Endüstri Mühendisliği / Journal of Industrial Engineering

https://dergipark.org.tr/tr/pub/endustrimuhendisligi

PREDICTION OF PARKING SPACE AVAILABILITY USING ARIMA AND NEURAL NETWORKS

Aslı SEBATLI-SAGLAM[1*], Fatih CAVDUR[2]

[1]Mudanya University, Faculty of Engineering, Architecture and Design, Department of Industrial Engineering, Mudanya, Bursa
ORCID No: https://orcid.org/0000-0002-9445-6740
[2]Bursa Uludag University, Faculty of Engineering, Department of Industrial Engineering, Nilufer, Bursa
ORCID No: https://orcid.org/0000-0001-8054-5606

Keywords

Parking prediction, Time series forecasting, ARIMA, SARIMA, Neural networks

Abstract

It may be critical for drivers to have information about the occupancy rates of the parking spaces around their destination in order to reduce the traffic density, a non-negligible part of which caused by the trips to find an available parking space. In this study, we predict parking occupancy rates (and thus, space availability) using three different techniques: (i) auto-regressive integrated moving average model, (ii) seasonal auto-regressive integrated moving average model and (iii) neural networks. In the implementation phase, we use the data set of the on-street parking spaces of the well-known "SFpark" project carried out in San Francisco. We take into account not only the past occupancy rates of parking spaces, but also exogenous variables that affect the corresponding occupancy rates as day type and time period of the day. We make predictions with different model structures of each of the considered methods for each parking space with different parking occupancy patterns in the data set and then compare the results to find the best model design for each

*Sorumlu yazar; e-posta: aslisebatli.saglam@mudanya.edu.tr

doi : https://doi.org/10.46465/endustrimuhendisligi.1241453

parking space. We also, evaluate the results in terms of the superiority of the methods over each other and note that the performance of neural networks is better than those of the other approaches in terms of the mean squared errors.

ARIMA VE SİNİR AĞLARINI KULLANARAK PARK YERİ UYGUNLUĞUNUN TAHMİNİ

Anahtar Kelimeler

Park tahmini, Zaman serisi tahmini, ARIMA, SARIMA, Sinir ağları

Öz

Sürücülerin park yerlerinin doluluk oranları hakkında bilgi sahibi olmaları, park yeri bulmak için yapılan yolculuklardan kaynaklanan trafik yoğunluğunun azaltılması açısından kritik olabilir. Bu çalışmada, üç farklı teknik kullanarak park doluluk oranları (ve dolayısıyla yer uygunluğu) tahmin edilmektedir: (i) otoregresif entegre hareketli ortalama modeli, (ii) mevsimsel otoregresif entegre hareketli ortalama modeli ve (iii) sinir ağları. Uygulama aşamasında, San Francisco'da gerçekleştirilen "SFpark" projesinin yol üstü park yerlerinin veri seti kullanılmıştır. Park yerlerinin yalnızca geçmiş doluluk oranları değil, bu doluluk oranlarını etkileyen gün tipi ve saat dilimi olmak üzere dışsal değişkenler de dikkate alınmıştır. Veri setindeki farklı park doluluk paternlerine sahip her bir park yeri için ele alınan yöntemlerin her birinin farklı model yapıları ile tahminler yapılmış ve ardından her bir park yeri için en iyi model tasarımını bulmak için sonuçlar karşılaştırılmıştır. Buna ek olarak, yöntemlerin birbirine olan üstünlüğü açısından da sonuçlar değerlendirilmiş ve ortalama karesel hatalar cinsinden sinir ağlarının performansının diğer yaklaşımlardan daha iyi olduğu gözlenmiştir.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 24.01.2023	Submission Date	: 24.01.2023
Kabul Tarihi	: 08.04.2023	Accepted Date	: 08.04.2023

1. Introduction

It is estimated that approximately 80% of the rapidly increasing world population will live in cities by 2050 (Chang and Kalawsky, 2017). With this increase in population and urbanization, the number of vehicles participating in urban traffic also rises causing a non-negligible part of the traffic density. Moreover, studies show that 30% of the traffic density is caused by trips to find

an available parking space, and this rate rises up to 50% during rush hours (Enriquez, Soria, Alvarez-Garcia, Velasco and Deniz, 2017; Jin, Wang, Shu, Feng and Xu, 2012; Lin, Rivano and Le Mouel, 2017). Drivers spend an average of 8.1 minutes finding an available parking space (Shoup, 2006). Cookson (2017) reports that a driver spends an average of 17 hours a year searching for a vacant parking space, and the duration goes up to 107 hours in big cities such as New York. A study shows that this causes 4.2 billion more hours of travel a year, equivalent to traveling around the world 38 times, and 2.9 billion gallons of extra fuel consumption (Schrank and Lomax, 2007; White, 2007). The time drivers spend in traffic to find an available parking space not only causes traffic jams, but also increases fuel consumption and carbon emissions. Moreover, the speed of the vehicles decreases while searching for an available parking space, which also significantly increases the environmental pollution. Carbon monoxide and nitrogen hydride emissions of a vehicle traveling at 20 kph (12.43 mph) increase by more than 50% compared to a vehicle traveling at 50 kph (31.07 mph) (Teng, Falcocchio, Lapp, Price, Prassas and Kolsal, 2001). Therefore, the problem of finding a suitable parking space is important for drivers in terms of both time and fuel consumption as well as it turns into a critical problem for society and the environment considering the resulted traffic congestion and environmental pollution. It may thus be critical for the drivers to have information about the occupancy rates of the parking spaces around their destinations before their arrivals. Here it comes the importance of parking prediction, and we can say that an accurate prediction model may significantly reduce the time it takes to search for an available parking space by influencing drivers' route decisions. In other words, predicting future available parking capacities when drivers depart, when they arrive at their destinations or some time in between will allow them to have information about the occupancy rates of the parks near the destinations, and thus making it possible to follow a more accurate route from their departure points to the candidate parking lots.

In this study, we make parking predictions using three different machine learning techniques: (i) Auto-Regressive Integrated Moving Average (ARIMA) models, (ii) Seasonal Auto-Regressive Integrated Moving Average (SARIMA) models and (iii) neural networks. In the implementation phase, we use the data set of the on-street parking spaces of the well-known “SFpark” project carried out in San Francisco. We make predictions with different model structures of these methods for each parking space in the data set and compare the results to find the best model design of each method for each parking space with different parking occupancy patterns.

The rest of the article is organized as follows: In the second section, park prediction studies in the literature are summarized. Details about the dataset and methods are presented in the third section. Implementation details and results are given in the fourth section whereas the final remarks are presented in the last section.

2. Literature Review

Prediction of the availability of the parking spaces is one of the most important factors in solving the traffic problems caused by the parking process. When the problem is considered individually, it gains importance for drivers in terms of time loss and fuel consumption directly affecting their route decisions. On the other hand, for the society and the environment, it has critical importance also as it affects the urban traffic density and traffic-related carbon emissions, and thus environmental pollution. In this section, we present a summary of park prediction studies in the literature.

