2022 25th International Conference on Computer and Information Technology (ICCIT)
17-19 December 2022, Cox's Bazar, Bangladesh

2022 25th International Conference on Computer and Information Technology (ICCIT) | 979-8-3503-4602-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCIT57492.2022.10055168

SlotFinder: A Spatio-temporal based Car Parking System

Mebin Rahman Fateha, Md. Saddam Hossain Mukta, Md. Abir Hossain, Mahmud Al Islam, Salekul Islam
Department of CSE, United International University (UIU)
Plot-2, United City, Madani Avenue, Badda, Dhaka-1212, Bangladesh
Email: mfateha171124@bscse.uiu.ac.bd, saddam@cse.uiu.ac.bd, mhossain171125@bscse.uiu.ac.bd, mislam171131@bscse.uiu.ac.bd, salekul@cse.uiu.ac.bd

***Abstract*—Nowadays, the increasing number of vehicles and shortage of parking spaces have become an inescapable condition in big cities across the world. Car parking problem is not a new phenomenon, especially in a crowded city such as Dhaka, Bangladesh. Shortage of parking spaces leads to several problems such as road congestion, illegal parking on the streets, and fuel waste in searching for a free parking space. In order to overcome the parking problem, we develop a spatio-temporal based car parking system namely, *SlotFinder*. We collect the data of 408 buildings those have parking slots from seven different locations. We then cluster these data based on time and locations. Later, we train location wise vacant parking spaces by using stacked Long Short-Term Memory (LSTM) based on their temporal patterns. We also compare our technique with the baseline models and conduct an ablation analysis, which outperforms (lower RMSE and MAE of 0.29 and 0.24, respectively) than that of the previous approaches.**

***Index Terms*—Stacked LSTM, Spatio-temporal, Car parking system, Machine Learning**

I. Introduction

Car parking is an emerging problem with the increasing number of vehicles in large cities world wide. Dhaka city is an unplanned city where roads, housing and offices are established without foreseeing the inconvenient future. Therefore, finding a parking space is a common problem created by the increased number of vehicles. Searching for a parking space requires time, effort and extra fuel. A global parking survey by IBM in 2011 shows that 20 minutes is spent on average in searching for a perfect parking space [7]. To solve the problem, we develop a spatio-temporal based car parking model, namely *SlotFinder*.

Dhaka city is one of the most densely populated areas in the world. In 2010, the population of Dhaka city was 14.7 million and in 2021, it became 21.7 million, which reflects a significant increase of population from 2010 to 2021 [3]. With the increasing number of population, the number of registered vehicles is also rising proportionally. Parking poses a serious challenge to the vehicle owners since there is a serious shortage of parking facilities. The parking problem leads to illegal parking on city streets, increasing intense traffic jam and worsening the annoyance of the moving traffic. Thus, the overall parking problem has motivated us to come up with a solution. Towards this direction, we develop a machine learning based spatio-temporal car parking system. We find several location aware applications in the literature [6], [11], [12], [14], [15], which largely apply RNN models to predict in terms of both space and time.

In this study, we collect data from 408 buildings from seven different areas of Dhaka city. Our dataset contains six spatio-temporal features. First, we apply k-Means clustering based on longitude and latitude to group up the regions with similar parking trend. Then, we use the clustering technique with k=7 to make seven clusters and each cluster consists of time intervals of vacant parking spaces in a region. Then, to learn our model the pattern of the parking vacancies, we apply RNN based stacked LSTM model on each cluster to predict vacant parking spaces (departure and arrival time).

In short, in this paper, we have following contribution:

- We build a dataset with available car parking time (i.e, departure and arrival time) and spatial features (i.e., latitude and longitude).
- We develop an efficient spatio-temporal based stacked LSTM for predicting the vacant parking spaces.
- We demonstrate that our model outperforms the baseline approaches.

II. Literature Review

In this section, we present several studies related to car parking systems. Kotb et al. [5] introduced a smart car parking system with static resource scheduling, dynamic resource allocation and pricing models, to optimize the parking system for drivers and parking owners. They combined real time reservation (RTR) with share time reservation (STR).

