Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 4, pp. 1191–1201, DOI 10.12694/scpe.v24i4.2550

SMART PARKING SYSTEM: OPTIMIZED ENSEMBLE DEEP LEARNING MODEL WITH INTERNET OF THINGS FOR SMART CITIES

SUDHA PRATHYUSHA JAKKALADIKI[*], PETRA POULOVÁ[†], PAVEL PRAŽÁK[‡], AND BARBORA TESAŘOVÁ[§]

Abstract. In the recent era of smart city ecosystems and the Internet of Things (IoT), innovative, intelligent parking systems must make cities more sustainable. Every year, the increasing number of city vehicles requires more time to search for parking slots. In large cities, 10% of the traffic congestion occurs because of cruising; drivers spend almost 20 minutes searching for free space to park their vehicles. The passing time of waiting for parking in the traffic leads the issues such as energy, pollution, and stress. There needs to be more than the developed solutions. Therefore, the necessary to create a parking slot availability detection system that informs the drivers in advance about the free parking slot based on location. This paper introduces an enhanced ensemble Deep Learning (DL) model designed to forecast parking slot availability through the integration of IoT, cloud technology, and sensor networks. The devised model, known as Ensemble CNN-Boosted Graph LSTM (ECNN-BGLSTM), is optimized using a Genetic Algorithm (GA) framework. The model's performance is rigorously evaluated using a dataset from Europe, and various metrics, including Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE), are employed for assessment. The experimental findings demonstrate the superior performance of the proposed model compared to existing state-of-the-art approaches.

Key words: Smart parking, Internet of Things (IoT), Optimization, Deep learning (DL), Convolution neural network (CNN), Graph Long short-term memory (LSTM), Genetic algorithm (GA).

1. Introduction. The population of the world is migrating from rural to urban places, which increases the population density of large cities. Based on the United Nations Population Division, the world's population is expected to move to cities by 2050. In practice, the global urban scenario requires advanced technologies to meet the demands of the smart city. The advancement in sensor networks and sensor technologies leads new models to build, deploy and promote sustainable development systems to escalate the challenges in urbanization [12, 1]. Urban mobility sustainability and traffic congestion reduction are critical challenges in urbanization where parking space is limited [22].

Cities are getting smarter with the increased utilization of the Internet of Things (IoT), and the applications have risen rapidly in various areas of cities, including transportation systems, hospitals, shopping malls, airports, etc. In IoT-based smart cities, the smart parking system is the most significant topic due to the increasing number of vehicles. In recent years, parking issues, while new vehicles join during traffic, have attracted many researchers [19]. With growing technologies, the IoT concept and deep learning (DL) are utilized in smart city planning that gradually tackles mobility issues and provides citizens with sustainable infrastructure for ecological, economic, and social scenarios [3]. Recently, most intelligent systems have been in the form of mobile-based solutions which helps drivers report traffic jams, accidents, road conditions, and alternative routes. However, parking is still challenging because of the more significant number of vehicles moving across the roads.

The primary motivations behind the research are:

1. With increasing urbanization and population growth, cities face challenges related to traffic congestion

[*] Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, Hradec Králové, Czech Republic (sudha.jakkaladiki@uhk.cz)

[†] Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, Hradec Králové, Czech Republic

[‡] Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, Hradec Králové, Czech Republic

[§] Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, Hradec Králové, Czech Republic

and parking shortages. Intelligent parking solutions help address these issues by making better use of existing parking spaces.

2. Many cities strive to reduce their carbon footprint and improve air quality. Smart parking can contribute to this goal by reducing traffic congestion and lowering vehicle emissions.
3. Smart parking aligns with the broader goal of enhancing urban mobility and creating a connected transportation network. It encourages the use of public transportation and other eco-friendly commuting options.

Researchers have developed different approaches using various types of collected data in the literature to solve the issues. Most researchers used machine learning (ML) and time series models to compute parking places' occupancy in the existing literature. Due to the increasing number of sensor data, the conventional decision support systems are not providing efficient performance like deep neural networks since the deep learning (DL) approaches help estimate linear and nonlinear samples. Comparatively, deep learning (DL) methods help predict occupancy, specifically feed-forward networks. Yet, simple deep neural networks still need to be integrated with temporal domain data to predict parking solutions [16]. Hanzl [10] conducted a study to direct drivers to find parking spaces in urban areas. Static and dynamic routing approaches are analyzed with their advantages and disadvantages. Canli and Toklu [7] developed LSTM based model to predict parking space availability. The mobile application-based dynamic access is provided with the LSTM model and the occupancy rates of the parks in the desired locations are displayed in the applications using the relevant parameters. Intending to develop a smart parking system using the DL model, this paper contributes the following:

1. Developed a smart car parking space availability prediction system using IoT and ensemble DL approaches.
2. Using the considered European market data, the available car parking space is predicted with the features extracted by the Ensemble CNN and Boosted Graph LSTM model. A Genetic algorithm optimizes the ECNN-BGLSTM to improve the model performance.
3. The model is evaluated based on the metrics such as Root mean square error (RMSE), Mean square error (MSE), and Mean absolute error (MAE). The experimental analysis results show that the proposed model outperforms the state-of-the-art approaches.

The remaining section of this paper is as follows: section 2 discusses the related works of literature on the smart parking system. Section 3 introduced the ensemble DL model for predicting the available parking space. Section 4 discusses the experimented results and compares them with state-of-the-art approaches. Section 5 concludes the proposed model with its merits and future directions.