As seen in Table 1, various machine learning techniques are frequently used in order to predict parking space availability in the literature. In addition to them, some other analytical models such as queuing theory and Markov decision processes are also used. We can say that especially neural networks and deep learning-based studies come to the fore here. Vlahogianni, Kepaptsoglou, Tsetsos and Karlaftis (2016), make two types of prediction, the first one is the prediction of the parking occupancy rate (in percent) for a selected area with a multi-layer perceptron structure based on historical data. The second prediction method proposed in the study considers the probability of an empty parking space continuing to be empty (in terms of time) via duration models developed for different areas. Pflugler, Kohn, Schreieck, Wiesche and Krcmar (2016), predict the availability of parking spaces via a neural network for different categories of the location, time, weather condition, traffic and event data. Tiedemann, Vögele, Krell, Metzen and Kirchner (2015) present a neural gas clustering-based hybrid method for the parking prediction. Li, Li and Zhang (2018) present a deep learning-based prediction system and they use a long-short term memory network structure in order to predict parking availability. They use some exogenous data such as time of day, weather condition and holiday as well as historical parking occupancy data. Camero, Toutouh, Stolfi and Alba (2018) use a deep learning technique with recurrent neural networks and predict the parking occupancy rate. They also optimize the network architecture with two different evolutionary algorithms. Arjona Linares, Casanovas-Garcia and Vazquez (2020) predict parking availability with different recurrent neural network structures, as long-short term memory and gated recurrent unit. They also include weather and calendar data to their approach in order to improve the prediction performance. Yang, Ma, Pi and Qian (2019) propose a solution approach that combines graph-convolutional neural networks and long-short term memory in order to model both spatial and temporal patterns of parking spaces considering not only the parking meter transactions data but also traffic and weather data.

There are also studies in the literature using other common machine learning techniques for parking prediction. Rajabioun and Ioannou (2015) predict parking availability with an autoregressive model, taking into account temporal and spatial relationships. Tamrazian, Qian, and Rajagopal (2015) propose

different machine learning algorithms in order to work with offline and online data, *k*-means clustering and *k*-nearest neighbor algorithm, respectively. They observe that the patterns of the offline model and the online model are different. Ziat, Leroy, Baskiotis and Denoyer (2016) predict both traffic density and parking occupancy using a representation learning based machine learning technique with the motivation of drivers looking for available parking spaces to cause a significant portion of the traffic density. Provoost, Kamilaris, Wismans, Van Der Drift and Van Keulen (2020) use both neural networks and random forests and examine the impact of web of things technology on parking availability prediction.

Table 1
Methods Used for Parking Prediction in the Recent Literature

Methods	Studies
ARIMA models	Badii et al. (2018), Kuhail et al. (2019), Zhao et al. (2020)
Clustering	Ionita et al. (2014), Richter et al. (2014), Stolfi et al. (2017), Stolfi et al. (2020), Tamrazian et al. (2015)
Decision trees	Awan et al. (2020), Fabusuyi et al. (2014), Ionita et al. (2014), Lu and Liao (2020), Zheng et al. (2015)
Deep learning	Arjona et al. (2020), Camero et al. (2018), Kuhail et al. (2019), Li et al. (2018), Yang et al. (2019)
Markov decision problems	Caliskan et al. (2007), Tilahun and Di Marzo Serugendo (2017)
Naïve Bayes	Fabusuyi et al. (2014), Lu and Liao (2020)
Neural networks	Awan et al. (2020), Badii et al. (2018), Balmer et al. (2021), Fabusuyi et al. (2014), Ionita et al. (2014), Pflugler et al. (2016), Provoost et al. (2020), Tiedemann et al. (2015), Vlahogianni et al. (2016), Zhao et al. (2020), Zheng et al. (2015)
Other machine learning techniques	Awan et al. (2020), Kuhail et al. (2019), Lu and Liao (2020), Rajabioun and Ioannou (2015), Stolfi et al. (2017), Stolfi et al. (2020), Tamrazian et al. (2015), Zhao et al. (2020), Ziat et al. (2016)
Queueing models	Tavafoghi et al. (2019), Xiao et al. (2018)
Random forests	Awan et al. (2020), Balmer et al. (2021), Lu and Liao (2020), Provoost et al. (2020)
Support vector machines	Badii et al. (2018), Zhao et al. (2020), Zheng et al. (2015)

In some studies, researchers analyze the results obtained using different machine learning techniques. (Awan, Saleem, Minerva and Crespi, 2020; Badii, Nesi and Paoli, 2018; Balmer, Weibel and Huang, 2021; Fabusuyi, Hampshire, Hill and Sasanuma, 2014; Kuhail, Boorlu, Padarthi and Rottinghaus, 2019; Stolfi, Alba and Yao, 2017; Stolfi, Alba and Yao, 2020; Zhao, Zhang and Zhang, 2020; Zheng, Rajasegarar and Leckie, 2015). Lu and Liao (2020), unlike other studies in the literature, combine parking prediction and parking space allocation. They first predict parking occupancy using different machine learning techniques, and then allocate parks with a matching-based strategy. Richter, Di Martino and Mattfeld (2014), on the other hand, present different data clustering strategies, noting that the required storage space for accurate prediction could decrease through clustering. In addition to predicting parking occupancy with classical machine learning techniques, Ionita, Pomp, Cochez, Meisen and Decker (2018), also focus on extracting clusters and similarity relationships for locations.

Parking prediction problem is considered as a queuing system or Markov decision problem in some studies. For example, Xiao, Lou and Frisby (2018) predict parking occupancy through a queuing model with their proposed two-stage solution approach where the model parameters and the parking occupancy are estimated in the first and second stages, respectively. Tavafoghi, Poolla and Varaiya (2019) develop two queuing model-based estimation methods in their study (i.e., a micro-level model and a macro-level model), and they present real-time probabilistic park occupancy predictions. In the study, each parking slot is handled separately with the micro-level model while at the macro-level, all parking slots are considered in an integrated manner. Caliskan, Barthels, Scheuermann and Mauve (2007) model each parking space as a queuing system and define a continuous-time homogeneous Markov chain in order to predict parking occupancy rates. In another study, Tilahun and Di Marzo Serugendo (2017) develop an agent-based solution approach and use dynamic and time-varying Markov chain to predict parking availability with the agents they created for all parks. These agents communicate with each other to predict parking availability in the neighborhood, and thus, not only the behavior of only a single park, but also those of the others around it are considered.

In this study, we make parking predictions using three different machine learning techniques: (i) ARIMA, (ii) SARIMA and (iii) neural networks. In the implementation phase, we use the data set of the on-street parking spaces of the well-known "SFpark" project carried out in San Francisco. We make predictions with different model parameters of these three methods for each parking space. We evaluate the results obtained with different parameters and find the best suitable model structure of each method for each parking space with different parking occupancy patterns. In other words, the main contribution of this study is that we propose three different machine learning methods for parking predictions by evaluating the results both in terms of the model structures of each method and the methods themselves by making predictions for the parking spaces with different behaviors. We also consider not only the historical

occupancy rates of the parking spaces, but also some exogenous variables that affect the occupancy as day type and time period of the day.