Gandhi et al. [4] introduced a system that directs information about open and full parking spaces via mobile or web application. This IoT system includes micro-controller and sensor devices with Electric Vehicle (EV)–charging points, which is situated in respective car parking space. Zacepins et al. [21] proposed a smart parking management based on video processing and analysis. In this paper they made a python application for real time parking lot monitoring. For the occupancy detection in parking, they used five classifiers (Logistic Regression, Linear Support Vector Machine, Radial Basis Function Support Vector Machine, Decision Tree and Random Forest). Shao et al. [17] introduced a range based kNN algorithm which is named as Range-kNN. Their proposed

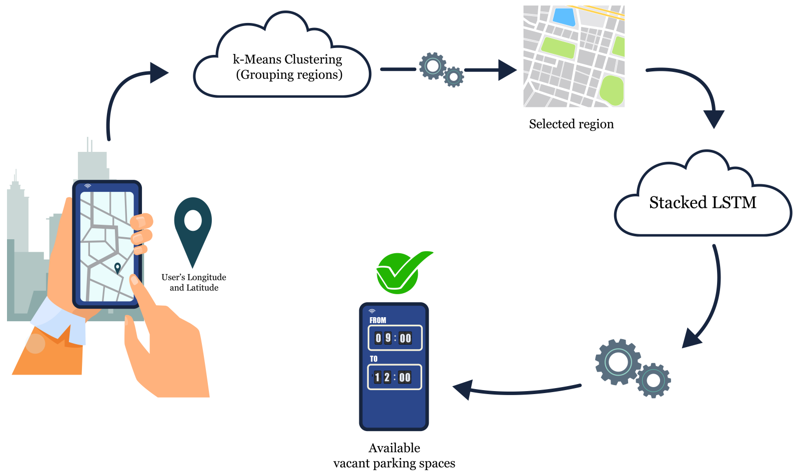

Fig. 1. Methodology of spatio-temporal based car parking system.

algorithm consists of two parts: expansion of query range and search algorithm.

We observe that the majority of the studies consider spaces only where temporal parameters are not taken in consideration while time is a vital factor. Towards this direction, we propose a novel stacked LSTM based car parking solution which considers both space and time.

III. Methodology

In this paper, we mainly work with spatial-temporal dataset. Figure 1 shows different steps of our proposed car parking system. First, we select a location which is ideally a crowded area and where most of the people have their own vehicles. Then, we annotate these areas with starting and departure times. Later, we find different pattern for the starting and departure times based on locations. By observing these patterns, we apply our spatio-temporal based machine learning technique. We briefly present the pipelines of our study.

A. Data Collection

In this section, we first select seven populated areas in Dhaka city for collecting data. In these areas, people have higher percentage (around 45%) [18] of their own vehicles.

TABLE I
Summary of collected data.

Dimension	408x6
Number of areas	7
Average departure time	8:00 AM - 9:00 AM
Average arrival time	5:00 PM - 6:00 PM
Duration of empty parking	9-10 hours
Average number of empty parking	5-10
Average parking spaces	20-25
Number of instances	408

We also choose those areas where parking problem is a common issue due to the increased crowd and less parking facilities. After area selection, the next step is to annotate the buildings to identify the parking spaces by its latitude and longitude. For collecting data, we conduct field visits in seven locations inside Dhaka such as Dhanmondi, Gulshan, Uttara, Mirpur, etc.[1]. We visit several residential buildings in order to get the parking information. We take some relevant information about parking spaces such as if there is any free parking space, for how much time in a day it remains vacant, whether a specific parking space remains occupied and free, etc. We take these information from 408 buildings of seven areas by taking face-to-face survey with building managers.

A summary of collected data is shown in Table I. Frequent departure time of vacant parking spaces is 8:00 am to 9:00 am and arrival time is 5:00 pm to 6:00 pm. Average duration of the vacant parking spaces is 9-10 hours. Each building has 5-10 vacant parking spaces on an average. However, for further research, we share the dataset for public use [2]. Table II shows the number of instances from each area based on average departure and arrival time.

TABLE II
Number of instances per area based on their average departure and arrival time.

