IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 17, 2021, accepted April 18, 2021, date of publication April 22, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074887

Deep Learning-Based Mobile Application Design for Smart Parking

H. CANLI[1] AND S. TOKLU[2]
[1]Department of Computer Engineering, Faculty of Engineering, Duzce University, 81010 Duzce, Turkey
[2]Department of Computer Engineering, Faculty of Technology, Gazi University, 06100 Ankara, Turkey

Corresponding author: H. Canli (hikmetcnli@hotmailcom)

ABSTRACT In the era of Internet of Things (IoT) and smart city ecosystems, there is a need for innovative smart parking systems for more sustainable cities. With the increasing number of vehicles in the cities every year, it takes more time to find parking spaces. The solution methods developed are no longer sufficient. The time that passes while waiting for a parking space in traffic carries with it problems such as energy, environmental pollution and stress. In this study, a deep learning and cloud-based new mobile smart parking application was developed to minimize the problem of searching for parking spaces. Within the application, a service has been developed based on deep learning with Long short-term memory (LSTM) to predict the parking space. Here, dynamic access is provided to the LSTM-based model previously created through the mobile device of the user, and the process of displaying the occupancy rates of the parks at the desired place is accomplished on the mobile device by entering the relevant parameters. By this means, both energy and time savings have been achieved. With the real-time car parking data collected in the city of Istanbul in Turkey, high accuracy results were obtained. In order to demonstrate the effectiveness of the model proposed, it was compared with the Support Vector Machine, Random Forest and ARIMA methods. The results have confirmed the high accuracy and reliability that was promised.

INDEX TERMS Smart city, deep learning, LSTM, support vector machine, random forest, ARIMA.

I. INTRODUCTION

Cities are getting smarter today with the increasing use of the Internet of Things (IoT). IoT applications are increasing rapidly in many areas of the cities such as transportation systems, airports, hospitals, shopping malls, etc. In 2021, devices connected to IoT are expected to exceed 27 million devices, and this is an enormous figure in the Information technology (IT) world [1]. Daily life has become easier compared to the last ten years with the use of this technology [2]. Smart parking systems are among the important topics of IoT-based smart cities. Increasing number of vehicles in the cities is the prominent factor for smart car parking systems to become important. Parking problems, which occur with the number of new vehicles joining the traffic every day, have attracted more attention in recent years [3], [4].

Istanbul, with its 15.5 million people, is the largest metropolitan area of Turkey and is considered to be the country's economic, cultural and historic center [5]. The biggest problem of the people living here is the traffic density and the problem of not being able to find a parking space, especially during their commute hours [6]. Few things in life can be more inconvenient for city dwellers than anxiously searching for parking spaces. In the literature, different solution methods are presented for problems for parking and search for parking spaces. Even though these methods yield considerably good results in the short term, new solution methods are needed as the number of vehicles participating in the traffic increases more and more each year.

In this study, a new method and a mobile application running on deep learning and cloud-based are presented to solve the problem of searching for parking spaces in the city of Istanbul. With the mobile application, the user can dynamically access the model developed using the deep learning method and the real-time and forward-looking results can rapidly be shown to the user. The process has two stages. In the first stage, information such as the capacity, empty capacity, latitude, longitude etc. is instantly taken from 731 parking spaces in Istanbul then this information is stored in the cloud. Present in the second stage is a deep

The associate editor coordinating the review of this manuscript and approving it for publication was Syed Islam.

learning-based Long short-term memory (LSTM) service that processes the requests coming from the mobile applications for parking spaces. This service predicts the occupancy rate of the relevant parking space in the short and medium term based on the requests that are received from the application and responds to the user.

The fact that it is shown among the deep learning models recognized to make successful predictions on multivariate time series created from large data set, has an influence in the use of LSTM, the prediction model proposed [7]. Naturally, there are many deep learning methods. However, as shown in the assessment section, very successful results have been obtained through LSTM. When designing a successful prediction method for parking occupancy, the accuracy of the flow data, the time and the parameters that affect the parking space occupancy are of great importance. The insufficient amount of data obtained in the prediction models collected so far has affected the success of the models designed. Before starting the studies conducted, about 4.5 million data of various features were collected and the collection continues. The fact that the success of deep learning models is parallel to the increase in data has been the source of motivation in this study.

In order to show the efficiency of the prediction model created, it was compared to the Support Vector Machine and the Random Forest model, which are machine learning methods, as well as to the ARIMA model, which is frequently used in the analysis of time series. When the experimental results obtained are examined, it is seen that the LSTM model suggested yields the best results with an accuracy rate of 99.57%.

In summary, the contributions of this study are as follows:

a) A new architecture and a new application based on deep learning have been developed for detection of parking occupancy in IoT-based smart cities.
b) With the "LSTM-based deep learning approach" and multivariate and multi-time series parking data set, more dominant results were obtained in determining the car park occupancy than could be determined with Support Vector Machine (SVM), Random Forest (RF) and ARIMA.
c) A large data set that can be used and inspired by other studies in the literature was created by collecting parking space data with different parameters for a year.

In the second section of this study, studies on smart parking systems are analyzed in detail. In section 3, deep learning based car park occupancy prediction was mentioned, and the proposed deep learning method LSTM architecture and the dataset used were explained. In section 4 the proposed model is explained and its details are examined in three sub-sections. Deployment model of the proposed method in the first section, flow diagram in the second section, model training and evaluation in the last section are discussed in detail. The success of the method suggested in section 5 and its contribution to the literature are explained and recommendations for future studies have been made.

II. RELATED WORKS

The population increases day by day in the cities, and the resources remain limited. Sustainable innovative solutions are needed to manage these limited resources [7]. With the increasing number of vehicles in cities and Internet of Things (IoT), innovative solutions are also required for limited parking spaces in smart cities [10], [11]. There are many studies in the literature carried out on the solution for parking space problems [12].

In the study conducted in [13], ways of directing drivers to parking spaces in urban areas were investigated. Static and dynamic routing were mentioned and advantages and disadvantages of these methods were discussed. In [14], the smart parking guidance system was introduced. An intelligent parking guidance algorithm was proposed, taking into account the factors that affected the selection of the carpark. With the algorithm and method proposed, drivers of the vehicles were enabled to find the most suitable parking space. In [15], the smart parking guidance study was conducted in a pilot area in the carpark of the Boston university.

In [16], a study was carried out to solve the problems experienced in parking spaces of the shopping malls, which are busy on weekends. A genetic approach was used to place a vehicle in a parking space. They reduced the waiting time in the queue by letting the drivers park in the most suitable parking spaces.

In [17], a new cloud-based intelligent vehicle parking system (SVPS) was proposed over ubiquitous VANETs to provide more robust parking solutions. The proposed SVPS architecture offers a unique algorithm that provides a convenient parking space information as well as reservation and suggestion options to facilitate vehicles in an efficient, real-time and precise manner. Simulations made demonstrate that the architecture proposed made use of the parking resources in the most efficient manner. In [18], a VANET based smart parking scheme (SPARK) was proposed for large car parks. They monitored the car park on all RSUs in a car park with the SPARK scheme. With the data they obtained, they provided real-time parking navigation, smart theft protection and the spread of friendly parking information.

In [19], a cloud-based architecture was created to bring together parking space service providers and drivers. Park owners that provide services can advertise their services thorugh this system. In [11], the performance of the algorithms used in search for parking spaces was examined in terms of energy and time. The hierarchy-based BST algorithm was used and it was shown to be more efficient than non-hierarchical aspects.

