Predicting Parking Occupancy via Machine Learning in the Web of Things

Jesper C. Provoost
Dept. of Computer Science
University of Twente, Enschede, The Netherlands
j.c.provoost@student.utwente.nl

Andreas Kamilaris
Dept. of Computer Science University of Twente, Enschede, The Netherlands
Research Centre on Interactive Media Smart Systems and Emerging Technologies (RISE)
Nicosia, Cyprus
a.kamilaris@utwente.nl

Luc J.J. Wismans
Centre of Transport Studies
University of Twente, Enschede, The Netherlands
l.j.j.wismans@utwente.nl

Sander J. van der Drift
DAT.Mobility, Deventer, The Netherlands
svddrift@dat.nl

Maurice van Keulen
Dept. of Computer Science
University of Twente, Enschede, The Netherlands
m.vankeulen@utwente.nl

Abstract—The Web of Things (WoT) enables information gathered by sensors deployed in urban environments to be easily shared utilizing open Web standards and semantic technologies, creating easier integration with other Web-based information, towards advanced knowledge. Besides WoT, an essential aspect of understanding dynamic urban systems is artificial intelligence (AI). Via AI, data produced by WoT-enabled sensory observations can be analyzed and transformed into meaningful information, which describes and predicts current and future situations in time and space. This paper examines the impact of WoT and AI in smart cities, considering a real-world problem, the one of predicting parking availability. Traffic cameras are used as WoT sensors, together with weather forecasting Web services. Machine learning (ML) is employed for AI analysis, using predictive models based on neural networks and random forests. The performance of the ML models for the prediction of parking occupancy is better than the state of the art work in the problem under study, scoring an MSE of 7.18 at a time horizon of 60 minutes.

Index Terms—Internet of Things; Web of Things; Parking Occupancy; Machine Learning; Neural Networks.

I. INTRODUCTION

The Internet of Things (IoT) enables embedded sensors to become easily deployed in urban areas for monitoring and surveillance of the ambient environment [1]. These sensors are capable of measuring environmental conditions and phenomena with high precision, such as temperature, humidity, radiation, electromagnetism, noise, chemicals, air quality etc. They can also measure urban infrastructure characteristics, such as traffic and flows of pedestrians. These devices, when deployed to urban environments, could provide useful data about the environmental context of the area under study [2]. IoT also involves local and global infrastructures enabling advanced services by interconnecting physical things together based on existing and evolving interoperable information and communication technologies such as cloud computing, cloud storage and other existing Internet technologies. IoT allows devices to communicate through the IP protocol, especially its IPv6 version, which is designed for billions of Internet-connected objects [3]. Internet connection permits updating information of things in real-time [4], storing/processing data on the cloud and taking advantage of Internet protocols for security, authentication, data integrity, message routing etc. [5], [6].

While IoT ensures connectivity and interoperability at the lower layers of the ISO stack (i.e. physical, data, networking and transport layer), the Web of Things (WoT) enables sensor devices to interact and communicate at the higher layers of the ISO stack (i.e. application and presentation layers) [7], [8]. WoT is about approaches, software architectural styles and programming patterns that allow real-world objects to be part of the World Wide Web. Through WoT, sensors start operating as tiny Web servers, being able to expose their capabilities as Web services [9]. This blending of Web-based and device-based services facilitate the development of
physical mashups [10]. Reusing the existing, successful and well-known standards of the Web allows to make any physical object part of the WoT, therefore directly addressable and usable using popular tools. Hence, the information gathered by the sensing capabilities of things can be shared employing open Web standards and semantic technologies, creating easier integration with other Web-based information, towards advanced knowledge [11].

Thus, people who have access to this knowledge may make more informed decisions during their everyday lives (e.g. smarter commuting) [12], while policy-makers will be able to develop wiser policies about urban systems and infrastructures (e.g. construction of new roads, management of parking areas, incentives to people to use public transportation etc.) [13], [14].

An important ingredient in the recipe of web intelligence for understanding dynamic urban systems and infrastructures, besides the real-time monitoring capabilities of WoT, is artificial intelligence (AI). Via AI, the data produced by WoT-enabled sensory observations can be analyzed and transformed to meaningful information which describes and predicts current and future situations in time and space [15]. Examples of AI, in this context, include machine learning, decision-making support and advanced knowledge representations.

This paper examines the impact of WoT and AI in urban systems, considering an existing, pragmatic, real-world problem, the one of predicting parking availability using parking occupancy rate. This rate is defined as the percentage of occupancy at any given time of some parking lot. For example, if the occupancy rate is 80% at a parking lot that has a capacity of 100 parking spaces, this means that 80 cars are parked there at some given moment. Traffic cameras are used as WoT sensors, accompanied by Web services giving real-time information about weather conditions. Besides, machine learning (ML) is employed for AI analysis, while the prediction of parking occupancy in a time horizon of 60 minutes would be the application under study.

This scenario (i.e. predicting parking occupancy) has been selected due to its importance in a smart city context. U.S. drivers spend an average of 17 hours searching for a parking spot every year [16], [17]. This amount is even higher in the U.K. and Germany with 44 and 41 hours per year, respectively. In Germany alone, the average driver wastes £896 per year on the hunt for a parking space. This amount aggregates to a yearly burden of €40.4 billion on the German economy. Furthermore, a survey of 17,968 drivers from 30 cities shows that 64% of participants experience stress while trying to find parking. Drivers that possess information on parking availability are 45% more successful in their decisions than those without knowledge of this information when arriving at their parking facility [18]. The statistics mentioned above indicate the importance of tackling this problem effectively and constitute a big motivation for this study.

The contributions of this paper are the following:

- It is yet another demonstration of how the WoT can be effectively combined with AI for tackling urban-related problems [15] (see Section III).

- It solves the problem of predicting parking occupancy by means of machine learning with high accuracy, comparable and better than the state of the art work in the field (see Sections II and V-A).

- It analyzes the impact of predictive variables involved, aiming to give some hints over the significance of each variable in the prediction outcome (see Section IV-D).