Area Name	Departure time	Arrival time	size
Dhanmondi	8:00 AM - 9:00 AM	5:00 PM - 6:00 PM	74
Gulshan	9:00 AM - 10:00 AM	5:00 PM - 6:00 PM	67
Uttara	8:00 AM - 9:00 AM	6:00 PM - 7:00 PM	63
Mirpur	7:00 AM - 8:00 AM	5:00 PM - 6:00 PM	71
Kallyanpur	7:00 AM - 8:00 AM	5:00 PM - 6:00 PM	48
Shyamoli	8:00 AM - 9:00 AM	5:00 PM - 6:00 PM	42
Mohammadpur	8:00 AM - 9:00 AM	5:00 PM - 6:00 PM	43
Total			408

[1]Google map: shorturl.at/lruTZ
[2]https://bit.ly/3yiNvFY

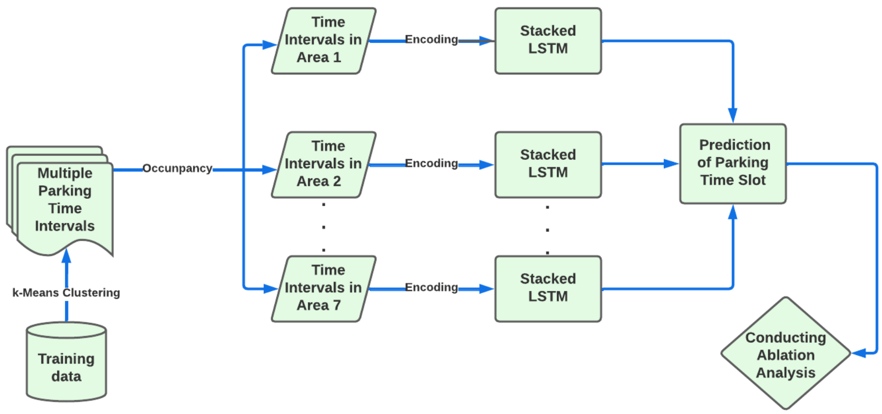

Fig. 2. Architecture of our SlotFinder System.

B. Building Model

In this section, we first describe our system architecture in Figure 2. Then, we present our clustering approach and discuss data preparation techniques. Next, we discuss how we apply our proposed stacked LSTM model over our dataset.

We take the parking information based on the time interval whether a parking space is vacant and store these information in the data set. After that, we apply k-Means clustering on the data set based on location. The availability of free parking space with time is different for each areas in Dhaka city. During our field study, we see some significant differences in parking trends among the seven areas. For example, the average departure time for Dhanmondi area is 8:00-9:00 AM and for Gulshan the average departure time is 9:00-10:00 AM. Since Dhanmondi has become the biggest hub of educational institutes and the majority of the schools start lectures around 8:00-8:30 AM. Therefore, the maximum parking spaces of this area get vacant around 8:00-9:00 AM. On the other hand, many private corporations start their offices in Gulshan and the maximum people who live in Gulshan has office or business around Gulshan. Thus, they leave their home around 9:00-10:00 AM . Before applying our machine learning algorithm, we group similar parking spaces together. Therefore, we apply k-Means clustering method with k=7 as we collect data from seven different areas to group 408 parking spaces. We use longitude and latitude values of each parking space as input feature for the clustering. Figure 3 shows seven clusters with respect to cluster centroids.

Data preparation is important because we prepare data for applying machine learning algorithms over a structured dataset. Data preparation helps to find efficient result with less error. Both numerical and categorical features are present in our dataset. The categorical data cannot be immediately interpreted by machines. In order to process the categorical data further, we transform it into numerical data. In our dataset, departure time and arrival time are in categorical form. We use label encoding to convert these two features into numerical form shown in Figure 4.

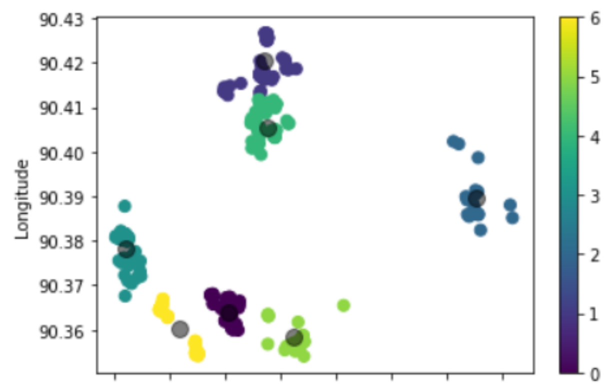

Fig. 3. Clusters visualization based on latitude and longitude.

