street parking includes garages and closed parking spaces which could be outdoors or indoors. Off-street, the problem is simpler since it is straightforward to count the number of available slots by counting the number of cars entering and leaving a closed parking space. Off-street parking management has been tackled quite well due to its simpler problem and data availability [10]. On the other hand, on-street parking is challenging due to the absence of parking entrances and significant changes in occupancy rates as more cars enter and leave the spots. On-street parking can directly affect streets regarding traffic congestion and air pollution. Numerous research has been done on both problems leading to an effective search for vacant parking spots. The research is usually based on parking spaces equipped with sensors to sense whether the spots are occupied and provide information. The data from occupancy sensors allows us to learn availability patterns and predict probabilities of parking occupancy of the spots. Based on Parking Sensor Data (PSD), various machine learning methods have been used to predict parking occupancy rates [1,11]. The most common ML used for parking prediction are Regression Trees [12–14], DTs [14], Support Vector Machine [13,15], Genetic Algorithm [16], Bayesian [17], and Neural Network [13,18]. The performance of these models depends on the accuracy of information provided to users about the availability of parking lots. However, multiple traffic factors may influence car parking activity regarding on-street parking. For example, the occupancy status may change due to other traffic factors present at that time, such as weather, pedestrian mobility, and traffic volume; therefore, the information provided by PSD is not very efficient. These factors can influence car parking conditions; therefore, it is essential to identify possible factors to predict future parking availability accurately.

It is necessary to install many sensors in various cities with substantial setup costs to collect data for contextual factors. Using a publicly available dataset can provide a good starting point for understanding the impact of external elements on the real-time prediction of the availability of parking spaces. Therefore, in this paper, we took advantage of publicly available data from the sensors deployed in the City of Melbourne (COM). Based on the literature review, on-street car parking, pedestrian, traffic, and weather data are identified as possible relevant categories of data that can influence prediction accuracy. None of the existing studies has investigated the influence of these factors on their predictions.

The main objective of this study is to design, build and evaluate an end-to-end ML pipeline for an on-street parking prediction using multisource data. We plan to integrate

the ML pipeline into a smart parking application for future experimentation. The research objectives are achieved through the following contributions:

1. We integrated four datasets, i.e., car occupancy, weather, pedestrian, and traffic datasets, for more reliable predictions. The research question is: does the integration of multisource data impact the prediction accuracy of ML/DL models?
2. We investigated the relationship of car occupancy data with other external factors such as pedestrian volume, traffic flow, and weather conditions. The research question is: to what extent can each external factor help in improving the accuracy of the occupancy prediction model?
3. We analyzed the performance of well-known and generally used ML/DL models (e.g., MLP, RF, DTs, KNN, GA, AB, and linear SVC) to identify the best prediction model. The research question is: which ML/DL model can be used to achieve a more accurate prediction?
4. Using basic streaming operations, we deployed the best On-Street Prediction (ONSP) model for real-time prediction. The simulation results have shown an ingest rate of 0.1 and throughput of 0.3 events per second, demonstrating a fast and reliable prediction approach for available slots within 5–10 min. The research question is: how to scale the solution for IoT-driven big data environments to achieve sustainable parking solutions?

The rest of the paper is structured as follows—Section 2 reviews related work to summarize previous research on parking prediction. Section 3 describes the methodology adopted for predictive analysis based on a multisource data-driven approach. Evaluation of machine learning algorithms and comparison results of different modeling approaches are discussed in Section 4. Finally, the future insights on intelligent real-time decision-making for smart city applications are discussed in Section 5, followed by the concluding remarks.

2. Review of the Scientific Literature

The prediction of car park availability is the subject that has received significant attention in the context of smart cities where parking facilities have installed sensors as part of their infrastructure. Many research efforts have focused on improving parking search efficiency, reservation, and prediction for an available parking space. For example, Kizilkaya et al. [19] used a hierarchical approach for predicting free parking spots using a binary search tree (BST). For the experiment, synthetic data is used with attributes such as parking distance, capacity, and availability status. The approach first searches for the nearest parking location and then finds a free spot in the nearest car park. Horng [20] used the Artificial Fish Swarm Algorithm(AFSA) to minimize search time and traffic congestion. The performance is evaluated through simulations by randomly distributing 300–1800 vehicles in a 5.0 $\times$ 5.0 km^2 field. The results are compared with the conventional opportunistic methods revealing the effectiveness of AFSA in terms of reducing search time and congestion. However, the studies discussed above are promising, but the focus is limited to exploring only algorithmic capability in the domain.

Similarly, Thomas and Kovoor [16] used Genetic Algorithm (GA) to solve the scheduling problem in the parking system, but the proposed prototype can only be used for reserving parking spots. The decision-making of parking slots is based on the maximized fitness score of the GA objective function. The performance analysis metrics include efficiency, utilization, and average waiting time. Customers can book a parking slot in advance for a specific time period. When parking time duration exceeds, the system sends a notification of time exceeded. All the parking information is stored in the cloud. A multi-criteria decision analysis-based Parking space Reservation (MCPR) algorithm is proposed by Rehena et al. [21] for improvement in the reservation algorithm. The MCPR automatically finds the nearest parking space based on the users' preferences, from parking space availability to pricing for the reservations. The studies mentioned in this section is a step further in optimizing the decision capability for predicting parking slot and highlighting the significance of using Machine Learning (ML) algorithms.

The previous research efforts indicate that several ML algorithms have been widely studied and explored in the direction of predicting occupancy. For example, Raj et al. [22] used parking data and tested the Random Forest method for predicting parking spots in a parking lot. The question is how contextual data and other ML methods can predict parking availability. Stolfi et al. [23] used historical car parking occupancy data from the Birmingham city council for testing various prediction strategies such as polynomial fitting, Fourier series, K-means clustering, and time series to predict future occupancy. The results are validated using K- fold cross-validation with the final output testing on unseen occupancy data. The solution is made available for the users through a webpage. However, it faces challenges due to the inconsistency in the sensor's data, as the data may not be updated for the whole day. Klandev et al. [24] used garage occupancy and traffic congestion data to predict the parking spot availability ratio within 60 min. They tested the XGBoost regression model, which received a low error rate confirming its efficiency of predictions.

Similarly, Claudio et al. [17] compared different prediction techniques utilizing traffic flow, weather, and historical data to predict parking in the city garages of Florence. The resulting solution proved the Bayesian regularized network for reliable and fast predictions. Zheng et al. [13] perform a comparative analysis of SVM, Regression Tree, and Neural Networks using San Francisco and Melbourne datasets to predict long-term occupancy in 24 h intervals. The results indicate that the Regression tree outperforms the other two methods they evaluated with the highest accuracy and minimum error rate.

The discussed research shows the importance of contextual data such as traffic and weather being used along with ML approaches, but they are only tested on off-street parking prediction. Alajali et al. [12] investigated the use of on-street car parking, pedestrian, and daily traffic count data to predict short-term parking slots using Boosting Regression Tree. The study was implemented for a particular location Central Business District Melbourne, using data for special days and events since getting pedestrian counts were costly and hard to scale. Here, only in one study, the impact of pedestrian data and traffic is utilized for on-street parking. The results show that multisource data had an improved performance using gradient boosting (GBRT) with MSE 0.029. Still, the results are reported with only pedestrian data as the traffic data lacked proper mapping with other sources due to limited availability.

One of the critical challenges in addressing parking prediction is considering the nature of underlying data, suitable predictive models, and the accuracy of real-time decision-making. All the research done has focused on either one or the other challenge. For example, Liu et al. [8] proposed an online parking guidance system considering the delay in real-time parking space availability. The authors discussed the multiuser online street problem, and the study was validated on a Melbourne dataset. The results illustrated that the proposed framework reduces 63.8% delay.

On the other hand, Vlahogianni et al. [25] proposed a two-step methodological framework for real-time car occupancy prediction based on sensor data. The first step predicts the real-time parking space using Recurrent Neural Networks (RNN). The second module is based on finding the available parking space with traffic volume. This approach, however, proved computationally expensive.

Among the studies discussed above, each has tried to solve different problems, such as minimizing delay and congestion, techniques to deal with inconsistent sensor data, and ML methods to improve the prediction accuracy for on-street and off-street parking prediction. Many studies used only car parking sensor data to evaluate the predictive performance of ML methods. Only a few studies focus on contextual factors such as traffic flow, weather, or pedestrian mobility data. However, these studies are evaluated on off-street prediction problems such as city garages and parking lots, where data accessibility of occupancy status is easier to obtain. None of the existing studies has investigated the relationship of car occupancy data with weather conditions, traffic count, and pedestrian mobility in their predictive models for on-street parking.

Additionally, most studies are evaluated via simulation, and very few are evaluated in real-time. Real-time studies lack computational scalability, which is crucial for today's smart city applications. There have also been gaps in one way or the other, such as taking advantage of multisource data for on-street parking and making a solution scalable for real-time predictions. This work proposes a scalable predictive solution for real-time on-street parking prediction utilizing multisource data. To the best of our knowledge, this is the first study that serves as a starting point toward integrating multisource in designing and developing real-time parking solutions.