In [20] the issue was looked over from a different perspective. Smart phones were used to detect parking spaces. When a driver parks or leaves the parking space, the relevant parking space can be detected. It gives high accuracy rate even in closed parking spaces. In [21], the performances of the sensors that control whether the parking spaces in the car parks are occupied or not were examined. These sensors exhibit different performances depending on the angle of the light and the distance. In [22], discussion was about decreasing the traffic arising from personnel at the entrance and exit of the car park. An automatic barrier system was developed with RFID scanners placed at entrances and exits. They reduced personnel costs and the density of entry and exit.

A method that was based on reservation was developed in [23]. They monitored each area in the parking space via IoT devices and created a reservation system through a mobile application. The use of IoT devices in car parks also brought along security problems with it. In [24], these security problems were emphasized and elliptical encryption method was proposed. It was demonstrated to be more efficient compared to other encryption methods.

In [25], presents a survey conducted on the needs of drivers in terms of smart services of parking infrastructures. Parking monitoring, parking reservation and recent trends that would benefit both the drivers and the parking operators were discussed. Optimization for parking spaces was studied in [26]. They recommended that parking areas be used more efficiently through Wireless Sensor Networks (WSN) by grouping vehicles based on their size and placing them in appropriate parking spaces.

With the development of IoT technology, WSNs have started to be used more widely in smart cities. Source localization and target tracking are among the most challenging problems for WSNs. There is an increasing number of researches that produce solutions to these problems using machine learning methods. In the study conducted in [52], they proposed SVM and twin SVM (TWSVM) methods to minimize the cost of source localization and target tracking for WSNs. In their experiments to determine the event detection zone, they showed the cost-reduction efficiency of the method they proposed. However, the experiments were made assuming a single event zone. They stated that they will apply the k-means clustering method in the future for multi-source studies. In the study conducted in [53], they tried to get higher efficiency by using machine learning methods from lower cost sensors in order to reduce the cost of locating vehicle fleets in smart cities. They used the Global Positioning System (GPS) as well as the more cost-effective Extended Kalman Filter (EKF) than Inertial Navigation Systems (INS). However, EKF performance depends on the dynamic changes of the vehicle and can vary rapidly due to environmental changes. Limiting the disadvantages of EKF, they proposed a robust and low-cost approach using EKF and SVM to reliably predict vehicle position. In the results they obtained, they achieved an improvement close to 94% in position accuracy compared to the results of EKF. In addition, their experimental results are slightly better than what they did with Random Forest Regression (RFR). In the study conducted in [54], a distributed scheme using WSN has been developed to detect a nuclear radioactive source in order to ensure public safety in smart cities.

With the development of machine learning and deep learning techniques in recent years, the detection of parking spaces through images has become one of the common areas of study. The biggest problem here is that there is no big data set on which a training would be carried out. In the study conducted in [27], a data set called PKlot was created from 65.899 images recorded from three different camera angles and shared for studies. In [28] and [36] a deep learning-based CNN model was proposed for visual detection of parking spaces. It was compared with AlexNet to show the effectiveness of the model proposed. In [29], SVM was used together with CNN to increase the performance of parking spot detection through images. Features extracted with the CNN method were grouped by the SVM method. Higher accuracy was obtained in this study, which was carried out in an outdoor environment. In the study conducted in [30], parking areas were monitored by a large scale zoom lens and motorized cameras. With the deep learning technique proposed, vehicles entering the parking space could be matched with the license plate number to be tracked together with the parking spaces. In the studies conducted in [31] and [32], deep learning methods were used to determine the parking space. They obtained the images used to detect the parking space from satellite and UAV cameras. In [33], image data in parking spaces were transformed into meaningful data with CNN and transferred to cloud-based servers. A mobile application to access this information was developed and parking spaces were predicted.

In the study conducted in [34], the detection of empty parking spaces was accomplished through sensors. They used an artificial neural network to predict parking spots. In [35], park occupancy data collected from two cities, San Francisco, USA and Melbourne, Australia were analyzed. Various prediction mechanisms were used for the parking space occupancy rate and the relative strengths of different machine learning methods were compared. In addition, factors affecting the car park occupancy rate were determined. In [37], potentially interesting trends and events were endeavored to be detected by using automatic clustering and anomaly detection techniques through parking data collected from the city of San Francisco. In [38], a hybrid approach consisting of LSTM and genetic algorithm to predict parking areas in the short term was mentioned.

In Table 1, a summary of the relevant studies grouped according to their distinct features is given in order to better understand.

When the studies are examined, the use of IoT devices in the solution of the parking space problem and deep learning methods demonstrates its effectiveness. There are important studies in determining the empty parking spaces0 in particular, with deep learning methods through images. It is possible to examine the solution suggestions developed in three groups. Reservation and algorithm-based approaches can be examined in the first group, studies on determination of empty parking spaces in the second group, and prediction of empty parking spaces in the short term in the third group.

TABLE 1. Outline of relevant studies.

	Ref.	Problem	Technique	Location	Accuracy
Parking Guidance	[11]	Energy consumption of search algorithms	IoT, Binary Search Tree	Middle East Technical University	-
	[13]	Parking search	Static and Dynamic Vehicle Guidence	-	-
	[14]	Parking search	Smart Parking Algorithm	-	-
	[15]	Parking search	Booking Algorithm	Boston University	-
	[23]	Parking search	IoT, Booking Algorithm	-	-
Parking Lot Optimization	[16]	Parking space optimization	Genetic Algorithm	Shopping mall	-
	[26]	Parking space optimization	IoT, WSN (Wireless Sensor Network)	-	-
Smart Parking architecture	[17]	Parking Search	SVPS, VANET, Cloud	Beijing, China	-
	[18]	Smart Parking Scheme for Large Parking Lots	SPARK, VANET	Conestoga mall parking lot, Canada	-
	[19]	Parking Search	IoT, Cloud	-	
-	[20]	Search for a parking spot	Mobile Phone	-	% 98.00
-	[22]	Personnel Cost Reduction	RFID	Parking Entry and Exit	-
Parking lot occupancy detection	[28]	Empty parking spot detection	CNN, AlexNet,	-	% 90.13
	[36]	Empty parking spot detection	CNN, AlexNet	-	% 90.00
	[29]	Empty parking spot detection	CNN + SVM	-	% 99.7o
	[30]	Parking lot monitoring	AlexNet, mAlexNet	-	% 98.12
	[31]	Object Detection	FCN (FullyConvolutional Network)	United Arab Emirates	% 85.00
	[32]	Car Detection	Mean-Shift Algorithm + CNN + SVM	-	% 93.60
	[33]	Empty parking spot detection and publish	CNN, IoT, Cloud	-	-
	[34]	Smart occupancy detection for road traffic parking	DELM (Deep Extreme Learning Machine)	-	% 91.25
	[35]	Parking occupancy prediction	Regression Tree (RT), Neural Networks (NN), Support vector regression (SVR)	San Francisco, ABD and Melbourne, Australia.	-
-	[37]	Anomaly Detection in Urban Car Parking	SVM	San Francisco, ABD	-

III. DEEP LEARNING BASED CAR PARK OCCUPANCY PREDICTION

Due to the large number of vehicles and the lack of sufficient number of car parks in large metropolitan areas, the time, energy and cost of drivers who need a parking lot to search for parking spaces are increasing day by day. It is predicted that this situation also increases the traffic density. In the architecture proposed for this problem, dynamic data collection is performed and future predictions are made using deep learning methods. In addition, the communication of the created model with mobile devices is done in an IoT-based manner. In this way, users will be able to see which parking lot is available at which time on mobile devices and make their plans within the framework of the smart city concept by using this architecture. In fact, the proposed architectural resemblance is very similar to the weekly weather forecast. Most users create weekly plans based on the weather forecast. Thanks to this architecture, users will be able to easily make weekly parking plans. Therefore, time, energy and cost efficiency will increase.