- From existing literature, it is the first paper that employs random forests for parking occupancy prediction. We note that it has been used in a similar problem, that of predicting occupancy trends in the bicycle service stations of Barcelona [19].

In this paper, we managed to obtain a Mean Square Error (MSE) of 7.18 for our prediction model in a 60-minutes prediction horizon. Also, we performed a comparison with state of art research work and results, to demonstrate the importance of our findings.

The rest of the paper is organized as follows: Section II describes related work in predicting parking availability and occupancy, while Section III presents the problem under study and the methodology involved. Then, Section IV shows the performance of ML-based models used, towards predicting parking occupancy, while Section V comments on the overall findings and proposes future work. Finally, Section VI concludes the paper.

II. RELATED WORK

This paper is situated in the following research areas:

- Applications of WoT.
- Prediction of parking occupancy via ML techniques.

Regarding applications of WoT, numerous demonstrations and proof-of-concept implementations have been published in various domains, such as smart homes and buildings, urban environments and smart cities, the smart grid of electricity, e-health and remote health services, mobile computing, smart agriculture etc. [20], [21].

Some applications of WoT target particularly urban computing [11], [22], urban planning [2], journey planning considering traffic conditions [23], design of WoT-enabled parking lots [24], as well as real-time information about available parking places via on-street parking sensors [25], [26].

Concerning ML-based techniques for predicting parking occupancy, the most important ones1 are listed in Table I. The table lists 12 relevant papers, the specific goal of prediction/estimation, the metric used for assessing accuracy, the performance under this metric, the ML-based method employed, as well as the environment under which the study was performed (i.e. simulation vs real-world).

As Table I shows, various goals have been set by the authors, such as estimating occupancy of parking spaces in general (more popular goal) and/or parking lots in particular, predicting waiting times for a parking slot, as well as estimating probabilities for free spaces and slots. All papers target parking spaces and lots, except from [27], which focuses on

1Here importance has been defined based on the overall accuracy recorded, the scale of the project in terms of volume and heterogeneity of input data, as well as the number of citations.
bicycle slots. Our work relates mainly with the research works estimating occupancy of parking spaces [13], [28]–[32].

The fifth column of Table I lists the ML method used to create the prediction model. A wide variety of different methods have been used, with Neural Networks and Recurrent Neural Networks (RNN) being the ones mostly used. Moreover, the vast majority of related work employed sensors and data from real-world deployments and not based on simulations.

Moreover, various metrics have been selected by the authors of related work for evaluating performance, such as deviation of predictions from actual availability, MSE, Mean Absolute Error (MAE), Root MSE (RMSE), Mean Absolute Scaled Error (MASE) and the coefficient of determination ($R^2$). MAE and MSE have been the most popular ones used (for their definition, see Section III-B). It is worth mentioning that each paper uses different sources of input data, summarized in Table II.

A discussion about the performance of the ML-based models of related work according to the different types of input data and metrics used, (also) in comparison to our work, is provided in Section V. In Section V, we additionally try to identify which predictive variables (i.e. from the ones listed in the first row of Table II) seem to be the most important.

III. PROBLEM DESCRIPTION AND METHODOLOGY

Cities around the world are becoming smarter, equipped with various emerging technologies (e.g. sensors, actuators, remote sensing, aerial photography, computer vision etc.) to become more dynamic and flexible in terms of addressing in near real-time the challenges that appear continuously due to over-population, crowds, the occurrence of disasters etc. Issues such as parking, traffic management and smart transportation are among the topics where intelligence affects smart services when AI is combined with IoT/WoT towards faster and more informed decision-making. In this context of smart and digital cities, real-time data collection and analysis are crucial factors.

In an ideal scenario, all sensors located inside smart cities are fully enabled according to the principles of WoT [7]–[9]. In this scenario, sensors expose their features as RESTful Web services using description languages (such as WADL) to describe these services, employing Semantic Web Technologies (i.e. ontologies, vocabularies and query languages) to describe data produced towards seamless interoperability with third-party machines and with humans. In parallel, Web servers with high processing/storage capabilities can interact with these WoT-enabled sensors in real-time, either in a client-server or a publish-subscribe model. They would be equipped with a range of AI tools and software, offering advanced predictive analytics in real-time.

This paper embodies some aspects of the vision mentioned above, focusing on the challenge of predicting parking occupancy. The area under study is the city of Arnhem, the Netherlands. The targeted parking area is the central and largest one in the city, located near the central train station. Section III-A below describes the data sources used, while Section III-B presents the ML-based models selected for the prediction of the occupancy rate of this parking area. The general architecture of the problem under study is illustrated in Figure 1.

A. Data Collection

To be able to make as accurate predictions as possible, considering the observations from related work, both historical and real-time data needs to be considered. Regarding historical data, the Open Parkeerdata portal of the Municipality of Arnhem provides historical transaction data (i.e. when car drivers pay for parking), which can be easily transformed for deriving historical occupancy rates. The data source provides transaction data of three parking garages in Arnhem, of which the Centraal Garage has been selected due to being the largest one in the city centre. Data was used from August 2017 to April 2019.

Regarding historical occupancy data, a look-back window of 60 minutes was defined. This window consists of a range of inputs which provide the model with knowledge about the occupancy rates during the previous 60 minutes, before the prediction attempt.

Traffic data (i.e. flow of number of vehicles per hour) was gathered from the Nationale Databank Wegverkeersgegevens (NDW) using its Dexter platform (historical data) as well as the Open Data Service of NDW (real-time feed). In total, eight measurement locations were selected, all situated at the orbital highways and freeways around Arnhem, specifically on highway exits and access roads (see Figure 1 for their locations). After considering the availability and validity of the sensors involved, traffic flow data from November 2017-April 2019 was selected.

The open databases of the Dutch meteorological institute KNMI (historical weather data) and the Weerlive API (for the real-time feeds) were also considered. Using an online Web service, the hourly data of several weather-related variables were queried. The measurements of the closest weather station were chosen (i.e. the Deelen station, 10 km from the city centre). The data source provides the air temperature (i.e. at 1.5-meter height) and rainfall (i.e. a binary variable denoting whether rain has fallen in the past hour) variables at a 10-minute interval. The hourly data from August 2017 to April 2019 was used.