3. Materials and Methods

The overview of the methodology is shown in Figure 1. In the first step, data preparation and integration are performed on the historical data (occupancy, pedestrian, traffic, and weather data) to evaluate the impact of each data segment on prediction accuracy. Then different ML techniques are implemented and evaluated to select the one with the highest prediction performance on the historical data. Finally, the best prediction model is deployed to perform the real-time prediction. In the next section, each step is described in detail.

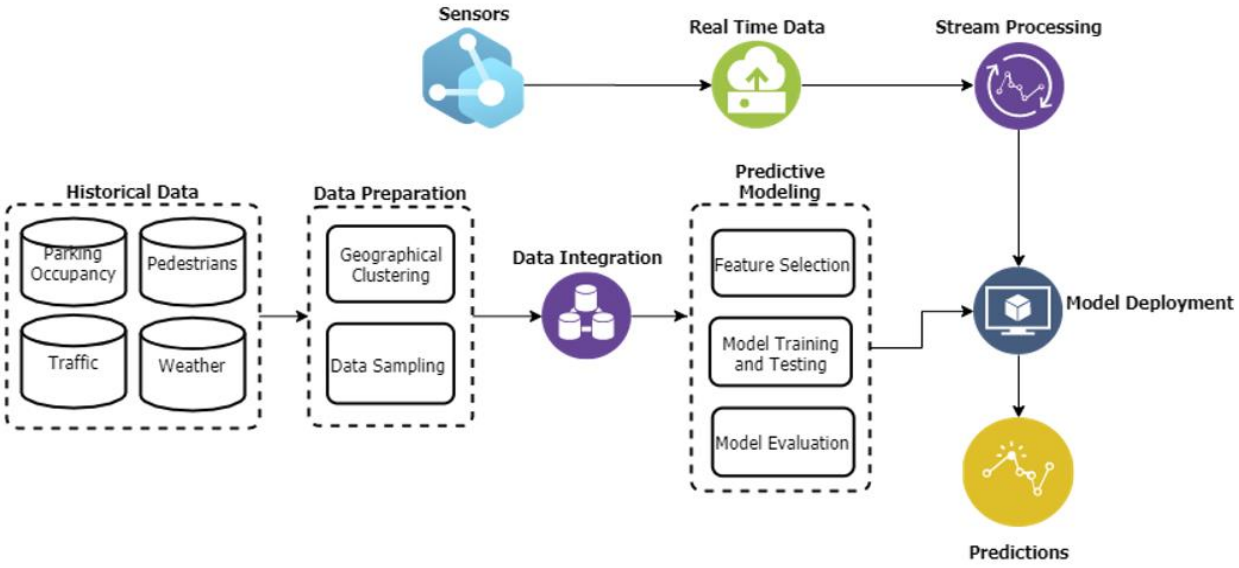

Figure 1. Overview of the methodology.

3.1. Data Collection and Description

This work took advantage of open data obtained from the City of Melbourne (COM) (https://data.melbourne.vic.gov.au/browse?category=Transport&sortBy=most_accessed&src=fpc, accessed on 25 February 2021). COM has created an Open Data Portal that contains a multitude of transport-related datasets collected from sensors installed on different streets of the city. We downloaded three datasets for different domains, i.e., parking occupancy (https://data.melbourne.vic.gov.au/Transport/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i, accessed on 25 February 2021), pedestrian (https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-2009-to-Present-counts-/b2ak-trbp, accessed on 25 February 2021), and traffic (https://data.melbourne.vic.gov.au/Transport/Traffic-Count-Vehicle-Classification-2014-2017/qksr-hqee/data, accessed on 25 February 2021), from 1 January 2017 to 31 December 2017. We utilized relevant APIs to retrieve the weather data (https://www.worldweatheronline.com/developer/api/historical-weather-api.aspx, accessed on 15 July 2020) for the same period to analyze the impact of weather data on parking slot prediction. The size of each data source is shown in Table 1.

Table 1. A summary of the datasets.

Data Source	Melbourne City
Time interval	1 January 2017 to 31 December 2017
Car parking sensor data	35.9 million records
Pedestrian sensor data	3.09 million records
Car traffic data	60.2 K records
Weather data	Hourly forecast

Car parking sensor data is primary data generated from ground sensors installed in each street in the city. It reports report car parking events in each slot, such as arriving time, departure time, presence or absence of a vehicle, etc. Parking event data is sufficient for basic modeling to identify the number of free and occupied spaces if slots are numbered. However, for the on-street parking problem, all parking slots are not numbered, and street sensors report the presence or absence of a vehicle at a specific slot. Furthermore, the space can quickly occupy based on traffic conditions and other environmental factors around the vicinity. Therefore additional data is needed to identify the impact of contextual factors on parking utilization. We used pedestrian, traffic, and weather datasets to analyze the implications of pedestrian mobility, traffic load, and weather conditions on car parking events for contextual factors. Pedestrian sensors report hourly counts of pedestrians at any given location, while traffic data reports hourly counts of vehicles. We downloaded historical weather data for the same city during the same period for every hour of the day. Tables 2–5 present the feature of each of the original datasets. The dataset from all four sources resulted in 21,334,807 rows, and each record consisted of 51 fields/features. The dataset is considered big data. Data preparation, feature selection, and integration are made to draw the most influential features from each original dataset discussed in the next section to extract the relevant records representing all data sources.

Table 2. Features description of on-street parking occupancy data.

Features	Description
Duration Seconds	Time difference between arrival and departure events
Area	City area used for administrative purposes
Street Id	A GIS key that describes the street segment where the sensor is located
Street Name	Street upon which the vehicle parked
BetweenStreet1	Closest intersecting street Id in front of the parked vehicle
BetweenStreet2	Closest intersecting street Id behind the parked vehicle.
Side Of Street	Side of the street on which the parking event occurred 1 = Centre; 2 = North; 3 = East; 4 = South; 5 = West
In Violation	This indicates that the Parking event exceeded the legal duration
Vehicle Present	Indicates whether the parking slot is free or occupied

Table 3. Features description of pedestrian data.

Features	Description
Month	Month of year (January, February, . . . December)
Mdate	Day of year (1, 2, 3, . . . , 31)
Day	Day of week (Monday, Tuesday, . . . , Sunday)
Time	Time of day (0 = midnight–1 a.m.; 1 = 1 a.m.–2 a.m.; 2 = 2 a.m.–3 a.m.; . . . ; 23 = 11 p.m.–midnight)
Sensor_Name	Sensor name
Sensor_ID	Sensor ID
Hourly_Counts	Total hourly sensor readings (count of pedestrians)

Table 4. Features description of traffic data.

Features	Description
location	The location of the sensors
suburb	The suburb where the road is located
speed_limit	The speed limit of the road
traffic_count	Total hourly sensor readings (count of vehicles)
average_speed	The average speed of vehicles crossing a sensor
85th_percentile_speed	The speed at or below which 85% of vehicles in traffic stream travel. This speed is likely to be influenced by traffic conditions, so it reflects the conditions during the analysis period.
maximum_speed	The maximum speed traveled over the sensor
road_segment, road_segment_1, road_segment_2	The road segment where the survey was conducted

Table 5. Features description of weather data.

Features	Description
maxtempC	day max temperature in °C (Celsius)
mintempC	day min temperature in °C (Celsius)
totalSnow_cm	total snowfall amount in cm
sunHour	total sun hour
uvIndex	UV Index
uvIndex.1	UV Index 1
moon_illumination	moon illumination in %
DewPointC	dew point temperature in °C (Celsius)
FeelsLikeC	feels like temperature in degrees Celsius
HeatIndexC	heat index temperature in °C
WindChillC	wind chill in °C
WindGustKmph	wind gust in kilometers per hour
cloudcover	cloud cover in percentage (%)
humidity	humidity in percentage (%)
precipMM	precipitation in millimeter (mm)
pressure	pressure in millibar (mb)
tempC	the hourly temperature in °C (Celsius)
visibility	visibility in kilometers (km)
winddirDegree	the wind direction in degrees
windspeedKmph	wind speed in kmph (kilometer per hour)

3.2. Data Preparation and Integration

In the data preparation phase, data is prepared in a more suitable way for parking prediction modeling. The overall process of data preparation is shown in Figure 2. The data is sampled and downsized using multistage cluster sampling [26] and non-proportional quota sampling [27]. The data is first clustered geographically among occupancy, pedestrian, and traffic segments in the multistage cluster sampling to identify common streets in different datasets. No overlapping was observed among these three datasets; therefore, clustering was performed among two segments, 'occupancy and pedestrian' and 'occupancy and traffic.' The weather data was mapped for both clusters, and both clusters were then merged. We found 20 common streets between pedestrian and occupancy data segments and seven overlapping streets among occupancy and traffic data segments.

In the next phase, sub-subsets are retrieved from both clusters using non-proportional quota sampling. The traffic data was far less than the pedestrian data, and to ensure the equal representation from uneven datasets, the occupancy weather and pedestrian (O_W_P) and the occupancy weather and traffic (O_W_T) subsets were retrieved with 49% of '0' class representation and 51% of '1' labels. The multistage cluster sampling resulted in two clusters, i.e., occupancy weather and pedestrian (O_W_P) and occupancy weather and traffic (O_W_T). The non-proportional quota sampling picked an equal set of

samples from both clusters. Finally, both groups' datasets are merged, resulting in a single dataset representing samples from all four sources. The downsizing of data resulted in 118,244 rows.