2. Related work. This section discusses the literature related to predicting smart car parking systems. Siddiqui et al., [20] accomplished the empty car parking services through sensors, and artificial neural networks (ANN) have been used for the prediction. Aizenberg et al., utilized the DL-based approaches initially based on ANN. The DL approach is differed from ANN by adding more hidden layers where the successive layers are taken from the output of the previous layers based on learning and data representation [6, 5]. Ghulam Ali et al., [1] proposed deep, long-term memory (DLSTM) by integrating IoT, cloud, and sensor technologies. Using the Birmingham parking dataset, three kinds of experiments are conducted to predict the availability of free parking space based on location, days of the week, and hours of the day. The result analysis shows that the DLSTM model outperforms other state-of-the-art approaches.

Liu et al., [14] developed a combination of Graph CNN and Recurrent neural network (RNN) to predict the real-time parking slot occupancy. Graph neural network has been utilized to extract the spatial data of the traffic flow, and RNN has been used to extract the temporal data of the traffic flow. The demonstration shows that the integration model performs better than other approaches. Carli et al., [8] proposed LSTM with a gated Recurrent unit (GRU) to predict the free parking slot availability, and the demonstration proves that GRU performs better than the LSTM model. Bock et al., [4] investigated the taxi fleets applicability to detect the on-street parking availability. The detection of free parking spots is analyzed using the taxi transit frequencies and the availability of parking spaces from vehicles equipped with GPS sensors. Tekouabou et al., [21] developed a combination of IoT and ensemble approaches to predict the free parking availability in the smart city. Using the Birmingham parking dataset, the obtained prediction accuracy was 94% using the bagging ensemble model. Kothai et al., [13] proposed a hybrid DL model using Boosted LSTM and CNN to predict the dynamic behavior of vehicle congestion in the road. From the traffic images, the CNN extracts the features, and BLSTM strengthens the weak classifiers. The real traffic scenario is simulated using OMNeT++ and SUMO with TensorFlow Python libraries. The experimental results show that the proposed model secured an accuracy of 98%.

Hébert et al., [11] created a high-resolution prediction model to prognosticate the accident circumstances on the roads by intersecting via big data analytics. Big data analysis is the approach that permits researchers to extract significant data from a larger amount of heterogeneous data. The authors employed balanced RF to sample the imbalanced data with various ML approaches such as decision trees, ANN, and Bayesian networks. Exploiting the features, including the attributes of weather, arterial segment, date, and time, road accidents are successfully prognosticated. Shengdong et al., [9] proposed a hybrid Multi-Model DL approach to forecasting the traffic flow by incorporating GRU and 1DNN to attain the features. The features of spatial local features have long dependencies, and correlations are processed by the end-to-end multimodel. The CNN-GRU model solves the issues with the long temporal dependencies of spatial-temporal correlations.

Moses and Parvathi [15] employed support vector regression to map the input using nonlinear mapping with M-dimensional features. The MSE estimates the performance by scaling the average error squares. The linear regression approach erects the scalar and independent response, and the DT approach reckons the information gain. Sellami and Alaya [18] developed Self-adaptive Multi kernel clustering for VANET to determine the load balance and other attainable resources. This model depicts the ambivalent density of deceleration, conveyance nodes, and bounded radio ranges using three stages such as cluster initialization, cluster adaption, and fusion stage. Compared to other approaches, this model poses the resultant to predict the traffic in the urban area, and the algorithm's complexity is optimized with the optimization algorithm. Ranjan et al., [17] integrate the CNN, CNN-transpose, and LSTM to redirect the congestion level. The encoder of the convolution process encodes the input spatial extraction features into a low-resolution latent state. The recurrent network with LSTM ascertains the time series data. The PredNet and ConvLSTM attain improved accuracy in predicting traffic congestion by associating the spatial and temporal features.

The literature mentioned above primarily discusses the proposed models in various ML and DL models. A research gap exists in validating these models using real-world traffic data from urban areas. Real-world validation will help assess the models' performance under diverse traffic conditions and validate their practical applicability. Addressing these research gaps would contribute to advancing traffic prediction and resource management in smart cities, leading to more efficient and reliable urban mobility solutions.

3. Proposed method. The developed framework is based on the API for finding parking, an Optimized DL model, a cloud-based parking server, and an open data portal. The API receives the parking request from the applications and returns the parking list, which is all suitable. The car parks within 2 km of the destination are listed as a choice for a parking slot. The direct request is made to the cloud-based smart parking server for each parking slot and analyzed through the developed DL approach. The cloud-based server receives the detected available parking slot in real-time and if the request is for a future date, then it is directed to the DL-based service. The ECNN-BGLSTM detects the parking slots in that service and returns the result. The data collection from the open data portal is described in the data preparation stage. These details are illustrated in the proposed model in Fig 3.1.

3.1. Smart car parking system (SCPS). The SCPS system informs the drivers regarding the parking slot availability of various parking locations. The checking of parking slot availability is a time-dependent problem. This paper developed a real-time response through a cloud server and optimized a DL-based approach to predict the available parking slot accurately. The developed SCPS highly depends on the optimized hybrid DL model, which provides accurate data on the availability of parking slots to the drivers. Based on the query requested by the driver, the SCPS analyzed the results and fetched the available parking slot response based on the request's specific time slot or a specific day. These fetched data are processed by the DL model and predict the free parking availability and transfer the data to the cloud car parking server to inform the driver regarding the availability.