3. Methodology

Research and publication ethics were complied with in this study. There is no need for any ethical approval.

In this study, we predict the parking occupancy rate of a certain location at a certain time period (in terms of day type and time period) using different methods. In this context, the data set of the on-street parking spaces of the well-known "SFpark" project carried out in San Francisco is used. The dataset includes data from April 2011 to the end of July 2013 for both off-street parking garages and on-street parking spaces in 10 districts of San Francisco. There are 409 blocks (on-street parking spaces) in 117 streets. On-street parking space data is obtained via sensors located on the roads. The dataset contains data for each block and time period depending on whether the parking lot is occupied or vacant. In other words, occupancy data is available for the type of the day (weekday or weekend) and each timestamp (for a 24-hour time period). In this study, we first analyze the average occupancy rate patterns for each block and time period. As shown in Fig. 1, we then select 12 blocks with different occupancy rate trends as we think that the corresponding sample of 12 blocks is a comprehensive representation of the various patters of occupancy rates overall the dataset. There are 48 categories on the x-axis showing the day type-time period pairs where the first 24 categories represent the time periods of the weekdays, and the next 24 categories represent the 24 hours of the weekends whereas the y-axis shows the corresponding occupancy rates. As seen in the figure, we can say that different blocks have different occupancy rate trends. For example, the average occupancy rate of block 41103 is almost uniform and close to 100%. Block 20100 also has a uniform pattern, and its average occupancy rate is nearly 50%. On the other hand, the average occupancy rates of some blocks, such as 38003 and 52001, are highly irregular and almost 0% in some time periods. A typical daily pattern has low occupancy rates during the night and high ones during the day on both weekdays and weekends, such as the trends of block 61418 and 70102 with their respective higher and lower rates.

Table 2 contains information about the 12 blocks (parking spaces) we consider. Table includes the location information of the relevant parks and the total number of parking slots in these parking spaces. In the following subsections, we provide the methodological details about the ARIMA and neural network models.

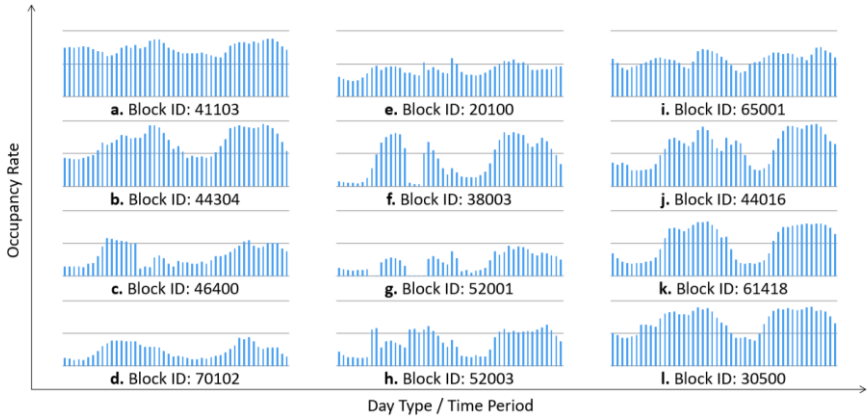

Fig. 1. Average Occupancy Rates of the Parking Spaces Considered

Table 2
Case Study

Parking Space	Block ID	Street	Block No.	District	# of Slots	Latitude	Longitude
P01	41103	Fell St	3	Civic Center	28	37.77597952	-122.4228170
P02	44304	Gough St	4	Civic Center	22	37.77722100	-122.4229967
P03	46400	Hayes St	0	Civic Center	10	37.77751740	-122.4163930
P04	70102	Van Ness Ave	2	Civic Center	11	37.77779374	-122.4193829
P05	20100	01st St	0	Downtown	39	37.79107566	-122.3991485
P06	38003	Davıs St	3	Downtown	11	37.79562207	-122.3980460
P07	52001	Kearny St	1	Downtown	12	37.78911427	-122.4037306
P08	52003	Kearny St	3	Downtown	11	37.79102933	-122.4039140
P09	65001	Sacramento St	1	Downtown	8	37.79479575	-122.3967070
P10	44016	Geary Blvd	16	Fillmore	12	37.78511116	-122.4303812
P11	61418	Post St	18	Fillmore	31	37.78557078	-122.4315510
P12	30500	Avıla St	0	Marina	4	37.80034419	-122.4403326

3.1 ARIMA Models

ARIMA method is first presented by Box and Jenkins (1976), and it is one of the most popular approaches for time series prediction. It is the integration of the Auto-Regressive (AR) model, which depends on the lagged values of the data, and the Moving Average (MA) Model, which depends on the error values of the previous predictions. The Integrated (I) expression here means that the past values are replaced by the difference between the current and previous values, which eliminates the changes in the level of the time series, and hence, removes the trend and seasonality effects. First-order differencing can be written as $y_t' =$

$y_t - y_{t-1}$ whereas second-order difference is $y_t'' = y_t - 2y_{t-1} + y_{t-2}$. ARIMA models are shown as $ARIMA(p, d, q)$ where p, d and q denote the degrees of AR model, difference and MA model, respectively. In Equation (1), the general form of the ARIMA model is given. Here, c is the intercept, $\varphi_1, \varphi_2, \dots, \varphi_p$ are parameters of the pth-level AR model, $\theta_1, \theta_2, \dots, \theta_q$ are parameters of the qth-level MA model, and $\varepsilon_t, \varepsilon_{t-1}, \dots, \varepsilon_{t-q}$ are the random error values. We can write it in backshift notation as shown in Equation (2). Note that $By_t = y_{t-1}$ where B is the backshift operator and it shift the data one period back. In general, dth-order difference can be written as $(1 - B)^d y_t$.

In the model structure used within the scope of this study, in addition to the classical ARIMA parameters, we also consider some exogenous variables as day type (x_1) and time period of the day (x_2), and hence, obtain the model in Equation (3) with the regression coefficients of β_1 and β_2.

Another method we use in this study is the SARIMA, which emerged by considering Seasonality (S) effect in the classical ARIMA model. SARIMA models can be shown as $SARIMA(p, d, q)(P_s, D_s, Q_s)_s$ where p, d and q denote the degrees of AR model, difference and MA model, respectively, and the lower-case (upper-case) letters represent the non-seasonal (seasonal) parameters while the seasonality period is represented by s. Corresponding model can thus be written as in Equation (4).

$$y_t' = c + \varphi_1 y_{t-1}' + \cdots + \varphi_p y_{t-p}' + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{1}$$

$$\left(1 - \varphi_1 B - \cdots - \varphi_p B^p\right)(1 - B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) + \varepsilon_t \tag{2}$$

$$\begin{aligned}\left(1 - \varphi_1 B - \cdots - \varphi_p B^p\right)&(1 - B)^d y_t \\ &= c + \beta_1 x_{1t} + \beta_2 x_{2t} + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t\end{aligned} \tag{3}$$

$$\begin{aligned}&\left(1 - \varphi_1 B - \cdots - \varphi_p B^p\right)(1 - \Phi_1 B^s - \cdots - \Phi_P B^{Ps})(1 - B)^d (1 - B^s)^D y_t \\ &= c + \beta_1 x_{1t} + \beta_2 x_{2t} \\ &+ \left(1 + \theta_1 B + \cdots + \theta_q B^q\right) + (1 + \Theta_1 B^s + \cdots + \Theta_Q B^{Qs})\varepsilon_t\end{aligned} \tag{4}$$