Holiday and event data were manually gathered from a variety of sources. The website of the Dutch government was used to retrieve the dates of national holidays and school holiday periods, both historically and in the future (i.e. this is used in the real-time application). Subsequently, the dates of events were retrieved from the event calendar of the Arnhem tourist office as well as the calendar of the Gelredome

\[\text{Gemeente Arnhem, Open Data: https://parkeerdata.nl/opendata/arnhem/}\]
\[\text{NDW Dexter: https://dexter.ndwcloud.nu/}\]
\[\text{KNMI, Uurgegevens van het weer in Nederland: https://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi}\]
\[\text{Weerlive.nl: http://weerlive.nl/delen.php}\]
\[\text{Rijksoverheid: https://www.rijksoverheid.nl/onderwerpen/schoolvakanties}\]
\[\text{Visit Arnhem: https://www.visitarnhem.com/evenementen}\]
### TABLE I: Precision results as presented in related work.

| Paper | Goal | Metric | Performance | Method | Environment |
|-------|------|--------|-------------|--------|-------------|
| [33]  | Estimate the occupancy state of parking lots | Relative deviation of predictions | 8% error for 25-minutes prediction | Queuing theory and a continuous-time homogeneous Markov model | Simulation |
| [34]  | Locate a suitable parking spot | N/A | N/A | Multi-hop wireless parking meter network | Simulation |
| [35]  | Predict block-level parking occupancy | MAE | 0.878 (time window not defined) | Convolutional Neural Network (CNN) and Stacked LSTM Autoencoder | Real-World |
| [36]  | Predict block-level parking occupancy | MAE | 1.69 (30 minutes in advance) | Graph CNN and LSTM | Real-World |
| [26]  | Probability of a free space to continue being free in subsequent time intervals, and the short-term parking occupancy prediction | MAE | 0.14 - 0.32 up to 30 minutes ahead | Weibull parametric models, genetically optimized multi-layer perceptrons | Real-World |
| [18]  | Predict real-time parking space availability | Differences between forecast and actual availability, MAE | Differences: less than 3% in the case of 1 hour, MAE: 0.12 - 5.06 | Real-time availability forecast (RAF) algorithm | Real-World |
| [13]  | Predict real-time parking space availability | MAE, $R^2$ | MASE is 1.37-2.72, $R^2$ between 0.895 and 0.975 for up to 60 minutes ahead | Bayesian Regularized Artificial Neural Network | Real-World |
| [37]  | Predict real-time parking space availability | | Correctly identifying all the fully-occupied bays | Queueing theory model based on a transient probability model | Real-World |
| [32]  | Estimate number of available parking spots | MAE, RMSE | RMSE: 60.84, MAE: 25.28 | General regression neural network (GRNN) | Real-World |
| [31]  | Estimate number of available parking spots | RMSE | 5.42 | LSTM and BPNN techniques | Real-World |
| [38]  | Estimate the probability for a free parking space depending on a certain cell in the grid (location in the city) | MSE | 0.16321 | Neural Network | Real-World |
| [27]  | Predict availability of bicycle slots, prediction of waiting times | RMSE | 3.12 for medium-term prediction (30-60 minutes) | Generalized Additive Models (GAM) | Real-World |
| [28]  | Predict occupancy rate | MAE | 6.7 - 10.2 | Recurrent neural networks (RNN) | Real-World |
| [29]  | Predict occupancy rate | MSE | 50-500 (much variation) | Comparison of several machine learning techniques | Real-World |
| [30]  | Predict occupancy rate | MAE, $R^2$ | (For 15 min. ahead) MAE: 1.3 - 15.4, $R^2$: 0.257 - 0.986 | Regression Tree, Support Vector Regression (SVR), Neural Network | Real-World |
| This paper | Predict occupancy rate | MSE, MAE | (For 60 min. ahead, NN) MSE: 7.18, MAE: 1.87 (For 60 min. ahead, RF) MSE: 7.98, MAE: 1.92 | Neural Network (NN), Random Forest (RF) | Real-World |

### TABLE II: Data sources in related work.

| Paper | Time of day | Day of week | Occupancy look-back window | Traffic flow | Holiday | Event | Weather | Location | Parking sensor |
|-------|-------------|-------------|---------------------------|-------------|---------|-------|---------|----------|---------------|
| [33]  | X           | X           | X                         | X           |         |       |         |          |               |
| [34]  | X           | X           | X                         |             |         |       |         | X        |               |
| [35]  | X           | X           | X                         |             |         |       |         |          |               |
| [36]  | X           | X           | X                         |             |         |       |         | X        |               |
| [26]  | X           | X           | X                         |             |         |       |         | X        |               |
| [18]  | X           | X           | X                         |             |         |       |         |          | X             |
| [32]  | X           | X           | X                         |             |         |       |         | X        |               |
| [31]  | X           | X           | X                         |             |         |       |         | X        |               |
| [38]  | X           | X           | X                         |             |         |       |         | X        |               |
| [27]  | X           | X           | X                         |             |         |       |         | X        |               |
| [28]  | X           | X           | X                         |             |         |       |         | X        |               |
| [29]  | X           | X           | X                         |             |         |       |         | X        |               |
| [30]  | X           | X           | X                         |             |         |       |         | X        |               |
| This paper | X        | X           | X                         |             |         |       |         | X        |               |
Fig. 1: General architecture: Area under study, data sources used and infrastructure.

stadium\textsuperscript{9}, considering that highly crowded sports matches and concerts are frequently organized there.

All data were resampled to fit in one-minute intervals, while a 2\textsuperscript{nd} order low-pass Butterworth filter (with a cut-off frequency of 0.05) was applied for smoothing. All the data used in this paper, mapped to predictive variables, are listed in Table II.