"Deep learning", first used by Igor Aizenberg *et al.*, is a subfield of machine learning and is based on Artificial Neural Networks (ANNs). The difference of deep learning from ANN is the hidden layers in its structure. Successive layers take the output of the previous layer as input and its structure is based on learning the representation of data [39]–[41]. When deep learning techniques are used with very large data, they yield better results than traditional data processing methods [39], [40], [42], [43]. Problems related to prediction of occupancy of a parking space require methods that will provide high accuracy performance. Otherwise, it is not preferred much because it does not reach a sufficient reliability level. Deep learning methods are therefore a more suitable option for the big data generated. In the methodology proposed, a data set was created from the data collected from the parking spaces for the deep learning model. The learning algorithm

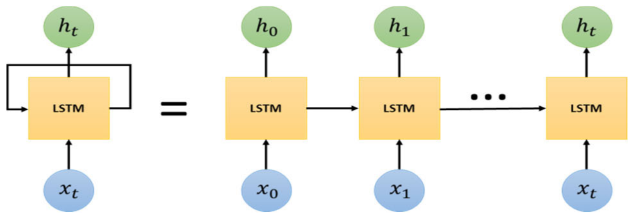

FIGURE 1. LSTM recurrent network.

was implemented with the support of KERAS Tensorflow, one of the Python libraries [44].

A. LSTM DEEP LEARNING ARCHITECTURE

The Long short-term memory (LSTM) network is an extension of the recurrent neural network (RNN) network used in the deep learning field [8], [45]. Unlike standard feed forward networks, LSTMs have feedback links. Sepp Hochreiter and Juergen Schmidhuber developed the LSTM in 1997 to solve the vanishing gradient problem found in RNN [47]. LSTM, which was later organized and popularized with the contribution of many people, now has a wide range of uses [46].

LSTM networks are well suited to classify, process, and make predictions based on time series data because there may be unknown time delays between significant events in a time series.

The LSTM cell contains these three gates:.

- The entrance gate; it controls the flow of input activations to the memory cell.
- The exit gate; it controls the output flow of cell activation.
- The forget gate; it filters the information in the input and previous output and decides which to be remembered or forgotten [48].

Besides the three gates, the LSTM cell contains cell update, which is the tanh layer, usually part of the cell state.

In each LSTM cell, three variable enter the cells:

- X_t, current input,
- h_{t-1}, previous output,
- C_{t-1} previous cell status

On the other hand, two variables come out of the cell:

- h_t current output
- C_t current cell status

Figure 2 shows the structure of the LSTM model. To implement the LSTM repetitive network, the LSTM cell must first be applied.

$$f_t = \sigma\left(W_f\,[h_{t-1}, X_t] + b_f\right) \quad (1)$$
$$i_t = \sigma(W_f\,[h_{t-1}, X_t] + b_i) \quad (2)$$
$$\tilde{C}_t = tanh(W_c\,[h_{t-1}, X_t] + b_C \quad (3)$$
$$C_t = f_t * C_{t-1} + i_t * tanh(W_c\,[h_{t-1}, X_t] + b_C \quad (4)$$
$$o_t = \sigma(W_o\,[h_{t-1}, X_t] + b_o) \quad (5)$$
$$h_t = o_t * tanh(C_t) \quad (6)$$

Variables:

- $X_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in \mathbb{R}^h$: forget gate's activation vector
- $i_t \in \mathbb{R}^h$: output gate's activation vector
- $h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{C}_t \in \mathbb{R}^h$: cell input activation vector
- $C_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{hXd}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters, which need to be learned during training where the superscripts d and h refer to the number of input features and number of hidden units, respectively.

First step of the LSTM cell is to decide which information is to be thrown out of cell status. This decision is taken by a sigmoid layer called "Forget Gate Layer" (f_t). It checks the X_t and h_{t-1}, then assigns C_{t-1} an output value, as a cell status between 0 and 1. If the output value is 1, it expresses "Forget This Fully", and i fit is 0, it means "Get Rid of This Fully".

The next step is to decide what new information to be stored in the cell state. It consists of two parts. First, a sigmoid layer called the "input layer" decides which values to update. Next, a tanh layer creates a new candidate value vector, $\tilde{C}_t$ can be added to the new state.

In the next step, to update the state, i_t and $\tilde{C}_t$ are combined. This scales the decision on how much to update each value, resulting in a candidate value. The previous decision is forgotten by multiplying C_{t-1} by f_t.

Finally, it is decided what will be sent to the output. This output depends on the cell state. But it may be the filter version. First, a sigmoid layer is run that decides which parts of the cell state the output will be sent to. The output of the sigmoid is multiplied by the new value between (+1) and (−1) generated from the cell state C_t by the o_t tanh layer.

B. ISTPARK DATASET

Istanbul Metropolitan Municipality (IMM) open data portal is a free portal where data published from municipalities and surrounding organizations can be accessed. It offers free access to many data sets under the titles Economy, Disaster Management, Energy, Life, Governance, Human and Environment. These data can be accessed with different methods such as xml, xlsx, csv and api. In addition, these data have Istanbul Metropolitan Municipality Open Data License and consist of public sector information licensed under 4.0 International (CC BY 4.0) [9].

In Figure 3, the locations of the car parks in Istanbul, from which we can obtain data instantly, are marked on the map. The ISTPARK data set is composed of approximately 4.5 million parking space data collected from 731 parking spaces at different periods over a year, and the data are still being collected. Data collected is stored on SQL Server 2014. Details of the parameters of the data set are given in Table 2.

IV. MODEL PROPOSED

With the development of deep learning methods in smart cities, parking space solutions that were earlier used are becoming obsolete. As resources become more and more

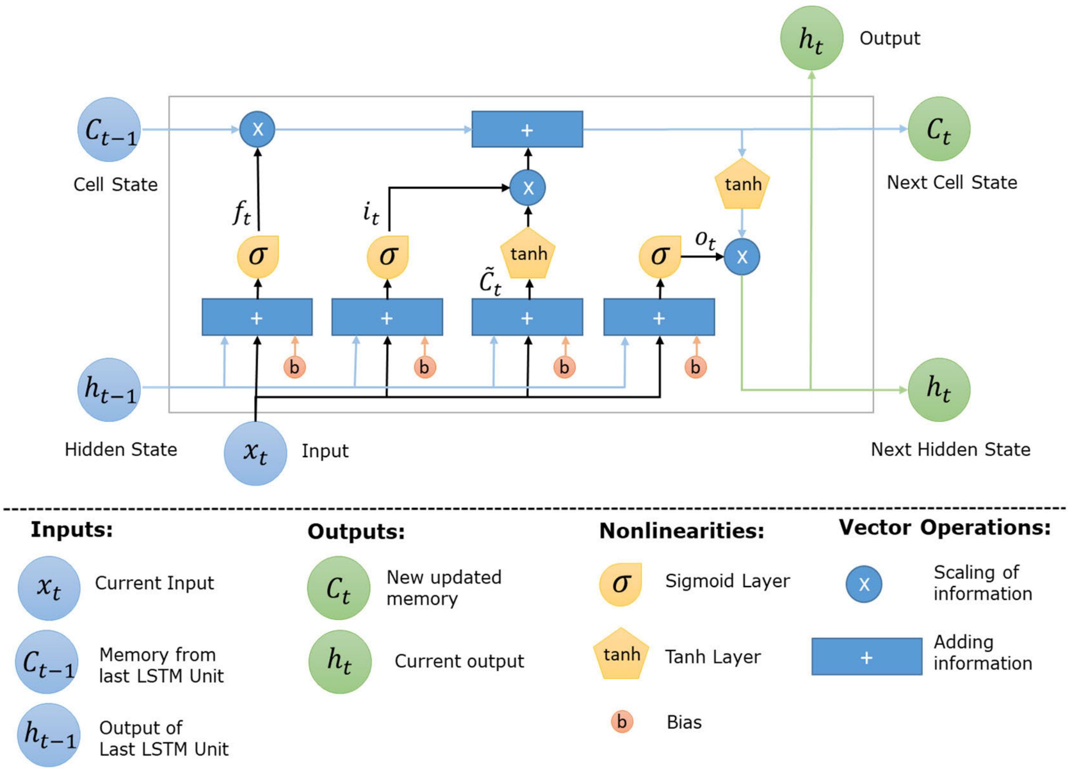

FIGURE 2. Representation of a LSTM model.