	Before Encoding			After Encoding		
Departure Time	8:00 AM	9:00 AM	8:00 AM	23	14	23
Arrival Time	5:00 PM	5:30 PM	5:30 PM	17	19	19

Fig. 4. Label encoding of departure time and arrival time.

Data scaling is a data preprocessing technique for numerical features. Data scaling is necessary for obtaining improved performance of many machine learning algorithms, including RNN. For this, various scaling are defined. We use Min-Max [13] scaling to scale our numerical features (range is 0 to 1).

Then, we split our dataset into train and test parts by 65% and 35%, respectively. We train our dataset by using 10-iterations with 10-fold cross validation. Parking event maintains a sequence and we also observe that the buildings in a specific region maintain a sequence of similar parking trend. For this reason, we apply stacked LSTM in each cluster that predicts departure time and arrival time. The basic LSTM model consists of a single hidden LSTM layer followed by a

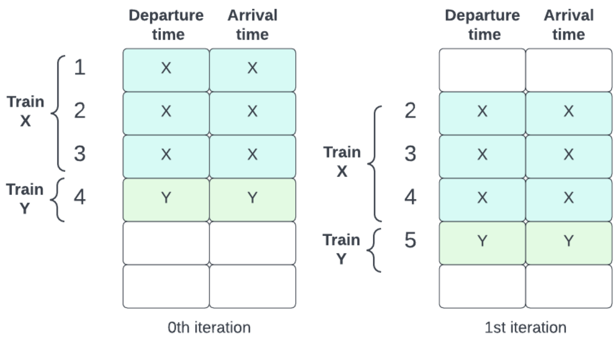

Fig. 5. Re-arranging of trainX and trainY.

conventional feedforward output layer. The Stacked LSTM is a model extension that contains multiple hidden LSTM layers, each of which contains multiple memory cells [8], [16]. Since our desired output predicts both the departure and arrival time, therefore we incorporate the stacked LSTM in our problem.

Our stacked LSTM model consists of several LSTM layers. In stacked LSTM architecture, the output of the first LSTM layer goes as input of the next LSTM layer. In our model, the first LSTM has 64 layers and the second LSTM has 32 layers. We use a dropout of 0.2 followed by a dense layer. In our experiment, we consider time steps to be 3, time steps denote that how previous data should be considered to predict the future parking time. Considering the time steps to be 3 gives us an optimal result. We take previous 3 parking times to predict the 4th parking timing. We split the data as X, Y. In the 0-th iteration, the first 3 values are in X and the 4th value is in Y and so on as shown in Figure 5. In this manner, we re-arrange both our train and test datasets. Figure 6 (a) shows the training loss and validation loss of a single cluster and Figure 6 (b) shows the training loss and validation loss of the seven clusters. We see from the figures that the training loss and validation loss both decrease and remain stable at a point.

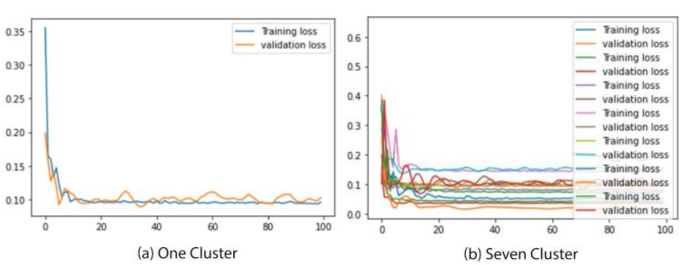

Fig. 6. Training Loss and Validation Loss.

C. Ablation Study

Components of a deep learning network are typically deleted or replaced as part of an experiment called an ablation study to determine how these changes affect the overall performance of the system. The performance of a model may remain stable, become better or get worse when these components are changed. Accuracy can be improved primarily by experimenting with various hyper-parameters like *optimizer, learning rates, loss functions* and *batch sizes*. Altering the architecture of the model has an effect on overall performance. In this study, we demonstrate six case studies by altering different system parameters and response of the system by the changes.

Evaluation Metrics: We apply three metrics to evaluate our proposed model: mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE). Six experiments are conducted as an ablation study, each changing a different components of the proposed Stacked LSTM model. A more reliable architecture with improved performance can be achieved by changing many components. This is accomplished by performing an ablation study on a number of components: batch size, hidden layer, loss function, optimizer, learning rate, activation function and dropout.