B. Prediction Models

The AI aspect of the paper is based on two well-known ML techniques: the neural networks (NN) and random forests (RF). Neural networks are multi-layer networks of neurons which can be used to classify or predict one or more output variables, based on a series of inputs. Random forests serve the same goals (i.e. classification or regression) but have a different structure, as they are mainly an ensemble of decision trees. NN have been widely used in related work (see Table I), while RF constitute a powerful method for performing regression, not used in related work. Furthermore, the decision was made to also incorporate a deep learning variant of the regular NN, namely the convolutional neural network (CNN) into the comparison. As prediction horizon, we decided to predict up to 60 minutes ahead in time, considering that related work has demonstrated predictions for 15-60 minutes.

The data processing and model development tasks were executed on a desktop computer with commodity hardware. This machine consists of an Intel i5-3570k CPU, an AMD R9 280X GPU and 16 GB of RAM. The Python libraries Scikit-learn, Keras and TensorFlow were used for the implementation of the machine learning models.

For training and testing the NN, CNN and RF models, the complete dataset (see Section III-A was divided into training (72%), validation (8%) and testing (20%). Given the sequential nature of the input data (i.e. time series), we maintained the chronological order of the input data, to test the model’s sensitivity to seasonal patterns during the validation and testing phase.

The NN, CNN and RF had various hyperparameters that required tuning\textsuperscript{[39]} for optimization of the three models under study. To perform this tuning, the relevant hyperparameters of the models were systematically tweaked, followed by repeated evaluation of the model performance on the validation set [40]. We acknowledge the existence of automated ML hyperparameter tuning software and tools [41], which can facilitate this task; however, we preferred to do this process manually.

These hyperparameters involved (among others) the number of hidden layers and number of neurons of the NN, as well as the learning rate of the NN/CNN. For RF, the hyperparameters involved the number of trees in the forest, maximum tree depth and the maximum number of features. During hyperparameter tuning, the NN and CNN were trained for 200 epochs till convergence. During the final training round, a total of 2000 epochs was used for the NN and CNN. To prevent overfitting, a model checkpoint was applied to save the model’s parameters

\textsuperscript{9}Gelredome: https://gelredome.nl/nl/evenementen
at the epoch where the lowest validation loss was measured.

Regarding the performance metrics used, MSE, MAE and $R^2$ have been selected. With $n$ being the number of data samples, $y$ being the observed value and $\hat{y}$ being the predicted value, the MSE can be defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y})^2$$

Similarly, the MAE can be defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y - \hat{y}|$$

With $\hat{y}$ being the mean value of $y$, the $R^2$ metric can be defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y - \hat{y})^2}{\sum_{i=1}^{n} (y - \bar{y})^2}$$

A combination of the MSE, MAE and $R^2$ metrics facilitates a direct comparison with relevant work [28]–[30] (see Section V).

IV. RESULTS

This section lists the results of the experiments performed after tuning the NN, CNN and RF models used for predicting parking occupancy rates. Section IV-B describes the performance of the models in terms of prediction, based on the MSE metric, while Section IV-D tries to identify the variables that served as the best predictors, via the technique of feature elimination.

A. Parameter Tuning

![Fig. 2: Results of grid search regarding number of neurons and number of hidden layers of the NN.](image)

1) NN: First step was to decide the NN’s architecture, i.e. number of layers and total neurons. Based on a grid search (see Figure 2), it is shown that the worst performance is found at the bottom right of the figure where a low number of neurons is divided over a large number of layers, leading to under-fitting of the model. The best performing models are located around the top-left and middle-left areas of the heatmap of Figure 2. This suggests that the NN performs best with an architecture in which a high number of neurons is divided between a relatively small number of hidden layers. The minimum MSE was observed at configuration (90, 4), i.e. with 90 neurons spread across 4 hidden layers.

Next, the hyperparameter of the learning rate was optimized, as shown in Figure 3. For all learning rates $\alpha$, the progression of MSE over time (i.e. the number of epochs) was visualized using a line graph. As expected, higher learning rates initially produce a rapid decrease of MSE, but then the model shows an unbalanced behaviour, stuck in local minima. On the contrary, lower learning rates demonstrate a stable decrease in the error but converge too slowly. According to the previously defined criteria, the learning rate $\alpha = 0.0001$ provides an optimal balance: after 200 epochs, the corresponding MSE error is the lowest, with a descending trend.

Using these optimized parameters, the final NN was trained over the course of 2000 epochs.

2) CNN: For the CNN, the previously determined architecture of the NN was adopted. Hence, the network also consists of 90 neurons spread across 4 hidden layers. However, two convolutional layers were added before the hidden layers: an 8x8 kernel for the traffic inputs and a 4x4 kernel for the look-back window. The traffic inputs were reshaped such that for every timestep, a two-dimensional array is obtained. The first dimension is then the look-back time, while the second dimension is the traffic sensor. The locations were ordered by their distance to the garage, such that any spatial correlations can be recognized by the model using the convolution process.

3) RF: The results suggest that the MSE is subject to exponential decay when the number of trees $n$ increases (see Figure 4). When there is only one tree in the ensemble, the
RF can essentially be regarded as an ordinary decision tree. The real power of the RF becomes evident when the number of trees grows. Around $n = 50$, the MSE seems to reach a plateau state. A higher number of trees are ineffective: no significant performance gain occurs any more, while the computational complexity rises dramatically.

Afterwards, based on a grid search concerning the hyperparameters $\text{maximum features}$ and $\text{maximum depth of the trees}$, it becomes clear that all three configurations of Figure 5 follow the same trend towards the maximum depth of the tree $d$. In the case where the maximum features equal the available number of features, the error decreases faster and reaches the plateau state at a significantly lower value of maximum depth $d$. Thus, this is the preferred option, keeping the maximum depth small to minimize the computational complexity (i.e., training and prediction times). Using this configuration, the minimum MSE is reached at a maximum depth of $d = 12$, after which no further gain in performance takes place.