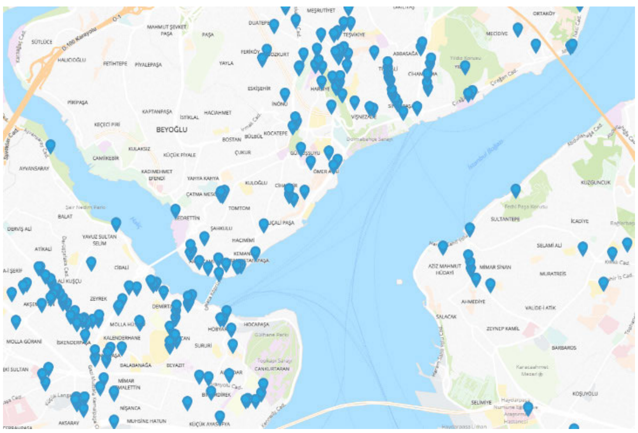

FIGURE 3. Locations of carparks in Istanbul.

limited, the need for innovative solutions is constantly increasing.

In this study, a new deep learning and cloud-based architecture that provides real-time and forward prediction has been developed in solving the problems related to search for a parking space. A web-based and mobile compatible application "ISTPARK" has been developed for users to find the most suitable parking space. Thus, the user will save energy and time with an application that dynamically predicts the occupancy rates of the parks where they want to go in periods of day and hour. The interface design of the application is shown in Figure 4.

TABLE 2. Ispark dataset parameters and descriptions.

Parameters	Data Type	Notes
CarparkID	Integer	Contains carpark ID info, is a Unique field
CarparkName	String	Contains carpark name info
Latitude	Double	Latitude info of carpark
Longitude	Double	Longitude info of carpark
Capacity	Integer	Total capacity of carpark
EmptyCapacity	Integer	Empty capacity of carpark
CarparkType	String	Type of carpark (On-Street, Indoor, Outdoor)
County	String	County carpark is located in
DateofUpdate	Date-time	Date and time of information update
BusinessHours	String	Business hours of carpark
FreeParkingMin	Integer	Duration of free parking (minutes)
MonthlySubscriptionFee	numeric	Monthly Subscription Fee
Address	String	Full address of carpark
AreaPolygon	Array[double][double]	Polygon info of carpark
Distance	Double	Distance to carpark
Tariff	List<>	Information on Tariff (string) & Fee (Double)
Location	String	Information on location carpark is in. One or more carparks are/may be bound in a location.

In this study, the city of Istanbul was used as the pilot region. It is significant that Istanbul was chosen as the pilot region because of it has the most crowded population and densest traffic volume of the country. There is an earthshaking

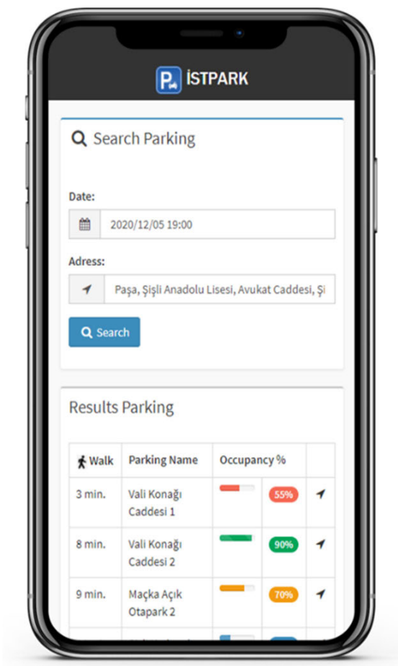

FIGURE 4. ISTPARK mobile interface.

park-finding problems with its population increasing every year passed.

The application interface is designed to be user friendly. The application was made more useful; it just asks for the date and address. The ''Date'' value refers to the date on which user wants to check the parking space occupancy information. ''Address'' value is the address of the location user wants to go to.

When the user presses the ''search'' button, the system suggests the most suitable parking space among the parking spaces close to the target location with smart criteria listing. These smart criteria are ''walk'' and ''occupancy''. ''Walk'' is the walking time between the car park and the target location. ''Occupancy'' is the occupancy or actual state of parking spaces near the target location. A list of car parks with at least one free parking space is listed. The list is in ascending order of walking time.

If the user chooses a future date, the Long short-term memory (LSTM) model is used for the occupancy rate of these parking spaces. LSTM offers an extraordinary advantage in predicting time series where traditional linear methods may be difficult to adapt to multivariate or multiple input prediction problems. It can model problems with multiple input variables almost seamlessly.

LSTM deep learning method was used by using 8 different futures over 4.5 million data from ISTPAK. In this way, 99.57% of the future occupancy rates of the car parks in the short and medium term have been predicted hour by hour. The database is constantly growing dynamically. The success of deep learning methods on big data has been dynamically compared. As a result of this comparison, a model is created from the LSTM deep learning method that gives the highest success rate. LSTM's dynamic learning model is updated every 30 thousand data (approximately once a week). Thus, as the data increases, it is predicted that the results obtained will approach 100% more and get more precise results. In addition, as the data increases, it will be more easily adapted to changing conditions. In addition, the error rate will approach zero. In the study, three different machine learning methods were used together with LSTM, and LSTM gave the most stable result among them. An objective dataset was created by receiving the data directly from the official institution, thus contributing to the smart city issue as a result of analyzing the created dataset with the deep learning method. With the model obtained here, data is obtained in real time with JSON format on mobile devices and served as a service to the end user. In this way, users coming from inside or outside of Istanbul will be able to easily make weekly plans and benefit from the smart city concept by using this architecture. In addition, users will be able to save time, energy and cost.

A. DEPLOYMENT MODEL

The application architecture is based on ''Parking Finder Api'', ''LSTM Based Service'', ''Cloud based - Smart Parking Server (CB-SPS)'' and ''IBB Open Data Portal (ODP)''. Parking Finder Api receives requests from the application interface and returns the most suitable parking list. All car parks within 2 km of the destination address are listed. For each parking space, a request is made directly to the CB-SPS (Cloud based - Smart Parking Server) Server or LSTM based service depending on the date. If the request for the occupancy rate of the parking space is in real time, the data of each parking space is instantly received from the CB-SPS server. If the request date is for a future date, the request is sent to the LSTM Based Service. Within this service, LSTM predicts the parking space occupancy rate with the deep learning model and returns the result. The collection of data from the IBB open data portal is described in detail in the data set creation section of ISTPARK. Figure 5 shows the deployment model of the method proposed.