1) Ablation Study 1 (Changing Hidden Layers): In our model, we use stacked LSTM. In this stacked LSTM we have 2 LSTM layers. In the first LSTM, we have 64 hidden layers and in the second LSTM, we have 32 hidden layers. To observe the model's performance we change the number of the hidden layers of both the LSTMs. In Table III, we present RMSE, MAE and Val_loss scores in Table III.

TABLE III
ABLATION STUDY BY CHANGING THE HIDDEN LAYERS.

Case Study	Hidden lyer	RMSE	MAE	Val_Loss
1	LSTM1-64 LSTM2- 32	0.30	0.25	0.03
	LSTM1- 128 LSTM2- 64	0.30	0.25	0.01
	LSTM1- 50 LSTM2- 50	0.30	0.25	0.02

2) Ablation Study 2 (Changing Batch Size): The term "batch size" refers to the number of training samples used in a single iteration. To determine the ideal batch size for our proposed model, we experiment with different batch sizes in our study. When we change our bacth size from 64 to 16 which gives RMSE from 0.3 to 0.29. The results are shown in Table IV.

TABLE IV
ABLATION STUDY BY CHANGING THE BATCH SIZE.

Case Study	Batch Size	RMSE	MAE	Val_Loss
2	16	0.29	0.25	0.01
	32	0.30	0.26	0.03
	64	0.30	0.25	0.01

3) Ablation Study 3 (Changing Optimizer): We use *Adam* [22] in our model as optimizer. The model gives us RMSE, MAE and Val_loss scores of 0.29, 0.25 and 0.01, respectively. If we change the optimizer to sgd and Nadam which give us an increase in the scores of RMSE, MAE and Val_loss shown in Table V. SGD does each iteration using a single sample, or a batch size of one. The sample is chosen and randomly shuffled in order to carry out the iteration. Nadam [19] is an optimizer combination of adam and RMSprop with Nesterov momentum.

TABLE V
ABLATION STUDY BY CHANGING OPTIMIZER.

Case Study	Optimizer	RMSE	MAE	Val_Loss
3	sgd	0.35	0.31	0.04
	adam	0.29	0.25	0.01
	nadam	0.30	0.25	0.01

4) Ablation Study 4 (Changing Learning Rate): We use learning rate of 0.01 which gives RMSE, MAE and Val_loss scores of 0.29, 0.24 and 0.01, respectively. If we replace the learning rate with 0.001 and 0.0001, the scores of RMSE, MAE and Val_loss increase. The results are shown in Table VI.

TABLE VI
ABLATION STUDY BY CHANGING LEARNING RATES.

Case Study	Learning Rate	RMSE	MAE	Val_Loss
4	0.01	0.29	0.24	0.01
	0.001	0.30	0.24	0.03
	0.0001	0.57	0.50	0.04

5) Ablation Study 5 (Changing Activation Functions): We use activation function of ReLU initially. It gives RMSE of 0.294 and MAE of 0.246. After replacing the activation function with softmax gives RMSE of 0.290 and MAE of 0.242. We also replace the activation function with tanh. The results are shown in Table VII.

TABLE VII
ABLATION STUDY BY CHANGING ACTIVATION FUNCTION.

Case Study	Activation Function	RMSE	MAE	Val_Loss
5	reLU	0.294	0.246	0.01
	softmax	0.290	0.242	0.01
	tanh	0.292	0.245	0.01

6) Ablation Study 6 (Changing Dropout): We also use dropout of 0.2 which gives RMSE and MAE scores of 0.290 and 0.242, respectively. We apply dropouts of 0.5 and 0.7 as well. The results are shown in Table VIII.

TABLE VIII
ABLATION STUDY BY CHANGING THE DROPOUTS.

Case Study	Dropout	RMSE	MAE	Val_Loss
6	0.2	0.290	0.242	0.01
	0.5	0.291	0.243	0.02
	0.7	0.291	0.243	0.01

D. Performance Analysis of the Best Model

After analyzing all the case studies, we set a model with lower error rate when the optimal batch size, learning rate, optimizer and number of hidden layers are used. Table IX shows the final configuration of our stacked LSTM model for predicting vacant car parking slots in different regions.

TABLE IX
CONFIGURATION OF PROPOSED ARCHITECTURE AFTER ABLATION STUDY.