The values selected for the hyperparameters above after optimization are summarized in Table III.

| Model   | Hyperparameter          | Value |
|---------|-------------------------|-------|
| NN      | Hidden Layers           | 4     |
| NN      | Neurons at each Layer   | 22-24 (90 neurons in total in a 24-22-22-22 structure) |
| NN      | Learning Rate $\alpha$  | 0.0001 |
| CNN     | Hidden Layers           | 4     |
| CNN     | Convolutional Layers    | 2     |
| CNN     | Neurons at each Layer   | 22-24 (90 neurons in total in a 24-22-22-22 structure) |
| CNN     | Kernel size             | 4x4 (occup. look-back window) and 8x8 (traffic flows) |
| NN      | Learning Rate $\alpha$  | 0.0001 |
| RF      | Number of Trees         | 50    |
| RF      | Maximum Tree Depth      | 12    |
| RF      | Maximum Number of Features | 42 (equals the number of features) |

B. Predictive Performance

The performance of the NN, CNN and RF models in terms of predicting the occupancy rate of the central parking of the city of Arnhem, the Netherlands is depicted in Figure 6. The figure shows the MSE value of the three models in different scenarios of increasing prediction horizon, from 5 minutes ($MSE = 0.14$ for the NN case) up to 60 minutes ($MSE = 7.18$ for the NN case). As expected, there is an exponential increase of the MSE as the prediction horizon increases. The results also demonstrate that the CNN performs better than the RF up to a horizon of 45 minutes, after which the RF performs slightly better. Furthermore, it can be observed that the regular NN performs better than the CNN on all predictive horizons. The results are also summarized in Table IV, considering various metrics and values of the predictive horizon.
TABLE IV: Summary of model performance.

| Metric     | NN     | CNN    | RF     | Predictive horizon |
|------------|--------|--------|--------|--------------------|
| MAE        | 0.65   | 0.72   | 0.71   | 15 min             |
|            | 1.13   | 1.19   | 1.16   | 30 min             |
|            | 1.55   | 1.71   | 1.56   | 45 min             |
|            | 1.91   | 2.20   | 1.92   | 60 min             |
| MSE        | 0.83†  | 0.92†  | 1.00   | 15 min             |
|            | 2.50†  | 2.63†  | 2.79†  | 30 min             |
|            | 4.69   | 5.18   | 5.18   | 45 min             |
|            | 7.18   | 8.27   | 7.98   | 60 min             |
| RMSE       | 0.93†  | 0.96†  | 1.00   | 15 min             |
|            | 1.58   | 1.62   | 1.67   | 30 min             |
|            | 2.17   | 2.28   | 2.28   | 45 min             |
|            | 2.68   | 2.88   | 2.82   | 60 min             |
| R²         | 0.99√  | 0.99√  | 0.99√  | 15 min             |
|            | 0.996  | 0.996  | 0.996  | 30 min             |
|            | 0.993  | 0.992  | 0.992  | 45 min             |
|            | 0.989  | 0.987  | 0.988  | 60 min             |

C. Model Efficiency

The models were also compared by their training time, as well as the computation time needed to generate a set of predictions. This gives an indication of the efficiency of each model type. Both the NN and CNN were trained for 2000 epochs, after which both models started to overfit slightly. For the NN, the overall training process took 49,320 seconds (approximately 14 hours). The CNN took slightly longer to train, namely 53,400 seconds (approximately 15 hours). The RF, however, took notably shorter to train than both other models, namely 1,211 seconds (approximately 20 minutes). It should, therefore, remain a careful consideration whether the shorter training times of the RF outweigh the performance increase of the NN.

After 100 prediction attempts, the mean prediction time of the feed-forward NN was 1.57 seconds. Using the same approach, the mean prediction time of the RF was slightly lower, i.e. 1.32 seconds. These results indicate that the RF is slightly more time-efficient than the NN. Yet, the differences are not substantial (i.e. only a fraction of a second) and it is therefore unlikely that the difference would be noticeable within a real-time predictive system. Presumably, the small differences in prediction time are caused by the relatively high number of paths which must be traversed through all 90 nodes in the NN case, as compared to the 50 trees of the RF.

Figure 7 shows how the MSE of the NN model changes in relation to the size of the training dataset (considering the 60-minute prediction horizon case). For example, by training the NN model with 25% of the initial dataset (as described in Section III-A), the penalty in performance is rather small ($MSE = 10.89$), compared to the performance when training with the complete dataset ($MSE = 7.18$). A remarkable plateau of the MSE is visible between the $\frac{1}{4}th$ and $\frac{1}{2}th$ fractions of the total dataset. Since the data is a continuous time series and therefore chronologically ordered, a potential reason for this abnormality could be the fact that the left-neighbouring fraction contains more relevant information than the right-neighbouring fraction, i.e. a regular workweek. The results of this test demonstrate that a good model can still be trained on a relatively small dataset, i.e. $\frac{1}{4}$ of the original, which corresponds to approximately 12 days of data. This suggests that our approach is robust and transferable towards other parking facilities where less data is available. Moreover, the model could converge faster and generalize better on the test set. However, it remains important to note that the MSE values start growing exponentially for further reductions of the initial dataset, so one should pay careful attention when considering a smaller dataset: when available, a large dataset should always be preferred over a smaller one.

D. Predictive Variables

It is important also to consider which of the prediction variables under study (see Table II and Section III-A) are the ones predicting better the parking occupancy rate. For this purpose, we analysed the input variable dependency using a feature elimination strategy, examining the impact of the remaining training data. Feature elimination entails that variables are categorically removed from the input dataset. For every variable (or category of variables) that is removed, the model is trained with the remaining inputs. The performance of the model is then compared with the performance of the original reference model. Since NN showed better results (see Section IV-B), it was selected for this exercise. The results of the feature elimination exercise are shown in Figure 8.

V. DISCUSSION

As mentioned during the previous sections, this paper tackles the problem of estimating parking occupancy in real-time, as this estimation could affect traffic management policies and citizen behaviour. However, the methodology and approach, by combining AI and IoT/WoT, could be well applied in various other problems of urban areas, such as crime prevention, disaster management and response, optimized administration of electricity infrastructures, efficient use of renewable energy etc. [15].