3.2. Decision support system using optimized ensemble DL model. The proposed DSS for parking slot availability prediction receives input from sensors and predicts the availability of various parking locations

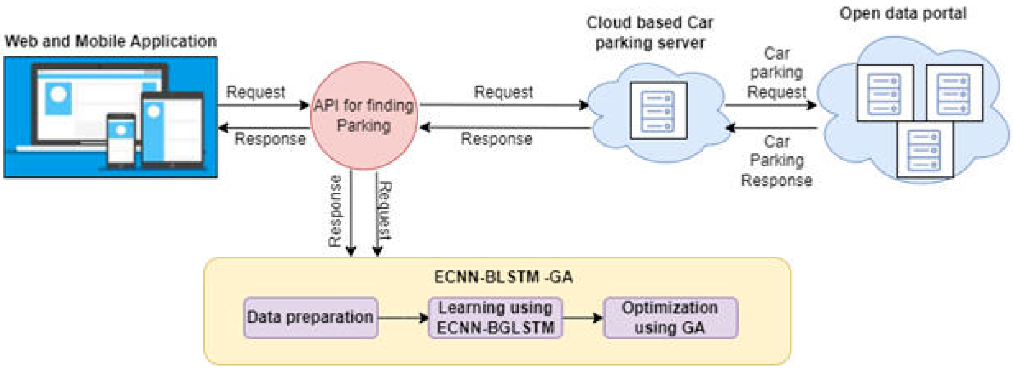


Fig. 3.1: Overview of the proposed smart parking detection system

at a specific time. Once the data of the request is obtained, applied ECNN-BLSTM model for prediction. Initially, CNN was employed to extract the features from the input car traffic images and has been trained by the BGLSTM model.

3.2.1. CNN. CNN has the powered ability to represent the features of the input image through neuron local connectivity and weight sharing. The layers include the convolution layer to learn the feature representation and the pooling layer to accomplish the invariance. In the convolution layer, the inputs are received by the neuron from the previous layer. The different features are learned by the kernels to convolute from the previous layer. The convolution process is denoted in Eqn 3.1

$$Y_F^l = \sigma\left(\sum_{i=1}^{F_{l-1}} Y_K^{l-1}, w_{KF}^l + b_F^l\right), \; F \in [1, F_l] \tag{3.1}$$

where F is the filter, l is the layer, Fth activation map of lth layer is denoted as Y_F^l, Kth activation map of (l-1)th layer is denoted as Y_K^{l-1}, W and b connects the fth activation map of lth layer at Kth position and bias respectively. The filters of layer ‘l’ are denoted as $F_l$and σ is the nonlinear activation function [17].

The pooling operations subdued the activation map spatial size as denoted in Eqn 3.2

$$Y_F^l\left(i,j\right) = \sigma\left(\sum_{a=1}^{f_{l-1}}\sum_{b=0}^{m-1}\sum_{c=0}^{n-1}\left(w_{KF}^l\left(b,c\right)\bigotimes Y_K^{l-1}\left(i+b,j+c\right)+c_F^l\right)\right) F \in [1, F_l] \tag{3.2}$$

where m, n is the size of filter in the convolution process, $Y_F^l\left(i,j\right)$ obtained from the output of the previous layer, b, and c are the kernel function. The convolution layer is entailed by the location of the l+1th pooling layers fth activation function by learning the output of the previous layer with the filter size (2,2). The update function is computed as in Eqn 3.3

$$Y_F^{l+1}\left(i,j\right) = \sigma\left(\sum_{a=1}^{f_{l-1}}\sum_{b=0}^{m-1}\sum_{c=0}^{n-1}\left(w_{KF}^{l+1}\left(b,c\right)\bigotimes Y_K^{l-1}\left(2i+b,j+c\right)+c_F^{l+1}\right)\right) F \in [1, F_{l+1}] \tag{3.3}$$

3.2.2. GLSTM. The traditional LSTM is based on the backpropagation model which consists of three gates including input, forget, and output gate. The graph LSTM is differentiated from the standard LSTM in terms of the graph structure where each tree node declares LSTM unit. The forward direction of the node captured the history and the backward direction characterizes the response. This mode has five layers including one input, three hidden, and one output layers. The input layer consists of a recurrent neural network (RNN) which denotes the extracted features from CNN. Each feature of the input layer is declared by the neuron.

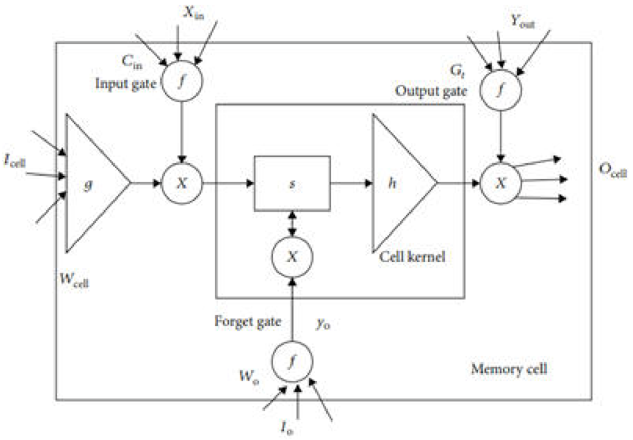


Fig. 3.2: GLSTM Cell structure [2]

The usage of a batch normalization layer reduces the overfitting issue which standardizes the prediction of the hidden layer. For the iteration t, the Graph LSTM computed the forward hidden sequence $\overrightarrow{h}$ and Backward hidden sequence $\overleftarrow{h}$ and the output sequence Y as in Eqn 3.4 and 3.5

$$\overrightarrow{h} = H(w_{F\overrightarrow{h}}\mathcal{F}_t + w_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1}\mathcal{F}_t + b_{\overrightarrow{h}}) \tag{3.4}$$

$$\overleftarrow{h} = H(w_{F\overleftarrow{h}}\mathcal{F}_t + w_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1}\mathcal{F}_t + b_{\overleftarrow{h}}) \tag{3.5}$$

$$Y_t = w_{\overrightarrow{h}Y}\overrightarrow{h}_t + w_{\overleftarrow{h}}\overleftarrow{h}_t + b_Y \tag{3.6}$$

where H is the hidden layer function, w is the weight matrices and b is the bias vector of each feature F. For each feature, the GLSTM cell is denoted in Fig 3.2.