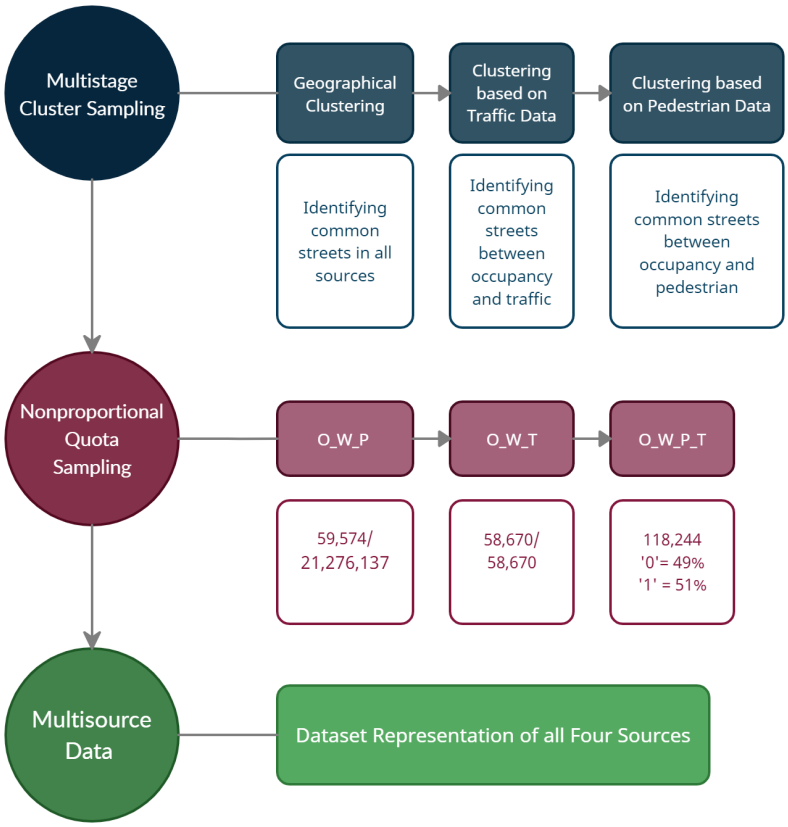

Figure 2. Two steps process of multisource data preparation.

Figure 3 shows all the features from the individual data sources that are included in the multisource data on-street parking prediction. The computed features include 'traffic_count', 'T_BetweenStreet1′, and 'T_BetweenStreet2′. The 'traffic_count' from the traffic segment is a count derived from summing all the vehicle features present at any given location at a specific time. The features 'T_BetweenStreet1′ and T_BetweenStreet2′ are derived from the attribute 'road_name' and mapped with the occupancy data. The final feature selection from the integrated data is conducted through algorithm scrutiny in the next stage under the predictive model.

3.3. Predictive Modeling

The core component of the on-street parking predictive model is understanding the impact of multisource data in batch and settings.

Various machine learning algorithms are trained in batch processing on historical data. We compared the testing performance of MLP, Linear SVC, KN, DT, GB, AB, and RF. RF has shown improved accuracy with 22 features; hence it is selected as the ONSP model. The features were ranked using GINI Index [28,29], which is computed as follows:

$$G(t) = 1 - \sum_{k=1}^{Q} p^2(k\backslash t)$$

where (t) is the node of a tree and $(p\ (k\backslash t)\ : k = 1, \ldots Q)$ are the estimated class probabilities and (Q) is the number of classes.

Apart from training and validating the ONSP on integrated data, the model is also evaluated to investigate its predictive capabilities of occupancy prediction in different com-

binations of pedestrian volume, traffic flow, and weather conditions. The feature capabilities and predictive performance of such combinations are discussed in the results section.

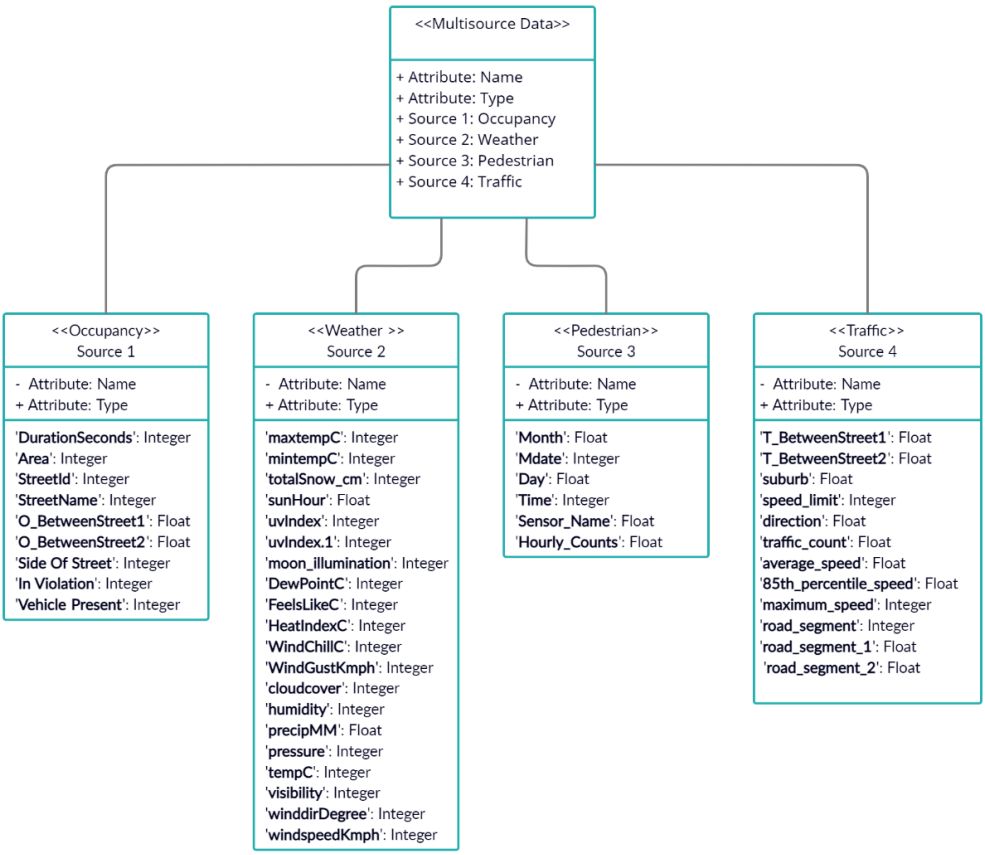

Figure 3. Features integration of multisource data.

In the next step, the model trained on the historical data is deployed on Watson Machine Learning for processing events on the fly using IBM Streams Flow. We used 100 instances of integrated data with 22 features for testing the streaming application using IBM Cloud Shell. The instances were simulated using the python REST API utilizing Flask (https://flask.palletsprojects.com/en/1.1.x/, accessed on 15 November 2021), deployed on the IBM Cloud Foundry service. An end-to-end pipeline for of entire process is shown in Figure 4. With the ready deployed model and data, source streamflow is generated for performing streaming analytics. A streaming analytics dashboard is launched to monitor predictions using the measure events per second (EPS), discussed in the result section.

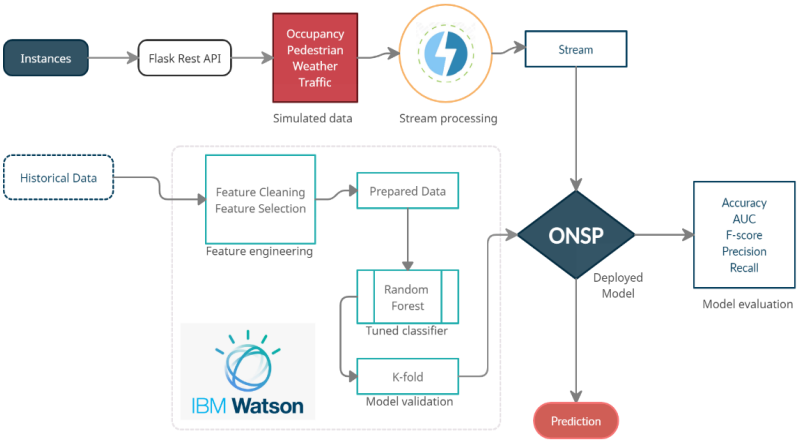

Figure 4. End-to-end pipeline for the ONSP model training, deployment, and real-time predictions.

4. Experimental Results

The experimental setup, along with the results of model training and validation on historical and real-time data, is discussed in this section.

4.1. Experimental Setup

The experiments are performed using various available resources. The system requirements used for data engineering are described in Table 6. To evaluate the impact of all four sources and model testing and validation, the Jupyter Notebook 6.0.1 is accessed via Anaconda 3 with python version Python 3.7.4. on Google Colaboratory (https://colab.research.google.com/notebooks/intro.ipynb, accessed on 12 September 2021). The ONSP training and deployment are performed on IBM Watson using IBM Cloud. All the required services, such as Watson Studio, Watson Machine Learning, streaming analytics, and IBM cloud object storage, were added to IBM Cloud using the Lite version. The Lite version is freely available but with limited computing hours.