Even though the differences are not significant, the results demonstrate that the NN model outperforms the RF one, in all different time values of the prediction horizon (see
Our model also performs better than [28], [29], [31], [32] by a significant margin. The approach of [28], which only predicts in a 60-minutes horizon, based on a testing process section with some caution. The same observation holds for [32], in which an MAE of 25.28 was calculated. In the case of [31], an RMSE value of 52.89 was found, which is higher than the $RMSE = 2.68$ of our approach, when predicting for a 60-minutes horizon. Moreover, the work in [29] claimed MSE values between $50 - 500$, with high variation between different tests and scenarios. Our approach is significantly more accurate at this comparison too, having a $MSE = 7.18$ when predicting the furthest time step ahead, i.e. 60 minutes.

It is likely that these substantial differences are caused by the fact that [28], [29], [31], [32] are all lacking a real-time look-back window of the last-known occupancy rate measurements, which seems to be an essential predictor variable (see Section IV-D and Figure 8). Hence, it is very likely that the real-time component increases dramatically the ability of a model to predict occupancy rate with higher accuracy.

Regarding [30], the lowest MAE was determined to be 1.3 when predicting in a 15-minutes horizon, using a regression tree approach. Our approach has demonstrated an $MAE = 0.65$ when predicting at the same horizon. Moreover, the work in [30] shows a $R^2$ value of 0.986 when predicting in 15-minutes horizon, while our approach demonstrates a value of $R^2 = 0.999$. Hence, even though the differences are not large, our approach performs better than [30].

Summing up, it is evident that our approach presents better results than the state of the art work in the field, by a small margin. However, we cannot draw strong conclusions about the best-performing ML models, since the prediction performance depends on the input data (see Figure 8), while related work employs different techniques and algorithms, training datasets and variables (see Table II).

### B. Factors Influencing Prediction

Comparing with related work (see Table II), our work has been one of the most complete in terms of covering the spectrum of possible input data towards solving the parking prediction problem. Most papers considered historical occupancy data (i.e. a look-back window), time of day and day of the week as important predictive variables. Traffic flows and real-time parking sensors have also been used, while weather information, holidays or events happening, as well as the location of the parking space inside some urban area have been considered too, with less popularity. The fact that each paper employs different types and volumes of input data makes comparisons difficult and perhaps unfair.

We can safely assume that the more data (and data sources) available, the more accurate the predictive models can be. This is the case of this paper, taking into account our comparisons with related work that employs (mostly) fewer data sources to build predictive models (see Section V-A). From related work (see Table II), the work in [35] uses as many data sources as our paper, and the result is to produce remarkably high accuracy (i.e. $MAE = 0.878$). As the authors mention, “incorporating information about spots available, temperature and weather that potentially influence parking behaviour can significantly improve the performance of parking occupancy prediction”. Unfortunately, the prediction horizon window...
in [35] has not been specified; thus, we cannot make any comparisons, plus the prediction of parking occupancy is only at the block level.

An important research question is which variables are good predictors as input data. Figure 8 implies that the occupancy look-back window is the most important one, followed by real-time traffic flows (i.e. close to the parking space) and day of the week. Aspects of time of day and weather conditions (i.e. temperature, rain) seem to have a smaller impact in prediction accuracy, while events and holidays play only a minor role. This could be considered surprising, considering that events could have a tremendous impact on the traffic within the city. However, each event is characterized by entirely different patterns of crowd attendance at different times (e.g. a football match compared to an outdoor festival). Thus, it is not easy to encode the predictive potential of events, without integrating somehow to the model the context of the event performed (i.e. type, popularity, expected attendance, access to public transport etc.). Nonetheless, our observation here is that events and holiday variables do not contribute much as comprehensive predictors, at least without additional contextual information.

Related work proposes some additional attributes that might influence the decision-making of drivers for parking alternatives. These factors that could affect the drivers’ parking activity involve walking distance or distance to destination, driving and waiting time, parking fees, the service level of parking lots, safety, the average speed of passing vehicles from traffic points etc. [18], [35], [42], [43]. Particularly, the exact number of available parking spaces in real-time of the parking lot under study or nearby ones is an essential attribute in the driver’s parking decision-making process [18]. Integration of these additional factors as input to our predictive models constitutes the task of future work.

Finally, we stress the fact that in many countries, the real-time occupancy of parking lots in a given area is becoming available via real-time parking sensors (see Table II, papers [26], [31], [34]). This information should be crucial for predicting occupancy rates in the near term, as it is demonstrated mostly in [26]. Our work has shown better results than [31], but this could be because Li et al. do not use the real-time look-back window of the occupancy, which seems to be the most important predictive variable according to Figure 8.

C. Towards a Real-Time Web of Things

A quickly expanding ecosystem of Web-enabled sensors is evolving worldwide. Web technologies began to penetrate in these new generations of embedded devices, which are deployed massively in urban environments and intelligent city applications [44]. The WoT can be considered as a real-time application platform [12], [45], [46], which offers increased interoperability among heterogeneous sensors, datasets and applications. The principles of the WoT can be used to enable the vision of digital, real-time cities, where citizens and policymakers have fast access to relevant information. Combined with AI, digital cities can become smarter [14].

In this paper, we presented a demonstration of this perspective, investigating the problem of predicting the parking occupancy rate. We combined WoT-based resources, together with ML-based models, to address this problem. Although the ideal situation, described at the beginning of Section III, involves RESTful Web services, Semantic Web technologies and real-time APIs, the reality is in many cases different. Historical transaction data was only available as text files, which we needed to parse manually. The real-time feed for the traffic flows did not follow a REST architecture; thus, it was not trivial to use it. Also, the traffic flow data was not accompanied by semantic metadata description; hence it was challenging to understand it at the beginning. The same occurred for the online Web service for weather information, although in this case information was easier to understand. Still, semantic information would help to describe better weather information, e.g. which weather stations were involved, where they were placed (e.g. under shade or exposed to the sun), accuracy rates etc.