For the particular iteration t, the input ‘I’, forget ‘$\mathcal{F}$’, output ‘O’ gate and cell state ‘$\mathbb{C}$’ with the activation function σ is updated as in Eqn 3.7 to 3.11

$$\mathcal{F}_t = \sigma(w_{F\mathcal{F}}F_t + w_{\overrightarrow{h}F}\overrightarrow{h}_{t-1} + w_{C\mathcal{F}}\mathbb{C}_{t-1} + b_{\mathcal{F}}) \tag{3.7}$$

$$I_t = \sigma(w_{F_I}F_t + w_{\overrightarrow{h}i}\overrightarrow{h}_{t-1} + w_{C_I}\mathbb{C}_{t-1} + b_I) \tag{3.8}$$

$$O_t = \sigma(w_{F_O}F_t + w_{\overrightarrow{h}o}\overrightarrow{h}_{t-1} + w_{C_O}\mathbb{C}_{t-1} + b_O) \tag{3.9}$$

$$\mathbb{C}_t = \mathcal{F}_t\mathbb{C}_{t-1} + I_t tanh(w_{F_\mathbb{C}}F_t + w_{\overrightarrow{h}\mathbb{C}}\overrightarrow{h}_{t-1} + b_\mathbb{C}) \tag{3.10}$$

$$\overrightarrow{h}_t = O_t tanh\mathbb{C}_t \tag{3.11}$$

For backward G-LSTM, the parameters are updated as follows in Equation from 3.12 to 3.18.

$$\mathcal{F}_t = \sigma(w_{F\mathcal{F}}F_t + w_{\overleftarrow{h}F}\overleftarrow{h}_{t-1} + w_{cF}\mathbb{C}_{t-1} + b_F) \tag{3.12}$$

$$I_t = \sigma(w_{f_I} f_t + w_{\overleftarrow{h}_I} \overleftarrow{h}_{t-1} + w_{c_I}\mathbb{C}_{t-1} + b_I) \tag{3.13}$$

$$O_t = \sigma(w_{f_O} f_t + w_{\overleftarrow{h}_O} \overleftarrow{h}_{t-1} + w_{c_O}\mathbb{C}_{t-1} + b_O) \tag{3.14}$$

$$\mathbb{C}_t = F_t\mathbb{C}_{t-1} + I_t tanh(w_{f_\mathbb{C}} f_t + w_{\overleftarrow{h}_C} \overleftarrow{h}_{t-1} + b_\mathbb{C}) \tag{3.15}$$

$$\overleftarrow{h}_t = O_t tanh\mathbb{C}_t \tag{3.16}$$

3.2.3. Ensemble CNN-BGLSTM. The ensemble model aggregates the BGLSTM and CNN to overcome the overfitting issue of the prediction system. The GLSTM model trained with the G_k^i over the training data of iteration k. Initially, G(i) is equally set as $G_1\,(i) = 1/n$. The modified output cell of Eqn 3.9 and Eqn 3.14 is declared as, (Forward) $O_k = \sum_{k=0}^{n}\left(\sigma(w_{F_k}F_k + w_{\overrightarrow{h}_k}\overrightarrow{h}_{k-1} + w_{C_k}\mathbb{C}_{k-1} + b_k)\right)$ 3.9

(Backward)$O_k = \sum_{k=0}^{n}\left(\sigma(w_{f_k}f_k + w_{\overleftarrow{h}_k}\overleftarrow{h}_{k-1} + w_{c_k}\mathbb{C}_{k-1} + b_k)\right)$ 3.14

The error function defined by the user is identified as the boosting outputs expressed mathematically as,

$$E_k = (O_{acutal} - O_k) \tag{3.17}$$

For the expression, the network parameter a_k is computed as,

$$a_k = 0.5\left\{ln\frac{(1-E_k)}{E_k}\right\} \tag{3.18}$$

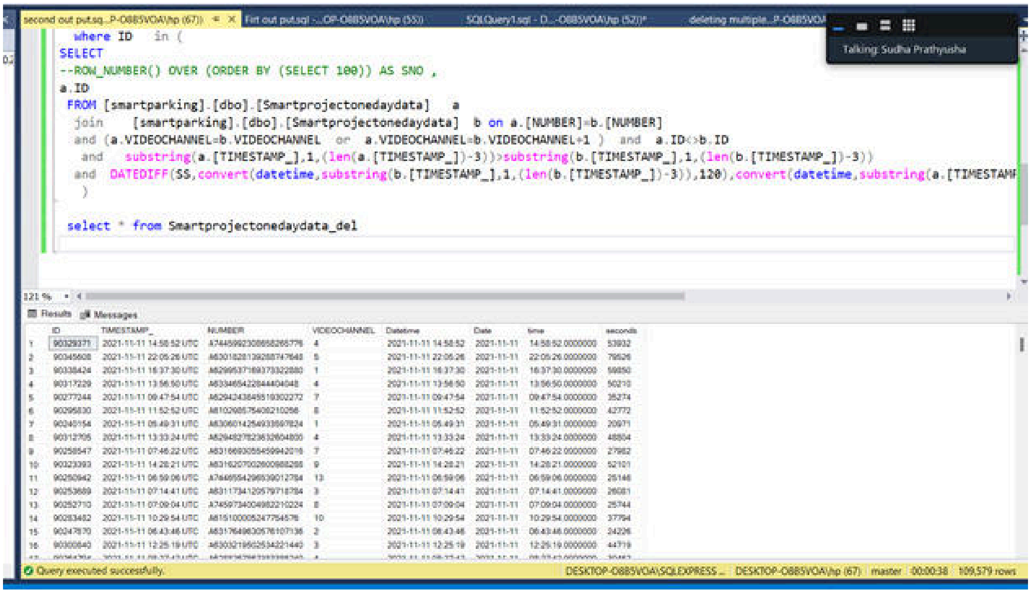
E_k is the final boosted ensemble output which is computed while the error is declared as zero and the mathematical expression is declared as in Eqnn 3.19