Configuration	Value
Data set size	408×6
Epochs	100
Optimization function	Adam
Learning rate	0.01
Batch size	16
Activation function	Softmax
Dropout	0.2

IV. COMPARISON WITH BASELINE MODELS

We also apply Recurrent Neural Network (RNN) [20], Autoregressive integrated moving average (ARIMA) [2] and LSTM [23] models to predict vacant car parking time as baseline models. However, in this case, out stacked LSTM model outperforms than that of the mentioned models. Table X shows the performance of different baseline models including our proposed stacked LSTM model.

TABLE X
PERFORMANCE OF DIFFERENT MODELS

Models	RMSE	MAE	Val_Loss
LSTM	0.35	0.35	0.04
RNN	0.45	0.48	0.03
ARIMA	0.40	0.30	0.04
Stacked LSTM	0.29	0.25	0.01

V. RESULTS AND DISCUSSION

We develop an efficient spatio-temporal based machine learning model for predicting vacant parking spaces. First, we apply a clustering method based on their location to group up the regions with similar parking trends. Later, we apply stacked LSTM to predict departure and arrival time of vacant parking spaces.

During data collection, we observe that different regions follow different parking trends. We see that the departure and arrival times of every area maintain a particular pattern. Thus, we apply stacked LSTM in each cluster that predicts departure and arrival times. The basic LSTM model consists of a single hidden LSTM layer followed by a conventional feedforward output layer. The Stacked LSTM is a model extension that contains multiple hidden LSTM layers, each of which contains multiple memory cells. Since our desired output predicts both the departure and arrival time, therefore we incorporate the stacked LSTM in our problem. Our stacked LSTM model consists of several LSTM layers. In stacked LSTM architecture, the output of the first LSTM layer goes as input of the next LSTM layer. In our model the first LSTM has 128 layers and the second LSTM has 64 layers. We use a dropout of 0.2 followed by a dense layer. Our ablation study finds the best configuration for our model. For the hidden layer, we use 128 hidden layers for the first LSTM and 64 for the second LSTM. We apply batch sizes of 64, 32 and 16. Batch size of 16 gives us the lower RMSE and MAE. We changed our activation function with reLU, softmax and tanh. Softmax performs better than the other activation functions. Initially, we apply adam optimizer then we also apply sgd and nadam but adam gives the optimal result. We stick with the initial learning rate which is 0.01 because after changing the learning rate with 0.001 and 0.0001, we find an increased RMSE and MAE scores. Lastly, we find changing the dropout values with 0.2 which gives the optimal result.

We consider applying RNN in our model, but we work with only 408 buildings for now but in reality there are thousands of buildings. In that case, RNN does not give optimal results because it has a long term dependency problem due to the vanishing gradient problem. LSTM overcomes the long term dependency problem. After training the model, the prediction for Dhanmondi, Gulshan and Mirpur clusters perform better than the rest of the clusters because the number of instances in Gulshan, Dhanmondi and Mirpur is large compared to other clusters. A few studies [1], [9], [10] show that weighted LSTM and HMM models also provide satisfactory results in predicting future events. However, our model has several limitations which can be an avenue for future research. Our model cannot find the nearest parking spaces and needs the longitude and latitude values of a specific building to predict vacant parking space.

The parking behavior of people might sound unrealistic as if they are robotic entities. However, we observe similar patterns of the parking behavior because the office time is generally 9:00 AM to 5:00 PM and the time of the educational institutes is mostly 8:00 AM to 4:00 PM. In addition to this, we mainly work with departure and arrival time. For this reason, we do not consider any spatial and managerial factors. Therefore, we do not need to apply any nearest neighbor or distance searching algorithms by using spatial data structures.

VI. Conclusion

In this paper, we have worked with real world spatio-temporal dataset to predict free parking time (departure time and arrival time). First, we have applied clustering method where k=7 in seven different locations (longitude and latitude) to group up the regions with similar parking trend. Later, we have applied stacked LSTM on each cluster to predict departure and arrival time of those parking spaces. We have trained with stacked LSTM to predict vacant parking spaces in a region. We have conducted an ablation study to see the impact of the components within the architecture and to select the optimal values of the components. We have also compared our techniques with the baseline techniques and found that our system outperforms than that of the previous approaches.