An important aspect to discuss is whether a prediction horizon of 60 minutes is enough for affecting traffic and for solving the parking problem. In a world of real-time IoT-based sensory information, we argue that 60 minutes is sufficient time for affecting traffic, e.g. by adapting traffic lights, giving incentives to drivers to take alternative routes etc. For the parking problem, 60 minutes are more than enough, since the prediction outcome could be integrated into existing route planning user applications. For example, when the user requires 30 minutes to reach some area, the application can search for nearby parking lots considering the predicted occupancy rates of these lots in the next 30 minutes.

Summing up, our demonstration indicates that there are still problems and challenges towards the digitization of smart cities and the use of open data for better understanding the urban landscapes. The principles of WoT would be crucial in this context, to better understand, reuse and integrate heterogeneous hardware, software, services, data and applications together. This can lead to advanced reasoning, better knowledge representations, faster and more efficient big data analysis, more accurate prediction algorithms by combining heterogeneous datasets etc. [2]. Summing up, the WoT, together with AI, constitute significant elements for the realization of intelligent, sustainable cities that truly serve their citizens, ensuring high quality of living, safety, security, health and happiness [14], [15].

D. Future Work

For future work, our goal is to further improve the prediction models by understanding which other predictive variables might affect occupancy rates, studying more elaborately the real-life scenarios where our prediction performance was lower. This might relate to some of the factors mentioned in Section V-B.

Furthermore, we intend to integrate our prediction model to some existing route planning user application (see Section V-C), examining how the prediction of parking occupancy rates works together with route planning applications for giving the best service possible to drivers in terms of finding a parking place.

Finally, it would be interesting to see whether the model could be generalized for other cities, urban infrastructures and
VI. CONCLUSION

This paper addressed a real-world problem, namely the one of predicting parking availability. Traffic and parking sensors were used as Web of Things sensor nodes, together with weather forecasting Web Services, accompanied by a look-back window of historical occupancy rates of the parking space under study.

Several machine learning techniques were used to solve the problem via the development of prediction models based on neural networks and random forests. The performance of the ML models for the prediction of parking occupancy was better than the state of the art related work in the problem under study, scoring a mean squared error (MSE) of 7.18 in a time horizon of 60 minutes. The historical occupancy rate (i.e. the look-back window) was the most important predictive variable, followed by traffic flows measured at the orbital highways around the city of Arnhem, The Netherlands.

This paper constitutes yet another demonstration of how the WoT can be combined with Artificial Intelligence to approach and tackle actual problems of cities and urban environments.

ACKNOWLEDGEMENTS

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 739578 complemented by the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

REFERENCES

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. Computer networks, 54(15):2787–2805, 2010.
[2] M Mazhar Rathore, Aawais Ahmad, Anand Paul, and Seungmin Rho. Urban planning and building smart cities based on the internet of things using big data analytics. Computer Networks, 101:63–80, 2016.
[3] Antonio J Jara, Miguel A Zamora, and Antonio Skarmeta. Glowlab IP: An adaptive and transparent IPv6 integration in the Internet of Things. Mobile Information Systems, 8(3):177–197, 2012.
[4] Cristina Turcu, Cornel Turcu, and Vasile Gaitan. Integrating robots into the Internet of Things. International Journal of Circuits, Systems and Signal Processing, 6(6):430–437, 2012.
[5] William Stallings. Computer networking with Internet protocols and technology. Pearson/Prentice Hall Upper Saddle River, NJ, USA, 2004.
[6] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianji Liu. Security in the internet of things: a review. In Computer Science and Electronics Engineering (ICCSEE), International Conference on, volume 3, pages 648–651. IEEE, 2012.
[7] Dominique Guinand, Vlad Mihai Trifa, and Erik Wilde. Architecting a mashable open world wide web of things. Technical report/Swiss Federal Institute of Technology Zurich, Department of Computer Science, 663, 2010.
[8] Erik Wilde. Putting things to REST. UC Berkeley: School of Information, 2007. Retrieved from https://escholarship.org/uc/item/1786t1dm.
[9] Adam Dunkels et al. Efficient application integration in IP-based sensor networks. In Proceedings of the First Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pages 43–48. ACM, 2009.
[10] Dominique Guinand, Vlad Trifa, Thomas Pham, and Olivier Liechti. Towards physical mashups in the web of things. In Networked Sensing Systems (INSS), Sixth International Conference on, pages 1–4. IEEE, 2009.
[11] Andreas Kamilaris, Andreas Pittsilides, Francesc X. Prenafeta-Boldu, and Muhammad Intizar Al. A Web of Things Based Eco-System for Urban Computing - Towards Smarter Cities. In Proc. of the 24th International Conference on Telecommunications (ICT), Limassol, Cyprus, May 2017.
[12] Andreas Kamilaris and Andreas Pittsilides. The Impact of Remote Sensing on the Everyday Lives of Mobile Users in Urban Areas. In 7th International Conference on Mobile Computing and Ubiquitous Networking (ICMU), Singapore, January 2014.
[13] Claudio Badili, Paolo Nesi, and Irene Paoli. Predicting available parking slots on critical and regular services by exploiting a range of open data. IEEE Access, 6:44059–44071, 2018.
[14] Michael Batty, Kay W Aitken, Rosalba Giannotti, Alexei Prozdnikhouk, and复制条目中缺失的参考文献。
[15] Mehdi Mohammadi and Ala Al-Fuqaha. Enabling cognitive smart cities using big data and machine learning: Approaches and challenges. IEEE Communications Magazine, 56(2):94–101, 2018.
[16] Graham Cookson and Bob Pishue. The Impact of Parking Pain in the US, UK and Germany. H2o INRIX Research. Online verfügbar unter http://inrix.com/research/parking-pain, zuletzt geprüft am, 21.2018, 2017.
[17] Donald C Shoup. Cruising for parking. Transport Policy, 13(6):479–486, 2006.
[18] Felix Caicedo, Carola Blazquez, and Pablo Miranda. Prediction of parking space availability in real time. Expert Systems with Applications, 39(8):7281–7290, 2012.
[19] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. Predicting occupancy trends in Barcelona’s bicycle service stations using open data. In 2015 SAI Intelligent Systems Conference (IntelliSys), pages 439–445. IEEE, 2015.
[20] Andreas Kamilaris and Andreas Pittsilides. Mobile Phone Computing and the Internet of Things: A Survey. IEEE Internet of Things Journal, 3(6):885–899, December 2016.
[21] Dezhe Zeng, Song Guo, and Zixue Cheng. The web of things: A survey. JCM, 6(6):424–438, 2011.
[22] Ralf Tonjes, Septimiu Nechifor, et al. Real time IoT stream processing and large-scale data analytics for smart city applications. In European Conference on Networks and Communications, Poster Session, 2014.
[23] Dan Puia, Stefan Bischof, Bogdan Serbanescu, Septimiu Nechifor, Jossiane Parreira, and Herwig Schreiner. A public transportation journey planner enabled by IoT data analytics. In 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), pages 355–359. IEEE, 2017.
[24] Sujith Samuel Mathew, Yacine Atif, Quan Z Sheng, and Zakaria Maamar. Building sustainable parking lots with the Web of Things. Personal and ubiquitous computing, 18(4):895–907, 2014.
[25] Jorge Lanza, Luis Sánchez, Verónica Gutiérrez, José Galache, Juan Santana, Pablo Sotres, and Luis Muñoz. Smart city services over a future internet platform based on internet of things and cloud: The smart parking case. Energies, 9(9):719, 2016.
[26] Eleni Vlachogianni, Konstantinos Kepaptsoglou, Vassileios Tsotsos, and Matthew G Karlaftis. A real-time parking prediction system for smart cities. Journal of Intelligent Transportation Systems, 20(2):192–204, 2016.
[27] Bei Chen, Fabio Pinelli, Mathieu Sinn, Adi Botea, and Francesco Calabrese. Uncertainty in urban mobility: Predicting waiting times for shared bicycles and parking lots. In 16th International Conference on Intelligent Transportation Systems (ITSC), pages 53–58. IEEE, 2013.
[28] Andrés Camero, Jamal Toutouh, Daniel H Stoltz, and Enrique Alba. Evolutionary deep learning for car park occupancy prediction in smart cities. In International Conference on Learning and Intelligent Optimization, pages 386–401. Springer, 2018.
[29] Daniel H Stoltz, Enrique Alba, and Xin Yao. Predicting car park occupancy rates in smart cities. In International Conference on Smart Cities, pages 107–117. Springer, 2017.
[30] Yanxrzh Zheng, Sutharsan Rajasegarar, and Christopher Leckie. Parking availability prediction for sensor-enabled car parks in smart cities. In Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pages 1–6. IEEE, 2015.
[31] Jianzhong Li, Jiming Li, and Wenzhe Zhang. Deep Learning Based Parking Prediction on Cloud Platform. In 4th International Conference on Big Data Computing and Communications (BIGCOM), pages 132–137. IEEE, 2018.
[32] Tayo Fabusuyi, Robert C Hampshire, Victoria A Hill, and Katsunobu Sasunuma. Decision analytics for parking availability in downtown pittsburgh. *Interfaces*, 44(3):286–299, 2014.