Table 6. Summary of the experimental setup.

System	Specifications
Local Machine	Intel Core i7-8565-U, 64-bit OS, 16.0 RAM
Google Collaboratory	Python 3 Google Compute Engine backend (TPU)
IBM Watson Studio	4 vCPU and 16 GB RAM, Default Python 3.6 S

4.2. Feature Selection

By applying the feature selection technique discussed in Section 3.3 the important features are selected based on their importance score to understand the predictive capabilities of features from integrated datasets. The resultant features identified by the proposed feature selection technique from the integrated dataset (O_W_P_T) are shown in Figure 5, ranked from highest to lowest scores. It can be observed from Figure 5 that DurationSecond and SideOfStreet received the highest score from the occupancy data. Other features from occupancy data are streetID and StreetName. Hourly_Count, Area, and time of the day received the highest scores from pedestrian data. Traffic between streets, which was mapped with the occupancy data points, received a high score from the traffic data. However, due to the small traffic dataset, the traffic condition does not show much significance. From the weather data winddirDegree, humidity and cloudcover have shown the highest score. Other features from weather data are WindGustKmph, pressure, moon_illumination, DewPointC, tempC, WindChillC, and feelLikeC, which indicates the importance of weather data in occupancy prediction. Only 22 features (shown in Figure 5) are retained for further analysis out of 51 features from the original datasets based on the feature scores.

Besides the feature's capabilities of the integrated dataset (O_W_P_T), understanding the importance of each feature as the possible predictive variable for different combinations of the datasets is worth mentioning. The importance of each feature in all possible combinations of the historical dataset is shown in Figure 6. These combinations include occupancy (O), occupancy_weather (O_W), occupancy_pedestrian (O_P), occupancy_traffic (O_T), occupancy_weather_pedestrian (O_W_P), occupancy_weather_traffic (O_W_T), and occupancy_pedestrian_traffic (O_P_T).

It can be observed from Figure 6 that the important features from Occupancy (O) data include DurationSeconds, Side Of Street, and the area between streets, making them basic features for all other combinations. When occupancy is combined with pedestrian data in (O_P), the hourly pedestrian count becomes the next influential variable, while in (O_T) combination, traffic count received the highest scores making the traffic flow the most influential variable from the traffic dataset. The winddirDegree, humidity, and cloudcover are amongst the top weather features in the O_W combination. The Hourly_Counts, Mdate, and Time from the pedestrian and traffic_count, maximum_speed, and 85th_percentile_speed

came as the top traffic features from the O_P_T combination. The figure proved that the parking duration (DurationSeconds) and location (Side Of Street, Area, O_BetweenStreet1, and O_BetweenStreet2) are present in every combination hence making them the most influential variables in every combination. The other influential variables are traffic count, pedestrian hourly count, and wind degree from traffic, pedestrian, and weather data, respectively, proving that the availability of a free parking slot is tightly coupled with the number of pedestrians, traffic flow, and weather conditions.

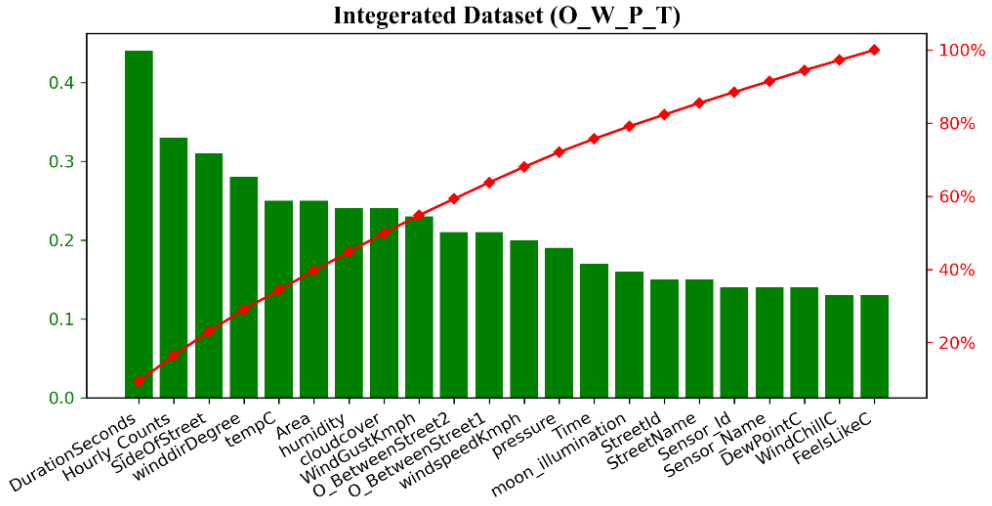

Figure 5. Feature importance scores of the integrated dataset (O_W_P_T).

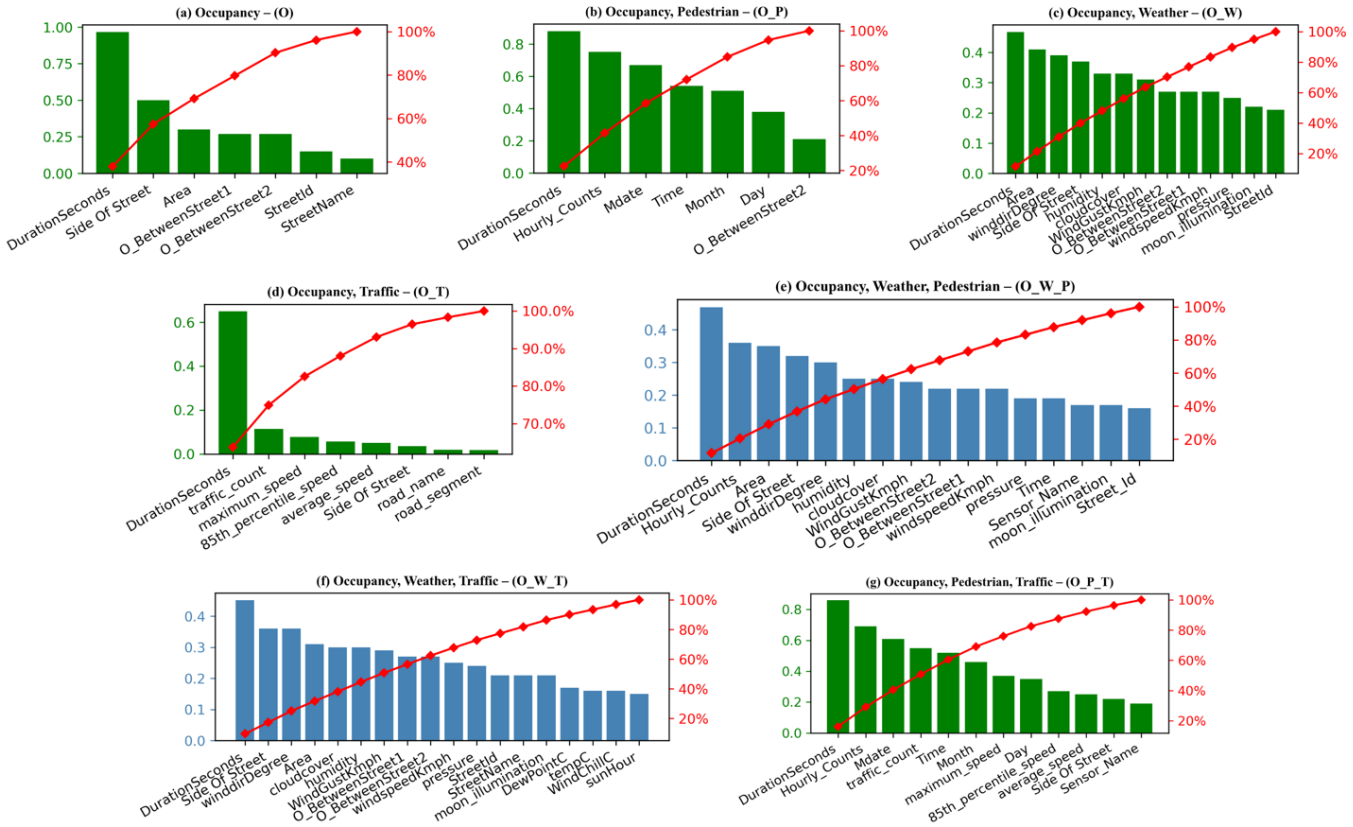

Figure 6. Feature importance scores of the different combinations of the dataset: (**a**) Occupancy (O); (**b**) Occupancy, Pedestrian(O_P); (**c**) Occupancy, Weather (O_W); (**d**) Occupancy, Traffic (O_T); (**e**) Occupancy, Weather, Pedestrian (O_W_P); (**f**) Occupancy, Weather, Traffic (O_W_T); (**g**) Occupancy, Pedestrian, Traffic (O_P_T).

4.3. Performance Evaluation of ML/DL Models

We performed two sets of experiments to evaluate the performance of selected algorithms, first on the integrated database and the second on different combinations of multisource datasets. At first ONSP model is trained using RF and tested on integrated data with the 22 most influential features. The performance is compared using well-known evaluation metrics accuracy, recall, F1-score, and AUC (Area under Curve) [11,30]. A comparative analysis with the various models such as DT, KNN, GB, AB, MLP, and linear SVC is performed. Although there are many ML/DL techniques in the literature, we choose these six techniques since they are widely used by the community and have proven the best results. Secondly, this is a preliminary work focused on identifying the impact of contextual data points on the prediction accuracy of the most used techniques. The model validation is performed using 6 K-fold cross-validations [31] for traditional ML approaches as part of training. For the deep learning model, MLP, we adopted five-layered sequential network architecture and tuned different hyperparameters discussed in the next section.