$$Y_k = sign\left(\sum a_k.O_k\right)/a_k \tag{3.19}$$

3.2.4. Optimization using GA. The genetic algorithm (GA) is a global optimization approach that mimics biological evolution in nature. It obtains the optimal solution through continuous evolution, based on the survival of the fittest. GA involves three operations: selection, crossover, and mutation. Crossover is the primary method for generating offspring and forming a new generation.

The GA process includes an initial population, parameter coding, designing a fitness function, genetic operations, and control parameter settings. The main ideas behind GA are as follows: (i) Generating the initial population randomly and conducting classification. The next population is generated through GA selection, crossover, and variations, serving as the first generation. (ii) Merging the parental population with the offspring population, generating the second generation, and conducting non-inferior classification. Selecting the appropriate individuals to create the new parental population based on each individual's focusing distance. (iii) Continuing the generation of offspring until the condition is satisfied.

GA-based optimization has several advantages over other approaches. It operates directly on decision variable coding and structural objects like sets, matrices, sequences, graphs, and trees. GA is not only a simulation of biological gene processes and genetic evolution but also facilitates the application of genetic operators. It has been successfully applied in various fields such as automatic function, number optimization, production scheduling, machine learning, and image processing. GA utilizes objective function values for searchability and fitness function to measure individual fitness, providing an advantage when dealing with difficult-to-derive objective functions. Another advantage of GA is its global search ability, starting from an initial population with diverse individuals to avoid getting stuck at local optima. GA follows probability rules rather than deterministic rules, offering flexible searching and reducing parameter influences. Its strong scalability allows it to be combined with various technologies.


second out put.sq...P-O8B5VOA\hp (67)) ⇐ ✕
Firt out put.sql -...OP-O8B5VOA\hp (55))
SQLQuery1.sql - D...-O8B5VOA\hp (52))*
deleting multiple...P-O8B5VOA
Talking: Sudha Prathyusha
121 %
Results
Messages

```
  where ID   in (
SELECT
--ROW_NUMBER() OVER (ORDER BY (SELECT 100)) AS SNO ,
a.ID
 FROM [smartparking].[dbo].[Smartprojectonedaydata]   a
  join    [smartparking].[dbo].[Smartprojectonedaydata]  b on a.[NUMBER]=b.[NUMBER]
  and (a.VIDEOCHANNEL=b.VIDEOCHANNEL  or  a.VIDEOCHANNEL=b.VIDEOCHANNEL+1 )  and  a.ID<>b.ID
   and   substring(a.[TIMESTAMP_],1,(len(a.[TIMESTAMP_])-3))>substring(b.[TIMESTAMP_],1,(len(b.[TIMESTAMP_])-3))
  and  DATEDIFF(SS,convert(datetime,substring(b.[TIMESTAMP_],1,(len(b.[TIMESTAMP_])-3)),120),convert(datetime,substring(a.[TIMESTAMP
   )

 select * from Smartprojectonedaydata_del
```

	ID	TIMESTAMP_	NUMBER	VIDEOCHANNEL	Datetime	Date	time	seconds
1	90329371	2021-11-11 14:58:52 UTC	A7445992308658265776	4	2021-11-11 14:58:52	2021-11-11	14:58:52.0000000	53932
2	90345608	2021-11-11 22:05:26 UTC	A6301828139268747648	5	2021-11-11 22:05:26	2021-11-11	22:05:26.0000000	79526
3	90338424	2021-11-11 16:37:30 UTC	A6299537189373322880	1	2021-11-11 16:37:30	2021-11-11	16:37:30.0000000	59850
4	90317229	2021-11-11 13:56:50 UTC	A633465422844404048	4	2021-11-11 13:56:50	2021-11-11	13:56:50.0000000	50210
5	90277244	2021-11-11 09:47:54 UTC	A6294243845519302272	7	2021-11-11 09:47:54	2021-11-11	09:47:54.0000000	35274
6	90295830	2021-11-11 11:52:52 UTC	A610298575406210256	6	2021-11-11 11:52:52	2021-11-11	11:52:52.0000000	42772
7	90240154	2021-11-11 05:49:31 UTC	A6306014254933597824	1	2021-11-11 05:49:31	2021-11-11	05:49:31.0000000	20971
8	90312705	2021-11-11 13:33:24 UTC	A6294827823632604800	4	2021-11-11 13:33:24	2021-11-11	13:33:24.0000000	48804
9	90258547	2021-11-11 07:46:22 UTC	A6316693055459942016	7	2021-11-11 07:46:22	2021-11-11	07:46:22.0000000	27982
10	90323393	2021-11-11 14:28:21 UTC	A6316207002600988288	9	2021-11-11 14:28:21	2021-11-11	14:28:21.0000000	52101
11	90250942	2021-11-11 06:59:06 UTC	A7446554296539012784	13	2021-11-11 06:59:06	2021-11-11	06:59:06.0000000	25146
12	90253689	2021-11-11 07:14:41 UTC	A6311734120579718784	3	2021-11-11 07:14:41	2021-11-11	07:14:41.0000000	26081
13	90252710	2021-11-11 07:09:04 UTC	A7459734004982210224	6	2021-11-11 07:09:04	2021-11-11	07:09:04.0000000	25744
14	90283482	2021-11-11 10:29:54 UTC	A615100005247754576	10	2021-11-11 10:29:54	2021-11-11	10:29:54.0000000	37794
15	90247870	2021-11-11 06:43:46 UTC	A6317649830576107136	2	2021-11-11 06:43:46	2021-11-11	06:43:46.0000000	24226
16	90300840	2021-11-11 12:25:19 UTC	A6303219502534221440	3	2021-11-11 12:25:19	2021-11-11	12:25:19.0000000	44719
17	[illegible]	[illegible]	[illegible]	4	[illegible]	[illegible]	[illegible]	[illegible]

Query executed successfully.