[33] Murat Caliskan, Andreas Barthels, Bjorn Scheuermann, and Martin Mauve. Predicting parking lot occupancy in vehicular ad hoc networks. In *65th Vehicular Technology Conference (VTC)*, pages 277–281. IEEE, 2007.

[34] Prithwish Basu and Thomas DC Little. Wireless ad hoc discovery of parking meters. In *MobiSys Workshop on Applications of Mobile Embedded Systems (WAMES)*, 2004.

[35] Soumya Suvra Ghosal, Abderrahman Bani, Amine Amrouss, and Issmail El Hallaoui. A deep learning approach to predict parking occupancy using cluster augmented learning method. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 581–586. IEEE, 2019.

[36] Shuguang Yang, Wei Ma, Xidong Pi, and Sean Qian. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies*, 107:248–265, 2019.

[37] Jiaqi Ma, Erin Clausing, and Yimin Liu. Smart on-street parking system to predict parking occupancy and provide a routing strategy using cloud-based analytics. Technical report, SAE Technical Paper, 2017.

[38] Christoph Pflügler, Thomas Köhn, Maximilian Schreieck, Manuel Wiesche, and Helmut Kremer. Predicting the availability of parking spaces with publicly available data. *Informatik 2016*, 2016.

[39] John D Kelleher, Brian Mac Namee, and Aoife D’arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT Press, 2015.

[40] Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.

[41] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-WEKA: Automated selection and hyper-parameter optimization of classification algorithms. *CoRR*, abs/1208.3719, 2012.

[42] Shi An, Bo Han, and Jian Wang. Study of the mode of real-time and dynamic parking guidance and information systems based on fuzzy clustering analysis. In *Proceedings of the International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, volume 5, pages 2790–2794. IEEE, 2004.

[43] William HK Lam, Zhi-Chun Li, Hai-Jun Huang, and SC Wong. Modeling time-dependent travel choice problems in road networks with multiple user classes and multiple parking facilities. *Transportation Research Part B: Methodological*, 40(5):368–395, 2006.

[44] Andreas Kamilaris, Nicolas Iannarilli, Vlad Trifa, and Andreas Pitsilides. Bridging the Mobile Web and the Web of Things in Urban Environments. In *First International Workshop the Urban Internet of Things (Urban IOT 2010)*, Tokyo, Japan, November 2011.

[45] Dennis Pfisterer, Kay Romer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, Henning Hasemann, Alexander Kröller, Max Pagel, Manfred Hauswirth, et al. SPITFIRE: Toward a Semantic Web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[46] Dan Puiu, Payam Barnaghi, Ralf Tönjes, Daniel Kümpner, Muhammad Intizar Ali, Alessandra Mileo, Josiane Xavier Parreira, Marten Fischer, Sefki Kolozali, Nazli Farajidavar, et al. CityPulse: Large scale data analytics framework for smart cities. *IEEE Access*, 4:1086–1108, 2016.