3.2 Neural Networks

Neural networks are extensively used as prediction models that can work with both categorical and time series data. In this study, we use the nonlinear auto-regressive network with exogenous inputs (NARX) as a recurrent dynamic network because of the feedback connections surrounding several layers of the network. The NARX network consists of a multi-layer perceptron which takes as input a window of past input and output values and computes the current output (Lin, Horne, Tino and Giles, 1996; Siegelmann, Horne and Giles, 1997). In

Equation (5), the general form of a neural network model is given, where y_t and x_t are time series data of the dependent output and independent input (exogenous) values, respectively. f function is a nonlinear function, whereas d_y and d_x are the number of target delays and input delays, respectively, where $d_y \geq 1$, $d_x \geq 1$ and $d_x \leq d_y$.

As seen in Fig. 2, we predict future values using a closed loop NARX structure which is also known as parallel architecture in the literature. First, we create an open-loop neural network which is also known as series-parallel architecture. Open-loop networks are trained using the time series of both inputs and targets (real values). In other words, the delayed target is used as an additional input. This structure generates output for the common time period with the input and target. This network structure is purely feed-forward, and it can be trained using the training algorithms for a multi-layer perceptron structure which contains three main layers as input, hidden and output layers. For the input $(y_{t-1}, y_{t-2}, \dots, y_{t-d_y}, x_{t-1}, x_{t-2}, \dots, x_{t-d_x})$ the output of the jth hidden neuron (z_j) is given in Equation (6). Here f_1 is the activation function, which is generally nonlinear, w_{ji} is the weight between input neuron i and hidden neuron j, and b_j is the bias of the hidden neuron j. So the output can be computed as shown in Equation (7). Note that f_2 is the activation function which is generally linear, n_h is the number of hidden neurons, w_{oj} is the weight between hidden neuron j and output neuron o, and b_o is the bias of the output neuron o. Output is computed by proceeding forward through the network layer by layer. After the error value is calculated, local gradients are computed backward through the network and weights are adjusted (Kumpati and Kannan, 1990). The Levenberg-Marquardt method is a frequently used training algorithm in the literature and a compromise between the Newton's method and the gradient descent (Levenberg, 1944; Marquardt, 1963). According to this method, the optimum adjustment applied to the parameter vector $\Delta\mathbf{w}$ is defined by Equation (8) where $\mathbf{g}$ is the gradient vector, $\mathbf{H}$ is the Hessian matrix, $\mathbf{I}$ is the identity matrix and λ is regularizing parameter.

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-d_y}, x_{t-1}, x_{t-2}, \dots, x_{t-d_x}) \tag{5}$$

$$z_{j(t-1)} = f_1\left(\sum_{i=1}^{d_y} w_{ji} y_{t-i} + \sum_{i=1}^{d_x} w_{ji} x_{t-i} + b_j\right) \tag{6}$$

$$y_t = f_2\left(\sum_{j=1}^{n_h} w_{oj} z_{j(t-1)} + b_o\right) \tag{7}$$

$$\Delta\mathbf{w} = [\mathbf{H} + \lambda\mathbf{I}]^{-1}\mathbf{g} \tag{8}$$

In the NARX model, after we train the open-loop network, we transform it into a closed-loop network structure and make prediction for the next period. Closed-loop neural networks, on the other hand, generate outputs depending on the time series of input parameters where the time series of the predicted outputs also feeds the network as an input. In other words, compared with the open-loop structure, the delayed target is replaced with the delayed output. As a result, this structure generates output for the common time period with the input. Therefore, it is known that it is more advantageous to use closed-loop network structure for multi-step-ahead prediction (Boussaada, Curea, Remaci, Camblong and Mrabet Bellaaj, 2018; Xie, Tang and Liao, 2009). As shown in Fig. 2, we consider two types of exogenous inputs as day type (x_1) and time period of the day (x_2) in the input layer. In the output layer, there is one single neuron representing the parking occupancy rate to be predicted both by its historical values and by the values of exogenous variables. Both input and output delays are taken as two units. Also, in the sample network representation in the figure, there is one hidden layer and ten hidden neurons in this layer. We also discuss different network topologies in the following section.

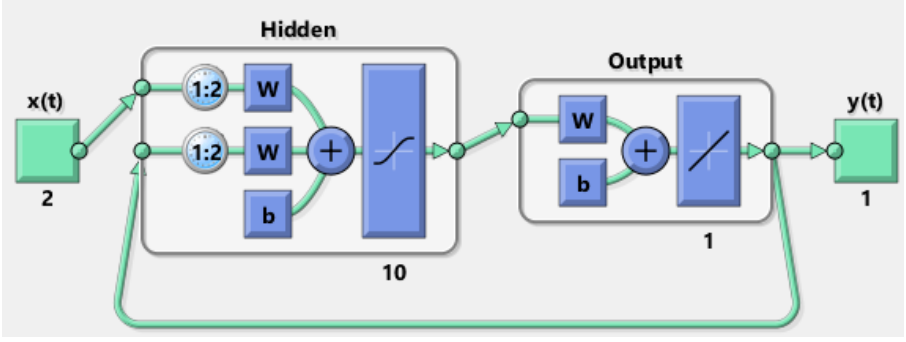

Fig. 2. An Example Representation of the Neural Network Structure

4. Implementation and Results

In this section, we present the implementation steps and the results obtained in our computational experiments. For each parking space, we make predictions with different model structures of the considered methods. The data set we use in the implementation phase belongs to the date range of July 2012 and August 2012 as a suitable sample of the dataset of the SFpark project providing a comprehensive coverage of the corresponding date range. We first estimate the model parameters of the ARIMA models and train the neural networks to estimate the network parameters. We thus determine the appropriate model structure for each method. After we determine the model, we make predictions and analyze the results in terms of mean squared errors. We implement all approaches in MATLAB. After data pre-processing for each parking space (i.e.,

preparing data of each parking space for prediction), we estimate the related model parameters using the 85% of the corresponding sample. We use the remaining 15% of the data in the prediction phase.

4.1 ARIMA Models

In Table 3, the results of the experiments for different ARIMA model structures are given. Here we consider model constructions consisting of combinations of different (one and two) values of p, d and q. In the table, the rows represent each model structure, and the columns represent the parking spaces. The last three columns contain the minimum, maximum and average performance values of each model, respectively. We observe that if the d value equals to one, the performance improves significantly compared to the cases where the d value equals to two which is the highest allowed value of d in ARIMA models is two in MATLAB. n-inv means that estimated model is non-invertible. The last three rows show the minimum, maximum and average performance values of the predictions for each parking space, respectively. The best result for each parking space is highlighted in gray as shown in the table.