The results obtained are shown in Table 7, from which we generated Figure 7, which clearly illustrates these comparative performances. Overall the ONSP showed the best results for all measurements compared to other models. It received a training accuracy of approximately 80%, higher than DT, KNN, GB, AB, Linear SVC, and MLP which is 77.7%, 64.0%, 61.7%, 58.8%, 51.5%, and 58.8%, respectively. Besides accuracy, the model is evaluated in terms of other metrics such as precision, recall, AUC, and F-score (see Table 7 and Figure 7). With an 81% value for each of these metrics, it is proved that ONSP is prone to a low number of false-positive rates compared to other classifiers under the study, where values vary from 51 to 78%. The comparative analysis revealed that the complex model, MLP showed the lowest performance with an average of 58.8% accuracy, 60% precision, 56% recall, 62% AUC, and 58% F-score. In contrast, one of the simplest ML models, DT, outperformed MLP with the results of 78% precision, recall, F-Score, and 77.7% accuracy. KNN and GB also outperformed MLP, and their performances were quite close to each other. KNN showed 63% average precision, while GB had 62% average precision. The average recall scores for KNN and GB were 63% and 61%, respectively. AB and Linear SVC showed the lowest performance of all other ML algorithms, with 58% and 51% testing accuracy.

Moreover, the results are compared with similar research on on-street car parking prediction [12], where Gradient Boosting has achieved maximum performance. However, the model was only evaluated to explore the relationship between car occupancy and pedestrian data of 13.2M rows with 57 streets. We validate the ONSP model to assess the relationship of multisource data, i.e., car occupancy, pedestrian, weather, and traffic data of size 22 M rows with 24 streets. In the data sampling, we shortlisted the common streets among different data segments, and it has shown its impact on the results by achieving a testing accuracy of 81%. Furthermore, we applied the feature reduction technique to select the most influential variables. The proposed ONSP model performed better than those that used the original features in terms of accuracy and other performance variances.

Table 7. Training and testing performance of ML/DL models on the integrated dataset.

Classifier	Training	Testing				
	Accuracy	Accuracy	AUC	Precision	Recall	F-Score
ONSP	79.8	81	0.81	0.81	0.81	0.81
DT	77.7	78	0.78	0.78	0.78	0.78
KNN	64.0	63	0.63	0.63	0.63	0.63
GB	61.7	61	0.61	0.62	0.61	0.60
AB	58.8	58	0.58	0.59	0.58	0.57
Linear SVC	51.5	51	0.51	0.65	0.51	0.36
MLP	58.8	58	0.62	0.60	0.56	0.58

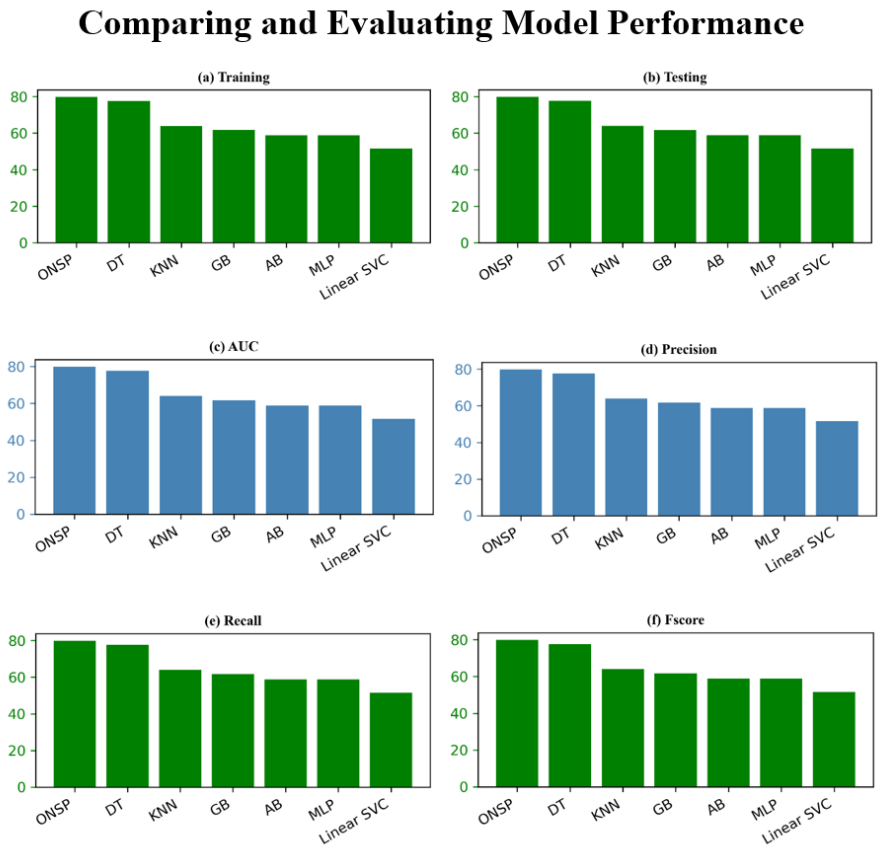

Figure 7. Graphical representation of predictive models' performance: (**a**) Cross-Validation accuracy at K = 6, (**b**) testing accuracy, (**c**) AUC—Area under the curve, (**d**) Precision, (**e**) Recall, and (**f**) F-score. In all the performance measures, ONSP has shown improved accuracy with CV = 80% and testing improvement with 81%, along with precise outcomes amongst all models at 0.81.

4.4. Performance Evaluation of Multilayer Perceptron (MLP) Neural Network

To evaluate deep learning with traditional ML approaches, we adopted MLP, which is the most common neural network. The MLP architecture consists of an input layer, three hidden layers, and an output layer. We ran multiple iterations to determine hidden layers and neuron size range by considering accuracy and loss. As a result, three hidden layers (24, 24, 8) with 24, 24, and 8 neuron sizes are being used in the network. The hyperparameters used to tune MLP are shown in Table 8. We trained MLP for three different epoch sizes of 100, 250, and 1500 on different batch sizes with adam optimizer and binary cross-entropy loss function to test the training loss and accuracy. The hyper-parameters "learning_rate" and "learning_rate_init" are responsible for optimizing and minimizing the loss function. We used the "adaptive" learning rate to keep the learning rate constantly equal to the initial learning rate as long as there is a decrease in the training loss in each epoch. "Activation" determines how active a specific neuron (hidden unit) is. We adopted the widely-used ReLU activation function to determine how active a specific hidden unit is.

The training and testing results of different epochs and batch settings (EiBi, where i is the size of each epoch and batch) are shown in Table 9. Figure 8 is generated from Table 9 to identify the most stable epochs and batch pairs in terms of loss and accuracy. In Figure 8, the loss has a slight decreasing curve as we move from E1B5 to E5B15 with an increased value of AUC. However, we can notice that F-score and recall dropped first but became stable again at E5B15. Hence, the E5B15 setting, with the epoch size of 500 (E5) and batch size of 1500 (B15), resulted in the highest training and testing accuracy of 0.588 and 0.584, respectively, with a loss of 0.653 and an F-score of 0.581. In Figure 9, the accuracy and loss curves of different epoch combinations and batch sizes are shown. It can be observed from

Figure 9c that epoch size of 500 and batch size of 1500 (E500, B1500) have better results compared to (E150, B1000) and (E500, B1000).

Table 8. Hyperparameters of MLP model.

Parameter	Value
hidden_layer_sizes	(24, 24, 8)
activation	ReLU
learning_rate	Adaptive
learning_rate_init	0.001
optimizer	Adam
Loss function	Binary cross-entropy
Batch size	Seven different batch sizes
Epocs	(250, 100, 1500)

Table 9. Training and testing of MLP at different epochs and batch sizes.