B. FLOW DIAGRAM OF PROPOSED MODEL

The steps of the method proposed are as follows;

1. The user enters into the application the address and the date s/he wants to travel. If the date s/he entered is past date, s/he will receive an error message.
2. If the target date is equal to the current date or is a future date, all car parks within a diameter of n meters of the destination address are searched.
3. For each car park searched for, first of all, action is taken by looking at the target date. If the target date is equal to the current date, the information of the car parks with at least one empty parking space is received in real time. In other cases, car parks searched for are inserted into the

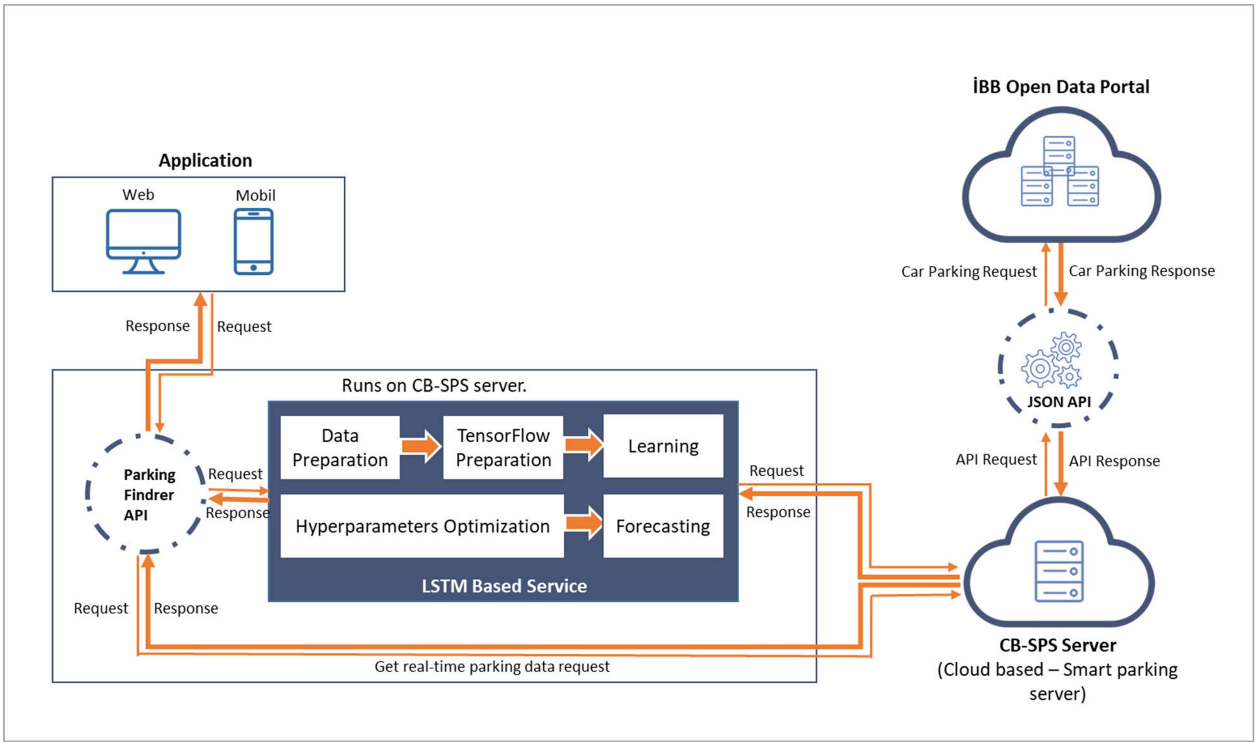

FIGURE 5. Proposed model deployment model.

LSTM model and a prediction is made for the occupancy rate. Parking space information on car parks with at least one empty parking space is received.

4. The parking information received is recommended to the user based on the walking distance between the car park and the target address and the occupancy rate.

The flow chart of the method proposed is given in Figure 6. Where A is the target address, t is the target time, and n is the current time. X_j is the information of car parks within n meters of the destination address. c is the number of car parks involved. K represents the total capacity of each car park, E represents the empty capacity and R occupancy rate of each car park. d stores information on car parks with at least one free parking space near the destination address. LSTM Model is the prediction model for the relevant car park and time.

The pseudo code of the method proposed in Algorithm 1 was given. IBB Map API has an important role here. The list of car parks close to the target address is obtained through this api. Then the code works for each parking lot found.

"GetParkingCapacity" and "GetParkingInstantCapacity" functions query directly on CB-SPS. The "LSTMPrediction" function is the intermediate layer that makes predictions for the respective car park. According to the date information received, it predict the occupancy rate of the car parking by looking at the historical data of the relevant parking lot and returns it in response. In the next section, the prediction of the car parking occupancy rate, accuracy performance and error rate were given in detail.

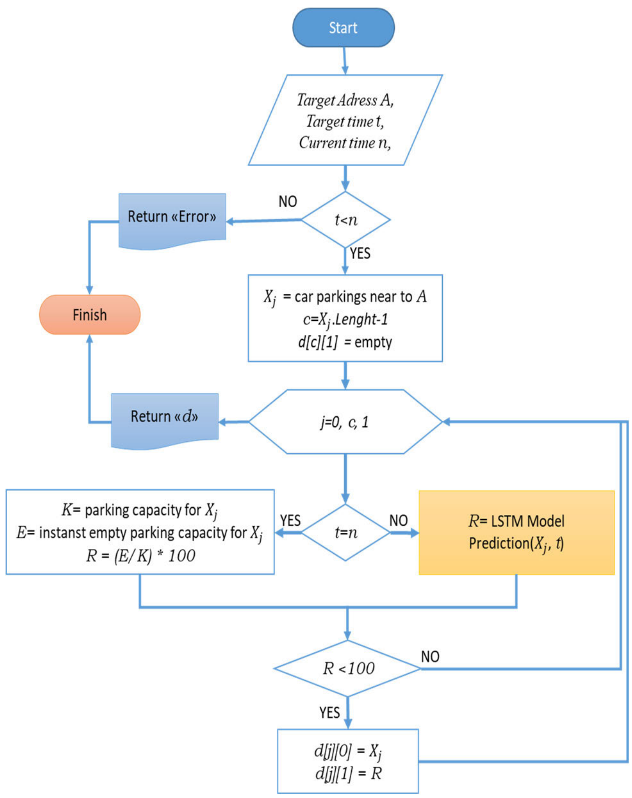

FIGURE 6. Flow diagram of proposed model.

It was also compared with other successful models used in time series estimation and evaluated.

Algorithm 1: The Prosed Model	**#of Times**
function GetParkingCapacity(*ParkID*)	-
//Connect CB-SPS Server and Run SQL Statement	-
return " select Capacity from tbl_Park where id = ParkID "	1
function GetParkingInstantCapacity (*ParkID*)	-
//Connect CB-SPS Server and Run SQL Statement	-
return " select top 1 EmptyCapacity from tbl_Park_Detail where id = ParkID order by RecordDate desc "	1
main:	-
//map: IBB Map API	-
Input: A: Target Adress,	-
t: Target time, n: Current time,	-
key:"Car Park"–keyword, m: 2000 meters,	-
K : Parking Capacity	-
E: Instant empty parking capacity	-
Output: d : List of available car parking	-
$A \leftarrow$ 'Write Target address:'	1
t $\leftarrow$ 'Write Target time:'	1
If ($t < n$) **Then**	n-1
X_j = map. Nearby(A, *key*, m)	1
$c = X_j$.length()	1
For $j = 0$ **to** c-1	n
$R = 0$ // *Parking Occupancy*	1
If ($t = n$) **Then**	n-1
K = GetParkingCapacity(X_j)	1
E =	1
GetParkingInstantCapacity(X_j)	1
$R = (E/K) * 100$	n-1
Else	n
R = *LSTMPrediction*(X_j)	-
End if	n-1
If (R <*100*) **Then**	1
d. Add(X_j, R)	-
End If	-
End For	1
return d	n-1
Else	1
return "error"	-
End if	

The computational complexity of the LSTM is $O(N_1(H^2 + HN_0))$, where N_1 is the number of features in inputs, H is the number of the hidden units, and N_0 is the number of outputs [41]. The computational complexity of operations performed in Jupiter Notebook has been computed as $O\ (n)$.