In general, we can say that we observe better performance parameters for the parking spaces with an occupancy rate pattern that does not contain large fluctuations, such as parking space P01 (block 41103). On the other hand, for some parking spaces, where the occupancy rates of these parking spaces differ greatly between day and night, and where the occupancy rates of the parks have large fluctuations, the performance values are at extreme values, such as parking space P11 (block 61418). The minimum and maximum values observed to measure the overall performance of the ARIMA models are indicated in green and orange, respectively. As seen in the table, the performance of this model ranges from 0.007 to 6.565, and as highlighted in blue, the best prediction in terms of averages is made with the $ARIMA(2,1,2)$ model. Since we observe better results when d equals to one, we analyze the effects of the different values of the p and q parameters when it is fixed to one, but we don't observe any more satisfactory results than the ones presented in the table.

The results of the experiments for different SARIMA model structures are given in Table 4. Here, unlike the ARIMA model, we also discuss the effect of seasonality as 12-hour, 6-hour and 4-hour. The last three columns of the table contain the minimum, maximum and average performance values of each model, respectively. The last three rows also show the minimum, maximum and average performance values of the predictions for each parking space, respectively. As before, n-inv means that estimated model is non-invertible. The best result for each parking space is highlighted in gray as shown in the table. The minimum and maximum values observed to measure the overall performance of the SARIMA models are indicated in green and orange, respectively and the

performance of this model ranges from 0.012 to 184.014. The best prediction in terms of average performance is made with the $SARIMA(1,1,2)(1,1,2)_{12}$ model as highlighted in blue. We can say that for each parking space there is no significant difference between the best results found with the SARIMA models when compared to the best results found with the ARIMA models. On the other hand, the results for the case where the integration parameter (d) is equal to two are noticeably worse in this model structure resulting a wide range of performance parameters.

4.2 Neural Networks

The results obtained for different neural network topologies are given in Table 5. Here, HL represents the number of hidden layers and HN is the number of neurons in each hidden layer. For each network topology, we train the network 100 times, and get the training performance in terms of mean squared errors for each training. We then compare performance results and save the network with the best performance. Finally, we make predictions with this network and, once again, calculate mean squared of errors. In the table, the rows represent each network topology, and the columns represent the parking spaces. The minimum, maximum and average performance values of 100 runs for each topology are also given. The last three columns contain overall minimum, maximum and average performance values, respectively, foreach network topology. The best result for each parking space is highlighted in gray as shown in the table. We cannot say that the changes in the network topology have significant effects. On the other hand, it is clearly seen in the table that the overall performance of the neural networks is better than those of ARIMA (both ARIMA and SARIMA) models. The 2 HL – 5/5 HN design with two hidden layers and five hidden neurons in each layer gives the best results in terms of average performance as highlighted in blue. We would like to point out that best performance value is equal to 0.005 for all network topologies. It is seen that the average performance values are very close to each other and range from 0.145 to 0.610, and hence, in terms of the parking space predictions, we clearly see that there are no large fluctuations compared to ARIMA-based methods.

We also analyze some extreme number of neurons in the hidden layers. When it is increased to 50, for instance, the results are not better than the current situation. On the other hand, if this number is 100, the results worsen significantly. Moreover, we examine the degree of delays. When we evaluate greater values than two, the results also get worse. When we reduce it to one, no significant differences are observed compared to the current situation.

Table 3
Results Obtained with ARIMA Models

Model	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	min	max	avg
(1,1,1)	0.026	0.253	0.048	0.007	0.016	0.092	0.090	0.085	0.045	0.145	0.132	0.143	0.007	0.253	0.090
(1,2,1)	0.395	2.574	0.153	0.051	0.577	1.587	2.525	4.396	0.113	0.114	6.323	0.810	0.051	6.323	1.635
(1,1,2)	0.025	0.168	0.049	0.007	0.016	0.096	0.078	0.113	0.045	0.146	0.124	0.141	0.007	0.168	0.084
(1,2,2)	0.468	4.059	0.222	*n-inv*	0.585	0.135	2.317	0.390	0.531	0.528	6.565	0.066	0.066	6.565	1.442
(2,1,1)	0.020	0.111	0.049	0.007	0.019	0.099	0.068	0.098	0.043	0.149	0.129	0.104	0.007	0.149	0.075
(2,2,1)	0.703	2.323	0.168	0.049	0.603	0.859	0.393	3.593	1.833	0.135	4.011	0.633	0.049	4.011	1.275
(2,1,2)	0.015	*n-inv*	0.048	*n-inv*	0.019	0.123	0.067	0.068	0.044	0.123	0.121	0.094	0.015	0.123	0.072
(2,2,2)	0.711	1.928	0.091	0.060	0.030	1.180	0.353	0.323	1.278	0.370	1.997	0.103	0.030	1.997	0.702
min	0.015	0.111	0.048	0.007	0.016	0.092	0.067	0.068	0.043	0.114	0.121	0.066			
max	0.711	4.059	0.222	0.060	0.603	1.587	2.525	4.396	1.833	0.528	6.565	0.810			
avg	0.295	1.631	0.104	0.030	0.233	0.521	0.736	1.133	0.491	0.214	2.425	0.262			