References

[1] Al Rafi, A.S., Rahman, T., Al Abir, A.R., Rajib, T.A., Islam, M., Mukta, M.S.H.: A new classification technique: random weighted lstm (rwl). In: 2020 IEEE Region 10 Symposium (TENSYMP). pp. 262–265. IEEE (2020)

[2] Benvenuto, D., Giovanetti, M., Vassallo, L., Angeletti, S., Ciccozzi, M.: Application of the arima model on the covid-2019 epidemic dataset. Data in brief **29**, 105340 (2020)

[3] BRTA: Number of registered motor vehicles in bangladesh (yearwise) Retrieved from. `shorturl.at/kqD78` (July 7, 2020)

[4] Gandhi, R., Nagarajan, S., Chandramohan, J., Parimala, A., Arulmurugan, V.: Iot based automatic smart parking system with ev-charging point in crowd sensing area. Annals of the Romanian Society for Cell Biology **25**(6), 6398–6409 (2021)

[5] Kotb, A.O., Shen, Y.C., Zhu, X., Huang, Y.: iparker—a new smart car-parking system based on dynamic resource allocation and pricing. IEEE transactions on intelligent transportation systems **17**(9), 2637–2647 (2016)

[6] Leon, M.I., Iqbal, M.I., Meem, S., Alahi, F., Ahmed, M., Shatabda, S., Mukta, M.S.H.: Dengue outbreak prediction from weather aware data. In: International Conference on Bangabandhu and Digital Bangladesh. pp. 1–11. Springer (2022)

[7] Liang, J.K., Eccarius, T., Lu, C.C.: Investigating factors that affect the intention to use shared parking: A case study of taipei city. Transportation Research Part A: Policy and Practice **130**, 799–812 (2019)

[8] Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: Proceedings. vol. 89, pp. 89–94 (2015)

[9] Mukta, M.S.H., Ali, M.E., Mahmud, J.: Identifying and predicting temporal change of basic human values from social network usage. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. pp. 619–620 (2017)

[10] Mukta, M.S.H., Ali, M.E., Mahmud, J.: Temporal modeling of basic human values from social network usage. Journal of the Association for Information Science and Technology **70**(2), 151–163 (2019)

[11] Nawshin, S., Mukta, M.S.H., Ali, M.E., Islam, A.N.: Modeling weather-aware prediction of user activities and future visits. IEEE Access **8**, 105127–105138 (2020)

[12] Rahman, M.M., Majumder, M.T.H., Mukta, M.S.H., Ali, M.E., Mahmud, J.: Can we predict eat-out preference of a person from tweets? In: Proceedings of the 8th ACM Conference on Web Science. pp. 350–351 (2016)

[13] Raju, V.G., Lakshmi, K.P., Jain, V.M., Kalidindi, A., Padma, V.: Study the influence of normalization/transformation process on the accuracy of supervised classification. In: ICSSIT. pp. 729–735. IEEE (2020)

[14] Riad, S., Ahmed, M., Himel, M.H., Mim, A.H., Zaman, A., Islam, S., Mukta, M.S.H.: Prediction of soil nutrients using hyperspectral satellite imaging. In: Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021. pp. 183–198. Springer (2022)

[15] Rupai, A.A.A., Mukta, M.S.H., Islam, A.N.: Predicting bowling performance in cricket from publicly available data. In: Proceedings of the International Conference on Computing Advancements. pp. 1–6 (2020)

[16] Sagheer, A., Kotb, M.: Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. Scientific reports **9**(1), 1–16 (2019)

[17] Shao, Z., Taniar, D.: Range-based nearest neighbour search in a mobile environment. In: Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia. pp. 215–224 (2014)

[18] Sharmeen, N., Houston, D.: Urban form, socio-demographics, attitude and activity spaces: Using household-based travel diary approach to understand travel and activity space behaviors. Urban Science **4**(4), 69 (2020)

[19] Tato, A., Nkambou, R.: Improving adam optimizer (2018)

[20] Tomasiello, S., Loia, V., Khaliq, A.: A granular recurrent neural network for multiple time series prediction. Neural Computing and Applications **33**(16), 10293–10310 (2021)

[21] Zacepins, A., Komasilovs, V., Kviesis, A.: Implementation of smart parking solution by image analysis. In: VEHITS. pp. 666–669 (2018)

[22] Zhang, Z.: Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). pp. 1–2. Ieee (2018)

[23] Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J.: Lstm network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems **11**(2), 68–75 (2017)