ID	Training				Testing				
	Epoch	Batch	Accuracy	Loss	AUC	Precision	Recall	F-Score	Accuracy
E1B5	100	500	0.563	0.661	0.589	0.546	0.866	0.669	0.562
E1B1	100	1000	0.565	0.657	0.593	0.547	0.862	0.669	0.563
E1B15	100	1500	0.583	0.657	0.619	0.586	0.636	0.610	0.583
E1B2	100	2000	0.587	0.654	0.619	0.586	0.622	0.603	0.581
E1B25	100	2500	0.588	0.653	0.623	0.590	0.636	0.612	0.587
E1B3	100	3000	0.582	0.653	0.621	0.599	0.566	0.582	0.583
E1B35	100	3500	0.584	0.654	0.619	0.595	0.558	0.576	0.579
E25B5	250	500	0.584	0.654	0.622	0.598	0.559	0.578	0.582
E25B1	250	1000	0.581	0.656	0.616	0.590	0.591	0.590	0.580
E25B15	250	1500	0.584	0.656	0.614	0.582	0.600	0.591	0.575
E5B5	500	500	0.582	0.654	0.620	0.602	0.523	0.560	0.579
E5B1	500	1000	0.586	0.653	0.620	0.606	0.526	0.563	0.582
E5B15	500	1500	0.588	0.653	0.622	0.600	0.564	0.581	0.584

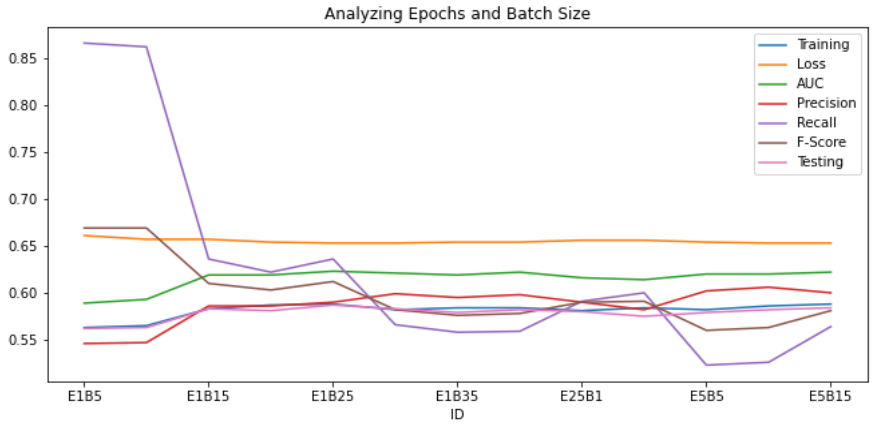

Figure 8. Comparative analysis of MLP performance using different epochs and batch sizes. The E1B5 shows higher recall and F-score but lower values of AUC, training, and testing. All the evaluation parameters converged at E5B15 with a lower loss curve and higher F-score, AUC, training, and testing accuracy.

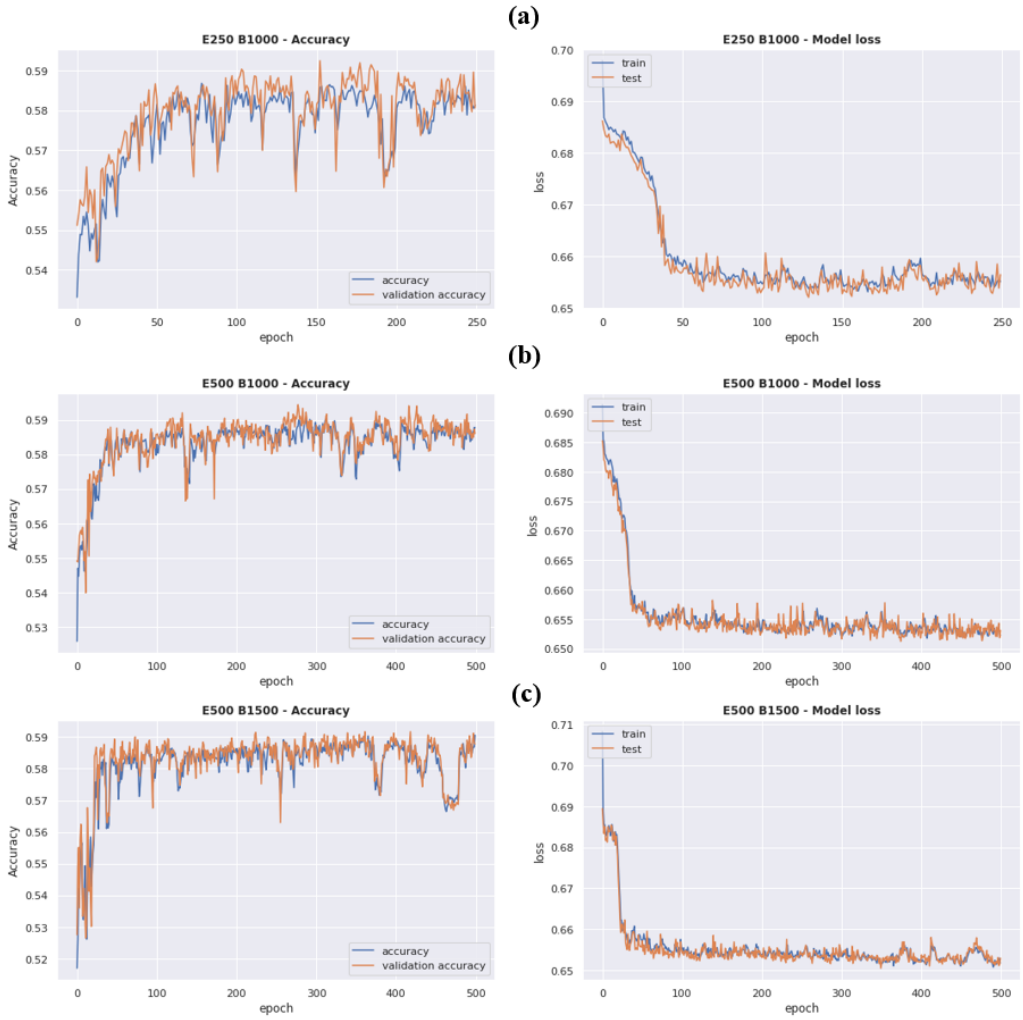

Figure 9. Visualization curves for MLP model accuracy and loss. The E2B1 at (**a**) shows a bumpy convergence but at the cost of overfitting. Both (**b**,**c**) indicate relatively stable learning convergence with higher accuracy and lower loss.

4.5. Impact of Multisource Data on the Prediction Performance

Apart from training and validating ONSP on integrated data discussed above, the proposed model is also evaluated to investigate the relationship of occupancy with pedestrian volume, traffic flow, and weather conditions resulting in eight possible combinations, as shown in Table 10. These combinations include occupancy (O), occupancy_weather (O_W), occupancy_pedestrian (O_P), occupancy_traffic (O_T), occupancy_weather_pedestrian (O_W_P), occupancy_weather_traffic (O_W_T), occupancy_pedestrian_traffic (O_P_T) and occupancy_weather_pedestrian_traffic (O_W_P_T). The performance in term of accuracy is observed as 71%, 79%, 67%, 60%, 80%, 79%, 62%, and 80%, respectively using 6 folds validation. The results indicate that the overall training accuracy of the model has increased when the occupancy data has integrated with traffic, followed by pedestrians and weather.

Interestingly, a higher performance is observed with the data segment mapped for all occupancy streets and weather, followed by pedestrians and traffic with common street mapping. For the pedestrian and traffic datasets, the performance varies with street mapping. For example, the pedestrian data sharing 20 common streets with the occupancy data resulted in an accuracy of 67% in O_P datasets, and traffic data sharing seven streets with occupancy data resulted in an accuracy of 60% in the O_T combination. The presence of weather data in any combination has improved the accuracy to 80%, 79%, and 80% in O_W_P, O_W_T, and O_W_P_T combinations. Performance could have been even higher

with more street data available for pedestrian and traffic segments since no common streets were identified between these two in the dataset used for this study.

Table 10. Training and testing accuracy of ONSP model on different combinations of the datasets.

Datasets	Training (CV = 6)	Testing
Occupancy (O)	72%	71%
Occupancy Weather (O_W)	79%	80%
Occupancy Pedestrian (O_P)	67%	67%
Occupancy Traffic (O_T)	61%	62%
Occupancy Weather Pedestrian (O_W_P)	80%	81%
Occupancy Weather Traffic (O_W_T)	79%	80%
Occupancy Pedestrian Traffic (O_P_T)	62%	62%
Occupancy Weather Pedestrian Traffic (O_W_P_T)	80%	81%

4.6. Prediction Using Stream Analytics

To scale the proposed solution for a big data environment, real-time analytics was performed. Previously in [25], the real-time prediction was implementable at a small scale, and due to missing location features, drivers spent more time searching for a parking space.

Hence, this study has fulfilled these gaps for on-street parking prediction using the ONSP for live predictions on IBM Watson, making the prediction system scalable for a big data environment. Also, the prediction is based on the precise location features (Side_Of_Street, Area, O_BetweenStreet1, and O_BetweenStreet2) that help reduce search time. The ONSP predicts a sample size of approximately 100 instances. A dataset of this size is used because of the limited cloud storage available in Lite Plan, but the whole procedure is scalable for large data using advanced plans.

The streaming flow is shown in Figure 10, where the first operator with the label 'Simulating' generates the data streams. The second operator with the label 'Python Model' is the trained ONSP model, which is used for predicting incoming tuples. The predicted outcomes can be accessed from the 'Debug,' the third operator. The analytics performance is evaluated using two performance metrics: ingest rate and throughput. Ingest rate shows the number of events submitted to streamflow per second. From Figure 10, we can see that 0.1–0.3 events are received by the sample data operator per second (shown in the blue line). The throughput rate measures the flow of predicted events, which is equivalent to the ingest rate (shown in the green line). Both graphs show a steady flow of events suggesting that instant predictions occur as soon as the events are ingested.

Figure 10. IBM Streaming Analytics—Visualizing the stream's input rate using ingest flow graph and the flow using the throughput graph for on-street parking predictions.

The live streaming was performed for 30 min with a constant rate of 0.1–0.3 events per second, resulting in 360 events (see Table 11). The total number of events shown in Figure 11 is 365, approximately what is calculated in Table 11.