Table 4
Results Obtained with SARIMA Models

Model	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	min	max	avg
$(1,1,1)(1,1,1)_{12}$	0.025	0.390	0.412	0.067	0.037	0.095	0.104	0.256	0.216	0.162	0.355	0.170	0.025	0.412	0.191
$(1,2,1)(1,2,1)_{12}$	2.889	0.050	6.811	0.217	0.375	13.912	0.677	0.864	14.930	48.746	36.562	0.273	0.050	48.746	10.525
$(1,1,2)(1,1,2)_{12}$	0.035	0.067	0.101	*n-inv*	0.025	0.203	0.059	*n-inv*	0.258	0.131	0.158	0.092	0.025	0.258	0.113
$(1,2,2)(1,2,2)_{12}$	0.124	0.124	0.246	0.034	0.418	16.540	0.521	2.707	7.925	29.516	12.195	0.266	0.034	29.516	5.885
$(2,1,1)(2,1,1)_{12}$	0.098	0.082	0.123	0.055	0.039	0.148	0.157	0.233	0.267	0.067	0.120	0.305	0.039	0.305	0.141
$(2,2,1)(2,2,1)_{12}$	10.554	39.370	0.175	0.219	0.041	3.533	0.057	3.743	11.848	0.546	31.645	13.822	0.041	39.370	9.629
$(2,1,2)(2,1,2)_{12}$	0.108	0.091	0.096	*n-inv*	*n-inv*	0.094	0.073	0.117	0.347	0.091	0.116	0.118	0.073	0.347	0.125
$(2,2,2)(2,2,2)_{12}$	0.033	12.293	*n-inv*	*n-inv*	0.047	18.197	0.095	0.663	22.031	5.307	15.771	0.976	0.033	22.031	7.541
$(1,1,1)(1,1,1)_{6}$	0.019	2.814	0.067	0.182	0.053	1.003	0.224	0.108	0.154	0.717	7.153	0.246	0.019	7.153	1.062
$(1,2,1)(1,2,1)_{6}$	5.795	6.527	0.127	0.232	1.246	5.434	0.492	19.766	3.339	5.649	81.066	0.800	0.127	81.066	10.873
$(1,1,2)(1,1,2)_{6}$	0.061	1.392	0.153	0.943	0.095	0.665	0.854	0.073	0.510	0.940	0.521	0.095	0.061	1.392	0.525
$(1,2,2)(1,2,2)_{6}$	1.654	0.347	0.185	0.296	0.935	11.865	0.607	1.148	16.984	20.406	121.949	*n-inv*	0.185	121.949	16.034
$(2,1,1)(2,1,1)_{6}$	0.012	0.400	0.123	0.017	0.029	0.191	0.104	0.118	0.482	0.064	1.736	0.292	0.012	1.736	0.297
$(2,2,1)(2,2,1)_{6}$	27.421	35.924	0.091	0.192	0.102	0.330	0.498	132.652	4.097	11.852	31.938	3.973	0.091	132.652	20.756
$(2,1,2)(2,1,2)_{6}$	0.023	0.576	0.068	0.015	0.049	0.116	0.051	0.087	0.389	0.117	0.195	0.118	0.015	0.576	0.150
$(2,2,2)(2,2,2)_{6}$	35.428	12.955	*n-inv*	0.213	0.030	0.380	*n-inv*	44.930	1.680	5.180	153.577	1.203	0.030	153.577	25.558
$(1,1,1)(1,1,1)_{4}$	0.278	2.121	0.167	0.019	0.061	0.743	0.034	1.125	0.373	0.286	5.404	0.236	0.019	5.404	0.904
$(1,2,1)(1,2,1)_{4}$	7.736	44.665	18.097	1.006	2.148	55.909	0.467	0.336	3.374	24.410	184.014	8.917	0.336	184.014	29.256
$(1,1,2)(1,1,2)_{4}$	0.196	0.902	0.106	0.031	0.028	0.703	0.067	0.268	0.277	0.222	5.074	0.179	0.028	5.074	0.671
$(1,2,2)(1,2,2)_{4}$	15.470	52.181	7.422	0.318	0.352	65.347	0.569	5.961	2.204	60.932	163.155	1.239	0.318	163.155	31.262
$(2,1,1)(2,1,1)_{4}$	0.059	1.577	0.219	0.019	0.021	0.115	0.492	0.265	0.491	0.078	1.461	0.264	0.019	1.577	0.422
$(2,2,1)(2,2,1)_{4}$	1.385	1.477	11.607	0.192	0.279	8.118	27.704	28.287	3.291	0.899	53.624	18.104	0.192	53.624	12.914
$(2,1,2)(2,1,2)_{4}$	0.021	0.228	0.145	0.231	0.030	0.140	0.044	0.304	0.495	0.054	1.110	*n-inv*	0.021	1.110	0.255
$(2,2,2)(2,2,2)_{4}$	*n-inv*	1.733	0.288	0.189	*n-inv*	0.206	4.285	*n-inv*	0.048	0.193	183.563	1.371	0.048	183.563	21.319
min	0.012	0.050	0.067	0.015	0.021	0.094	0.034	0.073	0.048	0.054	0.116	0.092			
max	35.428	52.181	18.097	1.006	2.148	65.347	27.704	132.652	22.031	60.932	184.014	18.104			
avg	4.758	9.095	2.129	0.223	0.293	8.499	1.662	11.091	4.000	9.023	45.519	2.412			

Table 5
Results Obtained with Neural Networks

Model		P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	min	max	avg
	min	0.008	0.018	0.028	0.005	0.013	0.042	0.027	0.023	0.031	0.027	0.019	0.059			
1HL – 5 HN	max	0.029	3.226	0.358	0.017	0.050	0.627	0.170	0.064	5.656	0.129	9.039	0.552	0.005	9.039	0.586
	avg	0.014	0.097	0.063	0.008	0.019	0.173	0.039	0.036	0.136	0.058	0.160	0.081			
	min	0.010	0.017	0.035	0.005	0.013	0.048	0.024	0.022	0.031	0.026	0.015	0.056			
1 HL – 10 HN	max	0.709	0.736	0.863	0.102	0.041	2.387	0.108	0.644	0.160	0.758	2.129	2.365	0.005	2.387	0.338
	avg	0.022	0.067	0.079	0.010	0.020	0.262	0.041	0.045	0.049	0.056	0.085	0.119			
	min	0.008	0.017	0.033	0.005	0.012	0.062	0.021	0.021	0.033	0.028	0.012	0.063			
1 HL – 20 HN	max	0.199	1.408	1.072	0.183	0.090	3.290	1.477	0.138	3.295	3.347	1.615	4.293	0.005	4.293	0.610
	avg	0.020	0.115	0.090	0.014	0.022	0.426	0.059	0.045	0.133	0.080	0.097	0.153			
	min	0.009	0.015	0.025	0.005	0.014	0.031	0.022	0.022	0.033	0.028	0.016	0.061			
2 HL – 5/5 HN	max	0.041	0.254	0.155	0.344	0.055	1.533	0.210	0.095	0.363	0.104	0.238	0.882	0.005	1.533	0.145
	avg	0.015	0.039	0.056	0.012	0.018	0.194	0.036	0.040	0.049	0.055	0.050	0.086			
	min	0.009	0.016	0.026	0.005	0.012	0.038	0.020	0.023	0.030	0.030	0.016	0.061			
2 HL – 5/10 HN	max	0.050	1.350	0.136	0.039	0.043	3.312	0.273	0.074	0.340	0.186	0.911	0.601	0.005	3.312	0.231
	avg	0.016	0.067	0.058	0.009	0.020	0.214	0.041	0.039	0.056	0.062	0.055	0.084			
	min	0.009	0.015	0.026	0.005	0.013	0.041	0.021	0.026	0.033	0.025	0.015	0.060			
2 HL – 5/20 HN	max	0.277	0.549	3.401	0.075	0.152	1.405	0.309	0.109	2.742	0.295	3.442	0.647	0.005	3.442	0.404
	avg	0.024	0.063	0.098	0.010	0.022	0.213	0.043	0.042	0.076	0.059	0.102	0.089			