DESKTOP-O8B5VOA\SQLEXPRESS ... DESKTOP-O8B5VOA\hp (67) master 00:00:38 109,579 rows

Fig. 4.1: Sample data

4. Results and Discussions. This section presents the experimental results and discussions using Keras, a neural network library. The Jupiter notebook was chosen for compiling the program. Geographic data from the European market was collected for processing. Initially, the dataset consisted of eight parameters: ID, timestamp, number, video_channel, date_time, date, time, and seconds, with 109,736 records. However, the fixed parameter values for each car park did not significantly affect the model's performance. After recording the data, abnormal entries were identified and removed through preprocessing to eliminate duplicate entries, missing values, and Langrange interpolation. The data was then standardized, resulting in 109,579 records. Fig 4.1 illustrates a sample of the processed data.

4.1. Metrics used. The predictive model performance is evaluated using the metrics such as Root Mean Square Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), and Accuracy values.

1. RMSE has been used to find the prediction model deviation which takes the values from 0 to $+\infty$. The closer the value of 0, the lower the prediction model deviation. The mathematical expression is stated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - Y)^2} \tag{4.1}$$

2. MSE: It depicts how the regression curse is closer to the set of points and it is generally a regression problem and it is stated as,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Y)^2 \tag{4.2}$$

3. MAE: Average error magnitude in the predictions and the low value indicates the accurate prediction.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - Y| \tag{4.3}$$

Table 4.1: Error assessment

Train-Test set	**Models**	**MAE**	**MSE**	**RMSE**	**MdAE**	**MSLE**
70-30%	SVM	2.5	8.5	2.91	0.032	0.014
	RF	2.43	7.12	2.66	0.033	0.017
	LSTM	0.97	2.53	1.58	0.023	0.008
	Proposed ECNN-BGLSTM	0.88	1.83	1.35	0.018	0.002
80-20%	SVM	2.71	8.62	2.93	0.035	0.015
	RF	2.53	7.81	2.79	0.036	0.019
	LSTM	1.02	3.12	1.77	0.034	0.010
	Proposed ECNN-BGLSTM	0.91	2.03	1.42	0.021	0.004
90-10%	SVM	2.83	9.12	3.01	0.057	0.017
	RF	2.87	8.37	2.89	0.043	0.021
	LSTM	1.11	3.18	1.78	0.045	0.011
	Proposed ECNN-BGLSTM	0.98	2.43	1.55	0.035	0.005

4. Median Absolute Error (MdAE):

$$MdAE = median\left(\sum_{i=1}^{N} |Y_i - Y|\right) \tag{4.4}$$

5. Mean Squared Log Error (MSLE): It is declared as,

$$MSLE = \frac{1}{N}\sum_{i=1}^{N} \left(\log\left((Y_{i})\ +1\right) - \log\left((Y)\ +1\right)\right)^2 \tag{4.5}$$

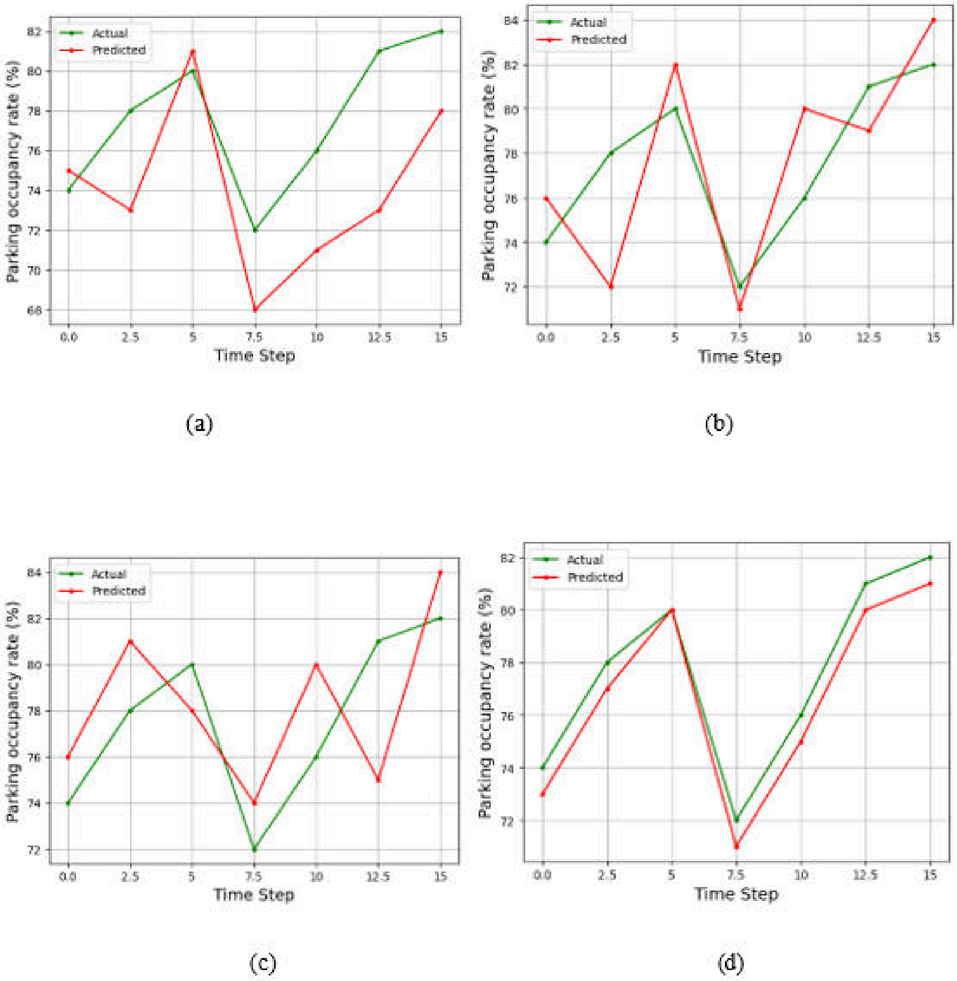
where Y_i is the actual available parking space and Y is the prediction of the available parking space predicted by the ECNN-BGLSTM.