TABLE 3. Model parameters.

Parameter	Value / Setting
Train size	%70, %80, %90
Test size	%30, %20, %10
Scaling	MinMaxScaler
Epochs	100
Activations	Tanh
Optimizer	Adam
Loss	mean_squared_error
Metrics	MAE, MSE, RMSE, Accuracy
Batch size	20
Verbose	1

C. MODEL TRAINING AND EVOLUATION

In this study, Keras, a Neural Network Library, was used for modeling. Jupiter Notebook was chosen as the compiler program. Model experiments were carried out on a computer with Intel (R) Core (TM) i7-8650U processor (maximum speed: 2.11 GHz), 16 GB ram and Intel (R) UHD Graphics with 620 graphic card.

The parameters of the models applied are given in Table 3. The data set created for training the models consists of 17 different parameters. However, the fixed values and parameters for each car park do not affect the performance of the model. List of parameters used for model training are:

1. Datetime: Is the date of Time Series. The format is yyyy-mm-dd HH:mm.
2. Capacity: Is the total parking space capacity of the carpark.
3. Empty Capacity: The number of vacant parking spaces in the carpark.
4. Location Density Rate: Is the average occupancy rate of the carparks in the same region at time t.
5. Day/Time: What time of day
6. Week/Day: What day of the week
7. Public Holiday: Indicates whether the related day is a public holiday.
8. Carpark occupancy: Is the info on occupancy rate of the carpark.

In order to show the efficiency and performance of the LSTM model proposed in this section, it was compared to the success rates of Random Forest, SVM and ARIMA methods. It is an important factor in choosing Random Forest and SVM as comparison models is that it is a widely used machine learning method and its successful performance in the analysis of time series for the ARIMA.

There are metrics that evaluate the success of predictive models created using machine learning or deep learning models. In this study, the success of the models was evaluated with Root Mean Square Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE) and Accuracy metrics and the model yielding the best results was determined. The RMSE metric is used for the degree of deviation of the model prediction. It takes a value between 0 and $+\infty$. The closer it is to 0, the lower the deviation of the model prediction [49].

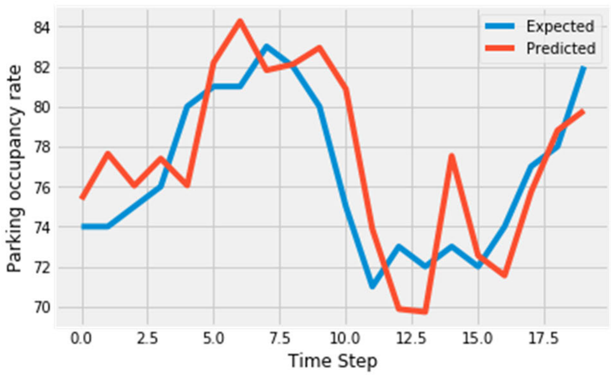

FIGURE 7. Random forest prediction.

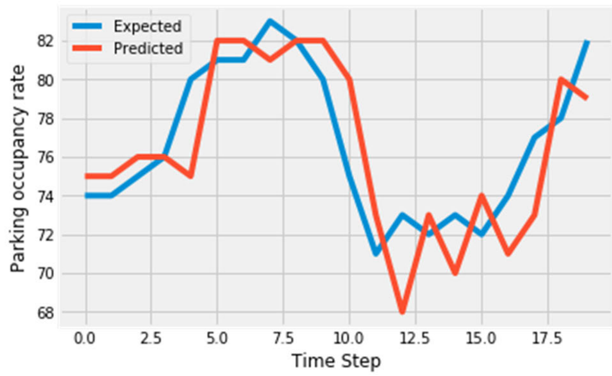

FIGURE 8. SVM prediction.

The expression of RMSE is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \dot{y})^2} \tag{7}$$

MSE depicts how close a regression curve is to a set of points. It is generally used in regression problems, that is, prediction of continuous values [50]. The expression of MSE is:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \dot{y})^2 \tag{8}$$

The MAE measures the average magnitude of errors in a range of predictions. The MAE value for accurate predictions is low [51]. The expression of MAE is:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \dot{y}| \tag{9}$$

Accuracy is the difference between the true rating and the value predicted.

In order to measure the quality of LSTM, SVM, RF and ARIMA models more accurately, training and testing of each model was carried out in three different scenarios. In the first scenario, 70% of each model dataset was trained and 30% of it was tested. In the second scenario, the models were trained with 80% of the dataset and tested with 20%. In the last scenario, 90% of the data set was used for training and 10% for the test. Table 5 shows the results obtained in these three scenarios. In addition, LSTM, SVM, RF and ARIMA models were run 100 epochs for all scenarios. The performances of the models were evaluated with MAE, MSE, RMSE and Accuracy metrics.

Figure 7 shows the prediction performance of the parking space occupancy rate of the Random Forest model. Blue series represent actual values and red series represent predicted values. While random forest makes good predictions in the linear moving series, it seems that it makes worse predictions in turns and at the maximum and minimum points.

Figure 8 shows the prediction performance of the parking space occupancy rate of the Support Vector Machine model.

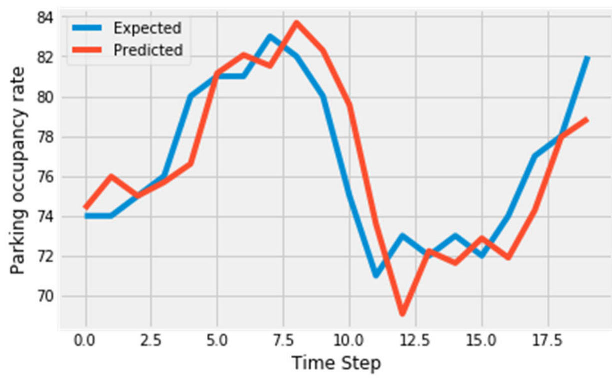

FIGURE 9. ARIMA prediction.

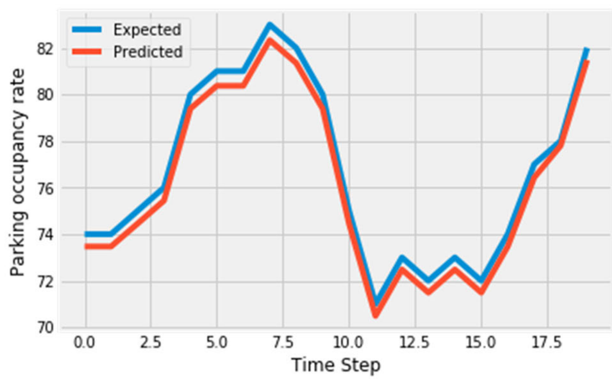

FIGURE 10. LSTM prediction.

While SVM model makes good predictions in linear moving series like Random forest, it seems that it makes worse predictions in turns and at maximum and minimum points.

Figure 9 shows the prediction performance of the parking space occupancy rate of the ARIMA model. ARIMA is already a frequently used method in the analysis of time series. In this study, it yielded better prediction results compared to that of SVM and Random forest models. However, it seems to be far behind the LSTM model.

TABLE 4. Performance comparison for different features.