Table 5 (continue)
Results Obtained with Neural Networks

Model		P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	min	max	avg
	min	0.009	0.019	0.026	0.005	0.013	0.044	0.017	0.020	0.031	0.028	0.013	0.063			
2 HL – 10/5 HN	max	0.072	2.339	0.112	0.034	0.058	0.903	0.106	0.217	1.257	0.331	0.132	0.155	0.005	2.339	0.187
	avg	0.017	0.093	0.057	0.009	0.019	0.207	0.037	0.041	0.060	0.056	0.041	0.078			
	min	0.008	0.018	0.024	0.005	0.012	0.044	0.018	0.018	0.031	0.026	0.010	0.058			
2 HL – 10/10 HN	max	0.049	0.584	0.402	0.437	0.059	0.772	0.096	0.137	3.985	0.095	1.869	1.510	0.005	3.985	0.308
	avg	0.017	0.050	0.067	0.015	0.020	0.226	0.038	0.045	0.118	0.050	0.067	0.095			
	min	0.009	0.017	0.025	0.005	0.014	0.041	0.019	0.017	0.035	0.026	0.016	0.059			
2 HL – 10/20 HN	max	0.556	1.569	0.484	0.071	0.072	3.019	1.163	1.652	1.748	0.109	1.017	0.212	0.005	3.019	0.361
	avg	0.030	0.123	0.083	0.010	0.021	0.353	0.062	0.059	0.097	0.049	0.067	0.084			
	min	0.009	0.016	0.034	0.005	0.013	0.044	0.017	0.019	0.031	0.027	0.014	0.062			
2 HL – 20/5 HN	max	0.151	1.263	0.233	0.024	0.065	1.696	0.294	0.514	1.848	0.114	0.225	0.999	0.005	1.848	0.236
	avg	0.018	0.058	0.065	0.008	0.022	0.258	0.046	0.047	0.084	0.052	0.048	0.087			
	min	0.009	0.015	0.033	0.005	0.012	0.061	0.017	0.023	0.033	0.024	0.010	0.059			
2 HL – 20/10 HN	max	0.038	1.493	0.278	0.051	0.054	5.617	2.164	1.460	1.269	0.103	0.409	0.282	0.005	5.617	0.402
	avg	0.013	0.088	0.075	0.009	0.021	0.350	0.080	0.060	0.089	0.049	0.046	0.080			
	min	0.009	0.017	0.032	0.005	0.011	0.053	0.022	0.012	0.031	0.024	0.016	0.060			
2 HL – 20/20 HN	max	0.089	2.358	0.226	0.067	0.109	2.573	0.387	0.206	2.795	0.118	1.864	1.898	0.005	2.795	0.393
	avg	0.021	0.155	0.079	0.011	0.022	0.343	0.051	0.043	0.163	0.050	0.081	0.142			

Although we use the Levenberg-Marquardt training algorithm (*trainlm*) in our computational experiments as the default training function in MATLAB, we also examine the effects of different training functions. We observe satisfying performance values with another training function (*trainbr*), in which the Levenberg-Marquardt training algorithm is used with Bayesian Regularization. In this method, weight and bias values are updated according to the Levenberg-Marquardt algorithm. A combination of squared errors and weights are minimized, and then correct combination is determined in order to set the suitable network structure based on the Bayesian inference techniques. This method is known for longer processing times compared to other training functions and it decreases the negative effects of higher weights. Although it gives satisfactory results in terms of the best performance scores it produces, we observe that the performance values lie in larger ranges. We finally consider another function (*trainscg*) which updates weight and bias values according to the scaled conjugate gradient method known as giving better results with small datasets. On the other hand, we do not observe better results than the best-known performance parameters.

5. Conclusions

On average 30% of traffic congestion is caused by drivers cruising in order to find a vacant parking space. This causes significant fuel and time loss for drivers. On the other hand, from the perspective of society and the environment, the problem becomes much more critical. Therefore, the solution of the parking problem becomes important for reducing the stress level of people, urban traffic congestion and environmental pollution. Prediction of the availability of the parking spaces is one of the most important factors in solving the parking management problem. In this study, we make parking prediction using three different machine learning techniques as ARIMA, SARIMA and neural networks. For each parking space in the considered data set, we make predictions with different model structures of these methods and then analyze the results in terms of mean squared errors. We compare the results and find the best model design of each method for each parking space. We also evaluate the results in terms of the superiority of the methods over each other. We can clearly say that the performance of neural networks is better both in terms of averages and the best results for each parking space.

Acknowledgments

Asli SEBATLI-SAGLAM is supported by the Turkish Council of Higher Education (CoHE) under the 100/2000 PhD Scholarship Program and the Scientific and Technological Research Council of Turkey (TUBITAK) under the 2211/A

National PhD Scholarship Program. We would like to thank CoHE and TUBITAK for their valuable supports.

Authors' Contributions

In this study; Asli SEBATLI-SAGLAM, created the models, performed the exponential studies, and wrote the article; Fatih CAVDUR outlined the study and edited the article.

Conflict of Interest

There is no conflict of interest to declare.

References

Arjona, J., Linares, M., Casanovas-Garcia, J., & Vazquez, J.J. (2020). Improving parking availability information using deep learning techniques. *Transportation Research Procedia*, 47, 385-392. Doi: https://doi.org/10.1016/j.trpro.2020.03.113

Awan, F.M., Saleem, Y., Minerva, R., & Crespi, N. (2020). A comparative analysis of machine/deep learning models for parking space availability prediction. *Sensors*, 20(1), 322. Doi: https://doi.org/10.3390/s20010322

Badii, C., Nesi, P., & Paoli, I. (2018). Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access*, 6, 44059-44071. Doi: https://doi.org/10.1109/access.2018.2864157

Balmer, M., Weibel, R., & Huang, H. (2021). Value of incorporating geospatial information into the prediction of on-street parking occupancy–A case study. *Geo-spatial Information Science*, 24(3), 438-457. Doi: https://doi.org/10.1080/10095020.2021.1937337

Boussaada, Z., Curea, O., Remaci, A., Camblong, H., & Mrabet Bellaaj, N. (2018). A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies*, 11(3), 620. Doi: https://doi.org/10.3390/en11030620

Box, G.E.P., & Jenkins, G.M. (1976). Time series analysis: Forecasting and control. *Holden-Day*, San Francisco.

Caliskan, M., Barthels, A., Scheuermann, B., & Mauve, M. (2007). Predicting parking lot occupancy in vehicular ad hoc networks. *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, IEEE, Dublin, 277-281. Doi: https://doi.org/10.1109/vetecs.2007.69

Camero, A., Toutouh, J., Stolfi, D.H., & Alba, E. (2018). Evolutionary deep learning for car park occupancy prediction in smart cities. *International Conference on Learning and Intelligent Optimization*, Springer, Cham, Kalamata, 386-401. Doi: https://doi.org/10.1007/978-3-030-05348-2_32

Chang, A.S., & Kalawsky, R.S. (2017). European transport sector interventions for smart city. *7th International Conference on Power Electronics Systems and Applications-Smart Mobility, Power Transfer & Security (PESA)*, IEEE, Hong Kong, 1-6. Doi: https://doi.org/10.1109/pesa.2017.8277778

Cookson, G. (2017). Smart Parking – A Silver Bullet for Parking Pain. Retrieved from : http://inrix.com/blog/2017/07/parkingsurvey.