Table 11. Streams flow EPS rate in 30 min timeframe.

EPS	No. of Minutes	No. of Seconds	Total Seconds	Total Events
0.1	15	15 × 60	900	0.1 × 900 = 90
0.3	15	15 × 60	900	0.3 × 900 = 270

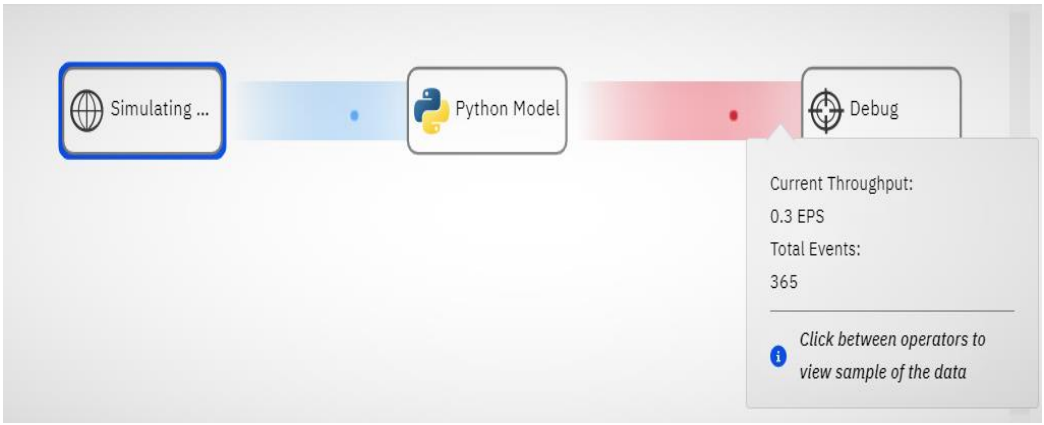

Figure 11. The total number of events at a throughput rate of 0.3 EPS.

The process of on-street parking prediction is summarized with a use case, 'next best option strategy,' shown in Figure 12. In Use Case—1, Alex predicts occupancy through a user request. On submitting a request, the feature values are ingested by the ONSP. Although the data is simulated in the use case, data generated from an IoT infrastructure can be ingested precisely in the same way. The ONSP is thus predicting the occupancy for Alex using data. In analyzing the prediction based on the observed results shown in Table 9, with a rate of 0.1, 0.2, and 0.3 EPS, 10, 5, and 3 events can be predicted in a 1-min duration using IBM Lite Plan (https://cloud.ibm.com/docs/Db2onCloud?topic=Db2onCloud-free_plan, accessed on 1 January 2022).

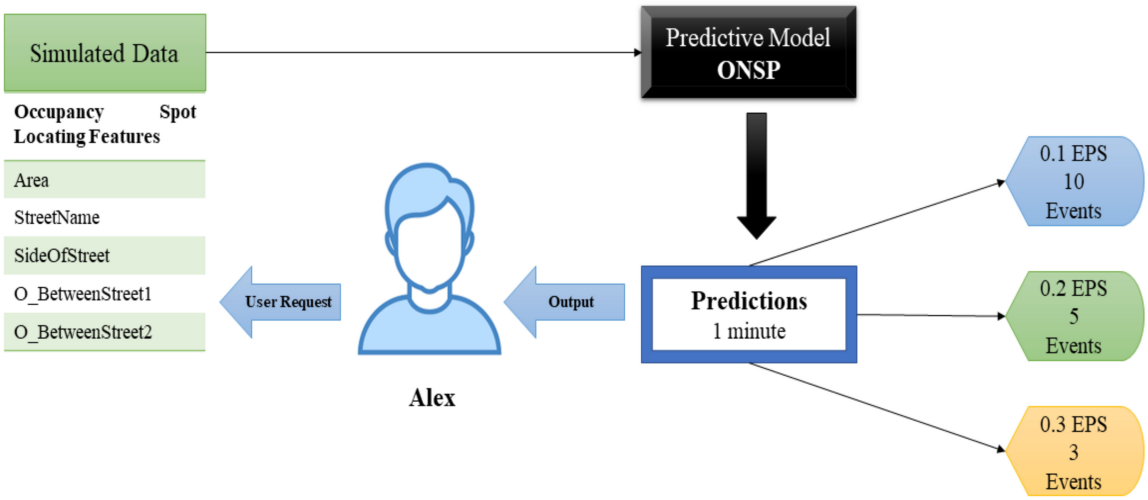

Figure 12. Use Case—1: User predicting on-street parking spot in using next best options strategy.

5. Future Implications

The prediction for on-street parking spots depends on the two types of models. The first is training the model on historical data, which is adopted in this study. As shown

in Figure 13, the second model is the most used for live predictions using basic stream operations. In the future, both model types can be integrated based on how advanced a parking spot prediction is needed. To predict a parking spot within 30 min, the ONSP model can select the best possible options based on the weather, traffic, and pedestrian load. On the other hand, the basic stream operations can be performed in predicting spots within a time span of 5–10 min.

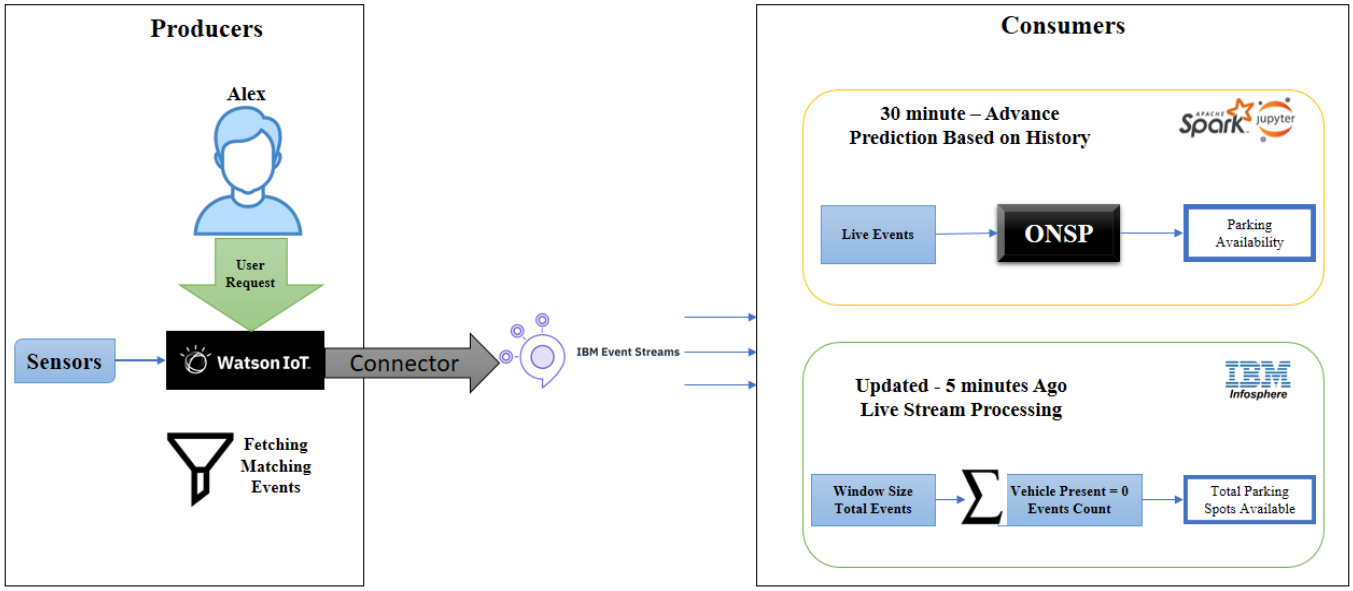

Figure 13. Use Case—2: User predicting on-street parking spots; a predictive modeling approach based on time frames such as 15 min and 30 min.

In the live prediction based on the window size, for instance, 5 min, all the number of records with 'vehicle presence' with a value equal to zero can be considered as available spots. If the live data is updated after every five minutes, the information of available spots will generate based on the buffer size; for example, in the last five minutes, the available parking spots were 6 or 8 based on the location. The processing of live streams can be performed in the IBM Infosphere (https://www.ibm.com/docs/en/streams/4.1.1?topic=welcome-introduction-infosphere-streams, accessed on 1 January 2022) environment. Use Case—2 demonstrates an end-to-end pipeline as a future perspective to fetch the data from sensors in predictions by utilizing the available services and our research work.

In the first step, the relevant records will be filtered based on the user requirements using the Watson IoT Platform. The filtered records will then be sent via Kafka as input to the next step. In the third step, depending on the size of the time window, the relevant model can be utilized for making live predictions.

6. Conclusions

This paper presents an end–end ML pipeline for parking prediction, which we plan to integrate into a smart parking application. The application will help drivers find the nearest parking spot in advance. We integrated contextual factors such as weather conditions, traffic flow, and pedestrian volume with occupancy data to determine parking space reliably and accurately. At first, feature engineering is performed to identify the predictive capabilities of different features from each dataset. Then several ML/DL algorithms were trained and tested on the different combinations of the historical dataset to evaluate if better results could be produced for the parking space availability prediction problem by using less complex algorithms. From the results of the comparative analysis, we found that Random Forest is the optimal solution for the parking space availability problem with an accuracy of 81%, and Decision Tree was the close second best model with 77.7% accuracy. These two models consistently outperformed one of the computationally complex algorithms

(Multilayer Perceptron). We selected Random Forest for subsequent analysis and real-time deployment for the On-Street parking prediction problem (ONSP).