4.2. Result analysis and comparison. The efficiency of the proposed model is compared with standard Deep Learning (DL) approaches, including SVM, Random Forest, and LSTM [7]. The model's quality is evaluated under three different cross-validation scenarios: 70:30, 80:20, and 90:10, where the first term represents the percentage of training data and the second term represents the percentage of testing data. The models were executed for 100 epochs, and their performance is presented in Table 4.1. Among the scenarios, the proposed model achieved an improved prediction rate with a reduced error rate compared to other approaches, specifically in the 70:30% scenario.

Fig 4.2 displays the actual and predicted values of both the proposed and existing approaches. The green line represents the actual prediction results, while the red line denotes the predicted results of the approaches. Specifically, Fig 4.2 (d) illustrates the efficient prediction of the parking occupancy rate using the proposed model, outperforming the other approaches. In contrast, Fig 4.2 (a) for SVM, Fig 4.2 (b) for RF, and Fig 4.2 (c) for LSTM show good predictions in linear time series but worse predictions at maximum and minimum points.

The accuracy comparison in terms of the number of epochs for predicting available parking spaces is depicted in Fig 4.3. As the number of epochs increases, the accuracy also improves. At 100 epochs, the proposed model achieved an enhanced accuracy of 98.5% for predicting parking space availability. In comparison, other approaches, such as SVM, RF, and LSTM, achieved accuracies of 90%, 91%, and 95.3%, respectively. Fig 4.4 shows the training loss comparison. As the number of epochs increases, the loss value decreases. Notably, the proposed ECNN-BGLSTM model outperforms other approaches by reducing the training loss to 0.032, whereas state-of-the-art approaches like SVM (0.09), RF (0.08), and LSTM (0.072) exhibit higher loss values.

5. Conclusion. IoT enables the intelligent interconnection between things, the cloud, and individuals, facilitating efficient action plans. This smart exchange of information is processed and sent to the cloud



(a)

(b)

(c)

(d)

Fig. 4.2: Prediction results (a) SVM (b) Random Forest (c) LSTM (d) Proposed ECNN-BGLSTM

server via the web, where it is utilized for car parking convenience. The challenge of finding available parking spaces in the European market often results in wrong parking, leading to vehicle congestion and traffic jams. Therefore, the development of a smart parking system that notifies users about available spaces is crucial for Europeans. In this paper, we present an intelligent smart car parking space availability prediction system that utilizes ensemble Deep Learning (DL) and IoT. Using data sourced from the European market, the prediction of available parking spaces is carried out by the Ensemble CNN and Boosted Graph LSTM model, utilizing feature extraction. To enhance the model's performance, Genetic Algorithm optimization is employed on the ECNN-BGLSTM. Evaluation of the model encompasses metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). The experimental analysis underscores the superior performance of the proposed model, achieving an enhanced accuracy of 98.5% with notable reductions in error rates and losses. Subsequent research will delve into examining traffic density with diverse parameters, alongside the implementation of novel strategies to efficiently address parking space availability and further extend the smart parking system to encompass smart charging, thus contributing to the realm of smart city applications.