Enriquez, F., Soria, L.M., Alvarez-Garcia, J.A., Velasco, F., & Deniz, O. (2017). Existing approaches to smart parking: An overview. *International Conference on Smart Cities*, Springer, Cham, Malaga, 63-74. Doi: https://doi.org/10.1007/978-3-319-59513-9_7

Fabusuyi, T., Hampshire, R.C., Hill, V.A., & Sasanuma, K. (2014). Decision analytics for parking availability in downtown Pittsburgh. *Interfaces*, 44(3), 286-299. Doi: https://doi.org/10.1287/inte.2014.0743

Ionita, A., Pomp, A., Cochez, M., Meisen, T., & Decker, S. (2018). Where to park? predicting free parking spots in unmonitored city areas. *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics,* Novi Sad, 1-12.Doi: https://doi.org/10.1145/3227609.3227648

Jin, C., Wang, L., Shu, L., Feng, Y., & Xu, X. (2012). A fairness-aware smart parking scheme aided by parking lots. *2012 IEEE International Conference on Communications (ICC)*, IEEE, Ottawa, 2119-2123. Doi: https://doi.org/10.1109/icc.2012.6364635

Kuhail, M.A., Boorlu, M., Padarthi, N., & Rottinghaus, C. (2019). Parking availability forecasting model. *2019 IEEE International Smart Cities Conference (ISC2)*, IEEE, Casablanca, 619-625. Doi: https://doi.org/10.1109/isc246665.2019.9071688

Kumpati, S.N., & Kannan, P. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1), 4-27. Doi: https://doi.org/10.1109/72.80202

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2), 164-168. Doi: https://doi.org/10.1090/qam/10666

Li, J., Li, J., & Zhang, H. (2018). Deep learning based parking prediction on cloud platform. *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*, IEEE, Chicago, 132-137. Doi: https://doi.org/10.1109/bigcom.2018.00028

Lin, T., Horne, B.G., Tino, P., & Giles, C.L. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6), 1329-1338.Doi: https://doi.org/10.1109/72.548162

Lin, T., Rivano, H., & Le Mouel, F. (2017). A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3229-3253. Doi: https://doi.org/10.1109/tits.2017.2685143

Lu, E.H.C., & Liao, C.H. (2020). Prediction-based parking allocation framework in urban environments. *International Journal of Geographical Information Science*, 34(9), 1873-1901. Doi: https://doi.org/10.1080/13658816.2020.1721503

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2), 431-441. Doi: https://doi.org/10.1137/0111030

Pflugler, C., Kohn, T., Schreieck, M., Wiesche, M., & Krcmar, H. (2016). Predicting the availability of parking spaces with publicly available data. *Informatik*, 2016, 361-374.

Provoost, J.C., Kamilaris, A., Wismans, L.J., Van Der Drift, S.J., & Van Keulen, M. (2020). Predicting parking occupancy via machine learning in the web of things. *Internet of Things*, 12, 100301. Doi: https://doi.org/10.1016/j.iot.2020.100301

Rajabioun, T., & Ioannou, P.A. (2015). On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, 16(5), 2913-2924. Doi: https://doi.org/10.1109/tits.2015.2428705

Richter, F., Di Martino, S., & Mattfeld, D.C. (2014). Temporal and spatial clustering for a parking prediction service. *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, IEEE, Limassol, 278-282. Doi: https://doi.org/10.1109/ictai.2014.49

Schrank, D.L., & Lomax, T.J. (2007). The 2007 urban mobility report. *Texas Transportation Institute The Texas A&M University System*.

Shoup, D C. (2006). Cruising for parking. *Transport Policy*, 13(6), 479-486. Doi: https://doi.org/10.1016/j.tranpol.2006.05.005

Siegelmann, H.T., Horne, B.G., & Giles, C.L. (1997). Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(2), 208-215. Doi: https://doi.org/10.1109/3477.558801

Stolfi, D.H., Alba, E., & Yao, X. (2017). Predicting car park occupancy rates in smart cities. *International Conference on Smart Cities*, Springer, Cham, Malaga, 107-117 Doi: https://doi.org/10.1007/978-3-319-59513-9_11.

Stolfi, D.H., Alba, E., & Yao, X. (2020). Can I Park in the City Center? Predicting Car Park Occupancy Rates in Smart Cities. *Journal of Urban Technology*, 27(4), 27-41. Doi: https://doi.org/10.1080/10630732.2019.1586223

Tamrazian, A., Qian, Z., & Rajagopal, R. (2015). Where is my parking spot? Online and offline prediction of time-varying parking occupancy. *Transportation Research Record*, 2489(1), 77-85. Doi: https://doi.org/10.3141/2489-09

Tavafoghi, H., Poolla, K., & Varaiya, P. (2019). A Queuing Approach to Parking: Modeling, Verification, and Prediction. *arXiv preprint arXiv:1908.11479*, 1-28.

Teng, H., Falcocchio, J.C., Lapp, F., Price, G.A., Prassas, S., & Kolsal, A. (2001). Parking information and technology for a parking information system. *Transportation Research Board 80th Annual Meeting*, Washington, 34.

Tiedemann, T., Vögele, T., Krell, M.M., Metzen, J.H., & Kirchner, F. (2015). Concept of a data thread based parking space occupancy prediction in a berlin pilot region. *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin Texas.

Tilahun, S. L., & Di Marzo Serugendo, G. (2017). Cooperative multiagent system for parking availability prediction based on time varying dynamic Markov chains. *Journal of Advanced Transportation*, 2017, 1-14. Doi: https://doi.org/10.1155/2017/1760842

Vlahogianni, E.I., Kepaptsoglou, K., Tsetsos, V., & Karlaftis, M.G. (2016). A real-time parking prediction system for smart cities. *Journal of Intelligent Transportation Systems*, 20(2), 192-204. Doi: https://doi.org/10.1080/15472450.2015.1037955

White, P. (2007). No Vacancy: Park Slopes Parking Problem And How to Fix It. Retrieved from: http://www.transalt.org/newsroom/releases/126, access date: 09.03.2020.

Xiao, J., Lou, Y., & Frisby, J. (2018). How likely am I to find parking?–A practical model-based framework for predicting parking availability. *Transportation Research Part B: Methodological*, 112, 19-39. Doi: https://doi.org/10.1016/j.trb.2018.04.001

Xie, H., Tang, H., & Liao, Y.H. (2009). Time series prediction based on NARX neural networks: An advanced approach. *2009 International Conference on Machine Learning and Cybernetics*, IEEE, Vol. 3, Hebei, 1275-1279. Doi: https://doi.org/10.1109/icmlc.2009.5212326

Yang, S., Ma, W., Pi, X., & Qian S. (2019). 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. Doi: https://doi.org/10.1016/j.trc.2019.08.010

Zhao, Z., Zhang, Y., & Zhang, Y. (2020). A comparative study of parking occupancy prediction methods considering parking type and parking scale. *Journal of Advanced Transportation*, 2020, 1-12. Doi: https://doi.org/10.1155/2020/5624586

Zheng, Y., Rajasegarar, S., & Leckie, C. (2015). Parking availability prediction for sensor-enabled car parks in smart cities. *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, IEEE, Singapore, 1-6. Doi: https://doi.org/10.1109/issnip.2015.7106902

Ziat, A., Leroy, B., Baskiotis, N., & Denoyer, L. (2016). Joint prediction of road-traffic and parking occupancy over a city with representation learning. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Rio de Janeiro, 725-730. Doi: https://doi.org/10.1109/itsc.2016.7795634