Feature(s)	Accuracy (%)			
	LSTM	RF	SVM	ARIMA
Capacity, Density, Time, Day,Holiday	99.57	89.34	87.01	90.22
Capacity, Density, Time, Holiday	96.91	83.22	81.95	88.70
Capacity, Density, Day, Holiday	96.06	82.90	81.47	88.81
Capacity, Time, Day, Holiday	98.90	87.43	86.60	90.12
Density, Time, Day, Holiday	99.02	88.91	86.90	89.01
Capacity, Density, Time, Day	98.70	87.01	86.30	88.80
Time, Day, Holiday	99.23	89.12	86.94	89.45
Capacity, Day, Holiday	96.01	82.77	80.98	87.12
Capacity, Density , Holiday	95.42	83.76	81.67	86.59
Capacity, Time, Holiday	95.01	81.18	79.01	85.90
Capacity, Density, Time	94.91	80.79	78.77	86.34
Density, Time, Day	98.91	87.40	86.70	92.90
Capacity, Time, Day	97.83	86.92	83.12	89.15
Day, Holiday	96.01	83.18	79.92	87.80
Capacity, Holiday	95.15	83.44	83.12	87.06
Capacity, Density	96.05	83.10	81.70	84.93
Density, Time	96.43	83.45	82.44	86.77
Time, Day	98.01	85.50	82.34	88.70
Time	97.67	84.90	82.01	88.12
Day	97.23	84.48	82.40	87.45
Holiday	96.12	84.12	83.25	86.90
Capacity	96.44	84.55	82.85	87.65
Density	95.90	83.82	83.01	87.20

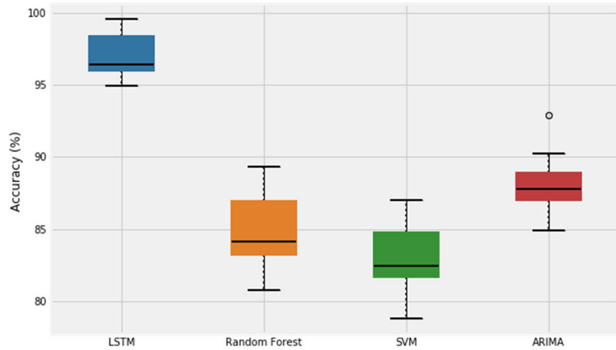

FIGURE 11. Accuracy boxplot.

TABLE 5. Error assessment of models.

Train - Test	Model	MAE	MSE	RMSE
%70 - 30	LSTM	0.92	2.54	1.59
	SVM	2.50	8.50	2.91
	RF	2.47	7.12	2.66
	ARIMA	1.87	7.29	2.70
%80 - 20	LSTM	1.02	3.12	1.77
	SVM	3.50	18.01	4.24
	RF	2.17	6.96	2.63
	ARIMA	1.76	5.08	2.25
%90 - 10	LSTM	1.11	3.18	1.78
	SVM	2.64	9.11	3.02
	RF	1.83	4.55	2.13
	ARIMA	1.43	4.06	2.01

Figure 10 shows the predicted performance of the parking space occupancy rate of the Long short-term memory model that were created. As can be seen in the figure, the LSTM model has succeeded very well in predicting the occupancy rate of the parking space.

When the predictive performances of the models are examined on the figures, it is seen that the LSTM model used for the parking space occupancy rate prediction in the architecture proposed yields much better results.

The combination of different features of LSTM, Random Forest, SVM and ARIMA was run for 100 epochs and the comparison in terms of accuracy is given in Table 3. The capacity given in Table 3 refers to the total parking capacity of the car park. Density refers to the average of the densities of the car parks in the same area. Time refers to the hour in the data set. Day refers to the day in the data set, Holiday takes the value 1 if the relevant day is a public holiday, 0 if not. For each combination, LSTM performed better than other methods.

As can be seen in Table 3, highest accuracy (99.57%) was achieved with using the combination of Capacity, Density, Time, Day and Holiday. Time and Day features seem to be the two most important parameters in prediction performance. It was observed that the lowest accuracy rate was obtained with the Density feature in single combinations. Here, the features to be used are determined based on the highest value. However, as seen in Table 4, even if the number of Features is decreased, predictions are at a level that can be accepted. This also offers the user the opportunity to reduce the number of features by considering different constraints.

In Figure 11, the boxplot of the accuracy values of the models compared is given. It can be seen on the graph that the LSTM model works more stable than other models.

Table 5 shows the error assessment of the models. LSTM model yielded the best results in three different scenarios divided into different training-test data sets. In its best performance, it yielded an MAE value of 0.92, an MSE value of 2.54 and an RMSE value of 1.59. This also shows that the model can accurately predict parking space occupancy.

V. CONCLUSION

Deep learning methods yield successful results in parking space occupancy prediction as in many areas. In this study, a new cloud and deep learning based architecture and a new mobile application are proposed for parking space occupancy prediction. To predict the parking space occupancy rate, a deep learning based LSTM model, which models multivariate and large data sets almost seamlessly, was used. In order to demonstrate the effectiveness of the model, models were created in different parameters and scenarios and compared with SVM, Random Forest and ARIMA models. The results are given in Table 3 and Table 4. When the data set was trained with a combination of Capacity, Density, Time, Day and Holiday, it resulted in an accuracy rate of 99.57%. This result demonstrates that the architecture and application created yielded successful results.

Today, traffic density is the biggest problem of metropolitan cities. Naturally, waiting in front of the car park, exiting from the car park and driving slowly while searching for a parking space are thought to cause this traffic density to increase. In subsequent studies, the effect of car parks on traffic density with different parameters will be examined and a new measure will be presented to reduce traffic density by using deep learning methods. In addition, by realizing reservation-based architecture in parking lots, both traffic density and cost can be reduced. At the same time, the layout of the parking lots can be realized by using deep learning and optimization methods in smart city planning.

REFERENCES

[1] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Implementing lightweight IoT-IDS on raspberry Pi using correlation-based feature selection and its performance evaluation," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2019, pp. 458–469.

[2] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and BOT-IoT attacks traffic identification for Internet of Things in smart city," *Future Gener. Comput. Syst.*, vol. 107, pp. 433–442, Jun. 2020.

[3] T. Lin, H. Rivano, and F. Le Mouel, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017.

[4] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172–70185, 2018.

[5] F. Canitez, D. Çelebi, and E. Beyazit, "Establishing a metropolitan transport authority in Istanbul: A new institutional economics framework for institutional change in urban transport," *Case Stud. Transp. Policy*, vol. 7, no. 3, pp. 562–573, Sep. 2019.

[6] M. Dener, M. A. Akcayol, S. Toklu, and O. F. Bay, "Zamana bağli dinamik en kisa yol problemi için genetik algoritma tabanli yeni bir algoritma," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 26, no. 4, pp. 915–928, 2011.

[7] I. Kok, M. U. Simsek, and S. Özdemir, "A deep learning model for air quality prediction in smart cities," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1983–1990.

[8] Z. Huang, F. Yang, F. Xu, X. Song, and K.-L. Tsui, "Convolutional gated recurrent unit–recurrent neural network for state-of-charge estimation of lithium-ion batteries," *IEEE Access*, vol. 7, pp. 93139–93149, 2019.

[9] İBB, "Dataset," İspark otopark detay bilgileri Web servisi, Tech. Rep., 2020.

[10] F. Al-Turjman and A. Malekloo, "Smart parking in IoT-enabled cities: A survey," *Sustain. Cities Soc.*, vol. 49, Aug. 2019, Art. no. 101608.

[11] B. Kizilkaya, M. Caglar, F. Al-Turjman, and E. Ever, "Binary search tree based hierarchical placement algorithm for IoT based smart parking applications," *Internet Things*, vol. 5, pp. 71–83, Mar. 2019.

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

[13] J. Hanzl, "Parking information guidance systems and smart technologies application used in urban areas and multi-storey car parks," *Transp. Res. Procedia*, vol. 44, pp. 361–368, Jan. 2020.