The results indicate that contextual features have improved the prediction accuracy by 10% compared to the models only tested on basic occupancy data. The most significant data points from the contextual features are wind direction, humidity, temperature, and pressure from the weather dataset, parking duration and location from the occupancy dataset, and hourly count and time of the day from the pedestrian dataset. We did not observe any significant impact on the traffic dataset; the reason is the limited traffic information available for occupancy data compared to weather and pedestrian. However, even with the smaller data size, the feature traffic count appeared to be the second most influential variable after parking duration, showing a tight coupling of traffic flow with parking space availability. The ONSP is deployed for real-time predictions using stream processing services in the next step. The simulation results showed an ingest rate of 0.1 and a throughput of 0.3 events per second, demonstrating a fast and reliable prediction approach for available slots within a time frame of 5–10 min.

Results prove the worth of the proposed prediction pipeline over existing techniques in predicting parking availability in time frames (e.g., 15 min, 30 min) by considering multiple contextual factors. The study has a limitation in terms of the datasets used. We used multiple datasets for the year 2017, and since then, there has been a significant update on multiple data sources, especially in terms of traffic flow in the City of Melbourne. Providing the updated data associated with multiple sources might impact the results. However, it is worth mentioning that this study aims to identify the predictive capabilities of underlying features from each data source used in the study so that future research will consider the influence of such variables on parking prediction. Another limitation is the real-time prediction of simulated data; however, the data generated by the IoT sensors and devices can be ingested by the ONSP model precisely in the same way. Hence, combining real-time data with ML approaches can be a useful step toward sustainable parking solutions.

In the future, we intend to explore more contextual features of the research study, such as special events around, to understand how temporal points of interest can impact prediction accuracy [32]. We will leverage the historical dataset to more recent data and evaluate more deep learning techniques (LSTM, GRU, etc.). Finally, we plan to develop a web-based application for the end-users to show parking space availability in a time window of 15, 30 min, and so on.

Author Contributions: Conceptualization, S.I. and A.M.; methodology, S.I., S.K. and M.A.; software, S.I.; validation, S.I., S.K. and A.M.; resources, M.A. and A.M.; data curation, S.I., N.N. and M.A.; writing—original draft preparation, S.I.; writing—review and editing, S.K.; visualization, S.I., M.A. and N.N.; supervision, A.M.; project administration, A.M.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Saba Riaz from SZABIST Islamabad and Asim Munir from International Islamic University Islamabad for their time and knowledge to help understand the statistical dimensions that proved very useful in carrying out the research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdeen, M.; Nemer, I.; Sheltami, T. A Balanced Algorithm for In-City Parking Allocation: A Case Study of Al Madinah City. *Sensors* **2021**, *21*, 3148. [CrossRef] [PubMed]
2. 4 Charts Explain Greenhouse Gas Emissions by Countries and Sectors. Available online: https://www.wri.org/insights/4-charts-explain-greenhouse-gas-emissions-countries-and-sectors (accessed on 13 March 2022).

3. Wong, M.S.; Wang, T.; Ho, H.C.; Kwok, C.Y.T.; Lu, K.; Abbas, S. Towards a Smart City: Development and Application of an Improved Integrated Environmental Monitoring System. *Sustainability* **2018**, *10*, 623. [CrossRef]
4. Syed, A.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 429–475. [CrossRef]
5. Kurek, A.; Macioszek, E. Impact of Parking Maneuvers on the Capacity of the Inlets of Intersections with Traffic Lights for Road Traffic Conditions in Poland. *Sustainability* **2021**, *14*, 432. [CrossRef]
6. CISCO. IoE-Driven Smart City Barcelona Initiative Cuts Water Bills, Boosts Parking Revenues, Creates Jobs & More; 2014. Available online: https://www.cisco.com/c/dam/m/en_us/ioe/public_sector/pdfs/jurisdictions/Barcelona_Jurisdiction_Profile_final.pdf (accessed on 3 March 2020).
7. Fadi, A.-T.; Arman, M. Smart parking in IoT-enabled cities: A survey. *Sustain. Cities Soc.* **2019**, *49*, 101608.
8. Liu, K.S.; Gao, J.; Wu, X.; Lin, S. On-Street Parking Guidance with Real-Time Sensing Data for Smart Cities. In Proceedings of the 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Hong Kong, China, 11–13 June 2018.
9. Rajabioun, T.; Ioannou, P.A. On-Street and Off-Street Parking Availability Prediction Using Multivariate Spatiotemporal Models. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2913–2924. [CrossRef]
10. Monteiro, F.V.; Ioannou, P. On-Street Parking Prediction Using Real-Time Data. In Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018.
11. Awan, F.M.; Saleem, Y.; Minerva, R.; Crespi, N. A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction. *Sensors* **2020**, *20*, 322. [CrossRef] [PubMed]
12. Alajali, W.; Wen, S.; Zhou, W. On-Street Car Parking Prediction in Smart City: A Multi-source Data Analysis in Sensor-Cloud Environment. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Guangzhou, China, 12–15 December 2017.
13. Zheng, Y.; Rajasegarar, S.; Leckie, C. Parking availability prediction for sensor-enabled car parks in smart cities. In Proceedings of the 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 7–9 April 2015.
14. Jelen, G.; Podobnik, V.; Babic, J. Contextual prediction of parking spot availability: A step towards sustainable parking. *J. Clean. Prod.* **2021**, *312*, 127684. [CrossRef]
15. Matijosaitiene, I.; McDowald, A.; Juneja, V. Predicting Safe Parking Spaces: A Machine Learning Approach to Geospatial Urban and Crime Data. *Sustainability* **2019**, *11*, 2848. [CrossRef]
16. Thomas, D.; Kovoor, B.C. A Genetic Algorithm Approach to Autonomous Smart Vehicle Parking system. *Procedia Comput. Sci.* **2018**, *125*, 68–76. [CrossRef]
17. Claudio, B.; Paolo, N.; Irene, P. Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access* **2018**, *6*, 44059–44071.
18. Camero, A.; Toutouh, J.; Stolfi, D.H.; Alba, E. Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities. In Proceedings of the International Conference on Learning and Intelligent Optimization, Kalamata, Greece, 10–15 June 2018.
19. Kizilkaya, B.; Caglar, M.; Al-Turjman, F.; Ever, E. Binary search tree based hierarchical placement algorithm for IoT based smart parking applications. *Internet Things* **2018**, *5*, 71–83. [CrossRef]
20. Horng, G.-J. The Adaptive Recommendation Mechanism for Distributed Parking Service in Smart City. *Wirel. Pers. Commun.* **2014**, *80*, 395–413. [CrossRef]
21. Rehena, Z.; Mondal, M.A.; Janssen, M. A Multiple-Criteria Algorithm for Smart Parking: Making Fair and Preferred Parking Reservations in Smart Cities. In Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, Delft, The Netherlands, 30 May–1 June 2018.
22. Raj, S.U.; Manikanta, M.V.; Harsitha, P.S.S.; Leo, M.J. Vacant Parking Lot Detection System Using Random Forest Classification. In Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019.
23. Stolfi, D.H.; Alba, E.; Yao, X. Predicting Car Park Occupancy Rates in Smart Cities. In Proceedings of the International Conference on Smart Cities, Malaga, Spain, 14–16 June 2017; Springer: Berlin/Heidelberg, Germany, 2017.
24. Klandev, I.; Tolevska, M.; Mishev, K.; Trajanov, D. Parking Availability Prediction Using Traffic Data Services. In Proceedings of the IEEE Proceedings of the ICT Innovations, Online, 24–26 September 2020.
25. Vlahogianni, E.I.; Kepaptsoglou, K.; Tsetsos, V.; Karlaftis, M.G. A Real-Time Parking Prediction System for Smart Cities. *J. Intell. Transp. Syst.* **2015**, *20*, 192–204. [CrossRef]
26. Shimizu, I. *Multistage Sampling;* Statistics Reference, Published Online: Wiley. 2014. Available online: https://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat05705 (accessed on 15 April 2020).
27. Labeeuw, W.; Deconinck, G. Customer Sampling in a Smart Grid Pilot. In Proceedings of the 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012.
28. Breiman, L. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
29. Nguyen, C.; Wang, Y.; Nguyen, H.N. Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *J. Biomed. Sci. Eng.* **2013**, *6*, 551–560. [CrossRef]

30. Brownlee, J. How to Use ROC Curves and Precision-Recall Curves for Classification in Python. Available online: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (accessed on 20 November 2021).
31. Shiekh, R. Cross Validation Explained: Evaluating Estimator Performance. Available online: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85 (accessed on 20 November 2021).
32. Pevec, D.; Babic, J.; Kayser, M.A.; Carvalho, A.; Ghiassi-Farrokhfal, Y.; Podobnik, V. A data-driven statistical approach for extending electric vehicle charging infrastructure. *Int. J. Energy Res.* **2018**, *42*, 3102–3120. [CrossRef]