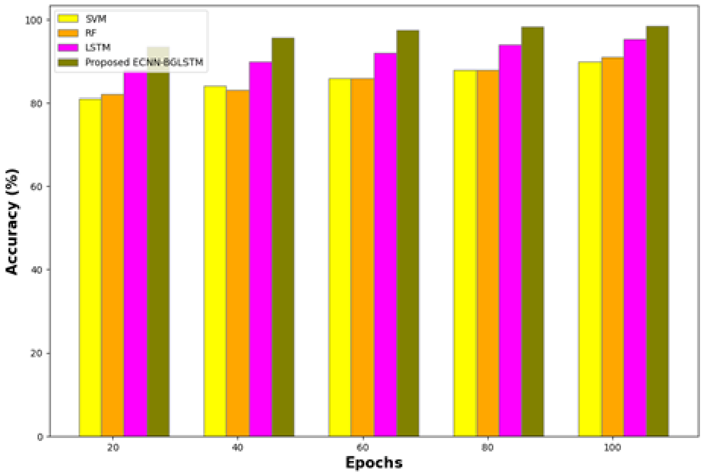


Fig. 4.3: Accuracy Comparison

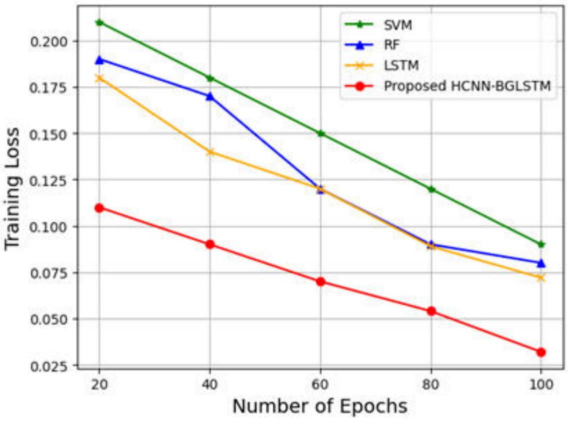


Fig. 4.4: Training Loss Comparison

Acknowledgment. The intention of the article was expressed and proposed within the framework of Smart parking project execution at the University of Hradec Kralove, Czech Republic and the study was supported by the SPEV project, run by the faculty of informatics and management, at the University of Hradec Kralove, Czech Republic.

REFERENCES

[1] G. Ali, T. Ali, M. Irfan, U. Draz, M. Sohail, A. Glowacz, M. Sulowicz, R. Mielnik, Z. B. Faheem, and C. Martis, *Iot based smart parking system using deep long short memory network*, Electronics, 9 (2020), p. 1696.

[2] S. Amudha and M. Murali, *Deep learning based energy efficient novel scheduling algorithms for body-fog-cloud in smart hospital*, Journal of Ambient Intelligence and Humanized Computing, 12 (2021), pp. 7441–7460.

[3] J. Bélissent et al., *Getting clever about smart cities: New opportunities require new business models*, Forrester Research, 3 (2010), pp. 1–31.

[4] F. Bock, S. Di Martino, and A. Origlia, *Smart parking: Using a crowd of taxis to sense on-street parking space availability*, IEEE Transactions on Intelligent Transportation Systems, 21 (2019), pp. 496–508.

[5] O. Brun, Y. Yin, J. Augusto-gonzalez, M. Ramos, O. Brun, Y. Yin, J. Augustogonzalez, M. Ramos, and E. Gelenbe, *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, 2019.

[6] 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, 8 (2020), pp. 183678–183689.

[7] H. Canli and S. Toklu, *Deep learning-based mobile application design for smart parking*, IEEE Access, 9 (2021), pp. 61171–61183.

[8] R. Carli, G. Cavone, S. Ben Othman, and M. Dotoli, *Iot based architecture for model predictive control of hvac systems in smart buildings*, Sensors, 20 (2020), p. 781.

[9] S. Du, T. Li, X. Gong, and S.-J. Horng, *A hybrid method for traffic flow forecasting using multimodal deep learning*, arXiv preprint arXiv:1803.02099, (2018).

[10] J. Hanzl, *Parking information guidance systems and smart technologies application used in urban areas and multi-storey car parks*, Transportation Research Procedia, 44 (2020), pp. 361–368.

[11] A. Hébert, T. Guédon, T. Glatard, and B. Jaumard, *High-resolution road vehicle collision prediction for the city of montreal*, in 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 1804–1813.

[12] A. Hussain, U. Draz, T. Ali, S. Tariq, M. Irfan, A. Glowacz, J. A. Antonino Daviu, S. Yasin, and S. Rahman, *Waste management and prediction of air pollutants using iot and machine learning approach*, Energies, 13 (2020), p. 3930.

[13] G. Kothai, E. Poovammal, G. Dhiman, K. Ramana, A. Sharma, M. A. AlZain, G. S. Gaba, and M. Masud, *A new hybrid deep learning algorithm for prediction of wide traffic congestion in smart cities*, Wireless Communications and Mobile Computing, 2021 (2021), pp. 1–13.

[14] J. Liu, J. Wu, and L. Sun, *Control method of urban intelligent parking guidance system based on internet of things*, Computer Communications, 153 (2020), pp. 279–285.

[15] A. Moses and R. Parvathi, *Vehicular traffic analysis and prediction using machine learning algorithms*, in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), IEEE, 2020, pp. 1–4.

[16] C. Pengzi, Y. Jingshuai, Z. Li, G. Chong, and S. Jian, *Service data analyze for the available parking spaces in different car parks and their forecast problem*, in Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences, 2017, pp. 85–89.

[17] N. Ranjan, S. Bhandari, H. P. Zhao, H. Kim, and P. Khan, *City-wide traffic congestion prediction based on cnn, lstm and transpose cnn*, IEEE Access, 8 (2020), pp. 81606–81620.

[18] L. Sellami and B. Alaya, *Samnet: Self-adaptative multi-kernel clustering algorithm for urban vanets*, Vehicular Communications, 29 (2021), p. 100332.

[19] 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 Generation Computer Systems, 107 (2020), pp. 433–442.

[20] S. Y. Siddiqui, M. A. Khan, S. Abbas, and F. Khan, *Smart occupancy detection for road traffic parking using deep extreme learning machine*, Journal of King Saud University-Computer and Information Sciences, 34 (2022), pp. 727–733.

[21] S. C. K. Tekouabou, W. Cherif, H. Silkan, et al., *Improving parking availability prediction in smart cities with iot and ensemble-based model*, Journal of King Saud University-Computer and Information Sciences, 34 (2022), pp. 687–697.

[22] E. I. Vlahogianni, K. Kepaptsoglou, V. Tsetsos, and M. G. Karlaftis, *A real-time parking prediction system for smart cities*, Journal of Intelligent Transportation Systems, 20 (2016), pp. 192–204.

Edited by: Sathishkumar V E
Special issue on: Scalability and Sustainability in Distributed Sensor Networks
Received: Aug 26, 2023
Accepted: Oct 28, 2023