[14] J.-H. Shin and H.-B. Jun, "A study on smart parking guidance algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 44, pp. 299–317, Jul. 2014.

[15] Y. Geng and C. G. Cassandras, "A new 'smart parking' system infrastructure and implementation," *Procedia Social Behav. Sci.*, vol. 54, pp. 1278–1287, Oct. 2012.

[16] D. Thomas and B. C. Kovoor, "A genetic algorithm approach to autonomous smart vehicle parking system," *Procedia Comput. Sci.*, vol. 125, pp. 68–76, Jan. 2018.

[17] Q. G. K. Safi, S. Luo, L. Pan, W. Liu, R. Hussain, and S. H. Bouk, "SVPS: Cloud-based smart vehicle parking system over ubiquitous VANETs," *Comput. Netw.*, vol. 138, pp. 18–30, Jun. 2018.

[18] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A new VANET-based smart parking scheme for large parking lots," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1413–1421.

[19] Y. Atif, J. Ding, and M. A. Jeusfeld, "Internet of Things approach to cloud-based smart car parking," *Procedia Comput. Sci.*, vol. 98, pp. 193–198, Jan. 2016.

[20] K.-C. Lan and W.-Y. Shih, "An intelligent driver location system for smart parking," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2443–2456, Apr. 2014.

[21] M. Bachani, U. M. Qureshi, and F. K. Shaikh, "Performance analysis of proximity and light sensors for smart parking," *Procedia Comput. Sci.*, vol. 83, pp. 385–392, Jan. 2016.

[22] Z. Pala and N. Inanç, "Smart parking applications using RFID technology," in *Proc. 1st Annu. RFID Eurasia*, Sep. 2007, pp. 1–3.

[23] CIEC, "The essential chemical industry—Online," in *Proc. Int. Conf. Internet Things Appl.*, 2016, pp. 266–270.

[24] I. Chatzigiannakis, A. Vitaletti, and A. Pyrgelis, "A privacy-preserving smart parking system using an IoT elliptic curve based security platform," *Comput. Commun.*, vols. 89–90, pp. 165–177, Sep. 2016.

[25] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, "Smart parking solutions for urban areas," in *Proc. IEEE 14th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2013, pp. 1–6.

[26] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: An application of optical wireless sensor network," in *Proc. Int. Symp. Appl. Internet Workshops (SAINT)*, Jan. 2007, p. 66.

[27] P. R. L. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot—A robust dataset for parking lot classification," *Expert Syst. Appl.*, vol. 42, no. 11, pp. 4937–4949, Jul. 2015.

[28] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Syst. Appl.*, vol. 72, pp. 327–334, Apr. 2017.

[29] D. Acharya, W. Yan, and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," in *Proc. CEUR Workshop*, vol. 2087, 2018, pp. 33–40.

[30] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj, and K. Liu, "An edge based smart parking solution using camera networks and deep learning," in *Proc. IEEE Int. Conf. Cognit. Comput. (ICCC)*, Jul. 2018, pp. 17–24.

[31] M. Napiorkowska, D. Petit, and P. Marti, "Three applications of deep learning algorithms for object detection in satellite imagery," in *Proc. IGARSS*, no. 1, 2018, pp. 4839–4842.

[32] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, Mar. 2017.

[33] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand, "Parking-stall vacancy indicator system, based on deep convolutional neural networks," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 655–660.

[34] S. Y. Siddiqui, M. A. Khan, S. Abbas, and F. Khan, "Smart occupancy detection for road traffic parking using deep extreme learning machine," *J. King Saud Univ. Comput. Inf. Sci.*, pp. 1–7, Feb. 2020.

[35] Y. Zheng, S. Rajasegarar, and C. Leckie, "Parking availability prediction for sensor-enabled car parks in smart cities," in *Proc. IEEE 10th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2015, pp. 7–9.

[36] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 1212–1217.

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

[38] L. Xiangdong, C. Yuefeng, C. Gang, and X. Zengwei, "Prediction of short-term available parking space using LSTM model," in *Proc. 14th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2019, pp. 631–635.

[39] O. Brun *et al.*, "IoT attack detection with deep learning to cite this version?: HAL Id?: Hal-02062091 IoT attack detection with deep learning," in *Proc. ISC Secur. Working*, Londres, U.K., Feb. 2019.

[40] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *J. Sensors*, vol. 2017, pp. 1–13, Aug. 2017.

[41] S. Cakir, S. Toklu, and N. Yalcin, "RPL attack detection and prevention in the Internet of Things networks using a GRU based deep learning," *IEEE Access*, vol. 8, pp. 183678–183689, 2020.

[42] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the Internet of Things," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 39–58, Nov. 2018.

[43] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, p. 1977, Apr. 2019.

[44] *Keras Documentation: About Keras*. Accessed: Jan. 11, 2020. [Online]. Available: https://keras.io/about/

[45] H. Yan, Y. Qin, S. Xiang, Y. Wang, and H. Chen, "Long-term gear life prediction based on ordered neurons LSTM neural networks," *Measurement*, vol. 165, Dec. 2020, Art. no. 108205.

[46] D. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, and Z. Sun, "Well production forecasting based on ARIMA-LSTM model considering manual operations," *Energy*, vol. 220, Apr. 2021, Art. no. 119708.

[47] X. H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, p. 1387, Jul. 2019.

[48] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 5–9.

[49] X. Ping, F. Yang, H. Zhang, W. Zhang, G. Song, and Y. Yang, "Prediction and optimization of isentropic efficiency of vortex pump under full operating conditions in organic rankine cycle waste heat recovery system based on deep learning and intelligent algorithm," *Sustain. Energy Technol. Assessments*, vol. 42, Dec. 2020, Art. no. 100898.

[50] V.-H. Nhu, N.-D. Hoang, H. Nguyen, P. T. T. Ngo, T. T. Bui, P. V. Hoa, P. Samui, and D. Tien Bui, "Effectiveness assessment of keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area," *Catena*, vol. 188, May 2020, Art. no. 104458.

[51] S. Abirami and P. Chitra, "Regional air quality forecasting using spatiotemporal deep learning," *J. Cleaner Prod.*, vol. 283, Feb. 2021, Art. no. 125341.

[52] I. Belhajem, Y. B. Maissa, and A. Tamtaoui, "Improving vehicle localization in a smart city with low cost sensor networks and support vector machines," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 854–863, Aug. 2018.

[53] G. Bovenzi, D. Ciuonzo, V. Persico, A. Pescapè, and P. S. Rossi, "IoT-enabled distributed detection of a nuclear radioactive source via generalized score tests," *Commun. Comput. Inf. Sci.*, vol. 968, pp. 77–91, Dec. 2019.

[54] S. H. Javadi, H. Moosaei, and D. Ciuonzo, "Learning wireless sensor networks for source localization," *Sensors*, vol. 19, no. 3, pp. 1–17, 2019.

H. CANLI was born in Ordu, Turkey, in 1992. He received the B.S. (Hons.) and M.S. degrees in computer engineering from Duzce University, Duzce, Turkey, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer engineering. His research interests include the Internet of Things, cyber security, computer networking, deep learning, machine learning, and data mining. His teaching areas are cloud computing, information and network security, and programming languages at undergraduate degree.

S. TOKLU received the B.Sc. degree in computer engineering from Eastern Mediterranean University, in 2004, and the M.Sc. degree in computer engineering and the Ph.D. degree in electric-electronic education from Gazi University, Ankara, Turkey, in 2007 and 2013, respectively. His research interests include wireless sensor networks, energy optimization, the Internet of Things, deep learning, and smart city.