Received 24 August 2022, accepted 2 September 2022, date of publication 12 September 2022, date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206384

RESEARCH ARTICLE

Travel Time Prediction Using Hybridized Deep Feature Space and Machine Learning Based Heterogeneous Ensemble

JAWAD-UR-REHMAN CHUGHTAI[1,2], IRFAN UL HAQ[1,2], OMAIR SHAFIQ[3], (Member, IEEE), AND MUHAMMAD MUNEEB[4]

[1]Department of Computer and Information Sciences (DCIS), Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad 45650, Pakistan
[2]Digital Disruption Laboratory, Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad 45650, Pakistan
[3]School of Information Technology, Carleton University, Ottawa, ON K1S 5B6, Canada
[4]Department of Mathematics, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Jawad-Ur-Rehman Chughtai (jawadchughtai@gmail.com)

This work was supported by the Khalifa University of Science and Technology.

ABSTRACT Travel Time Prediction (TTP) has become an essential service that people use in daily commutes. With the precise TTP, individuals, logistic companies, and transport authorities can better manage their activities and operations. This paper presents a novel Hybridized Deep Feature Space (HDFS) based TTP ensemble model (HDFS-TTP) for accurate travel time prediction. In the first step, extensive endogenous and exogenous data sources are augmented with traffic data obtained using sensors. Next, we used Principal Component Analysis (PCA) and Deep Stacked Auto-Encoder (DSAE) for feature reduction. We generated feature spaces of deep learning models, namely Convolutional Neural Network (CNN), Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), and fed them to a model based on Support Vector Regressor (SVR) for predicting travel times. Two best-performing models are selected, and their feature spaces are hybridized to boost feature space. On this boosted feature space, we employed SVR for final prediction. Our proposed HDFS-TTP ensemble can learn complex non-linearities in traffic data with the varying architectural design. The performance of our proposed HDFS-TTP ensemble using hybridized and boosted feature spaces showed significant improvement in test data in terms of Root Mean Square Error (62.27 ± 1.58), Mean Absolute Error (13.38 ± 1.09), Maximum Absolute Error (104.66 ± 2.77), Mean Absolute Percentage Error (2.50 ± 0.03), and Coefficient of determination (0.99714 ± 0.00044).

INDEX TERMS Travel time prediction (TTP), hybridized deep feature space (HDFS), machine learning (ML), recurrent neural network (RNN), heterogeneous ensemble.

I. INTRODUCTION

With the advent of Global Positioning Systems (GPS) based systems, there has been a significant surge of interest in Location-Based Services (LBS) among academics and industry. Travel Time Prediction (TTP) is one such application that acts as an essential service in Intelligent Transportation Systems (ITS); more specifically, in Advanced Traveler Information Systems (ATIS) and navigation applications. With an accurate and reliable ATIS, individuals can plan their trips more effectively, while transportation organizations and logistic companies can manage and run their operations more efficiently. In contrast to freeway Travel Time (TT), urban network TT is affected by many traffic (endogenous) and non-traffic (exogenous) factors making it a challenging task.

The existing TTP methods can be divided into route-based and data-driven approaches. The overall TT in route-based approaches is computed by adding segment time and transition time (waiting time due to signals, turns, etc.) between segments. Based on the formulation of overall TT,

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

route-based approaches are further divided into segment-based (uses segment time and ignores inter-segment correlation [1]) and path-based approaches (uses segment time and intersection delays [2], [3]). The TT estimate $\hat{y}_i$ of a path-based approach [4] is represented in equation 1.

$$\hat{y}_i = \sum_i (\hat{s}_i) + \sum_j (\hat{t}_j) \tag{1}$$

where $\hat{s}_i$ is the TT estimated for the i-th road segment and $\hat{t}_j$ is the time estimate during the j-th transition.

On the other hand, data-driven approaches formulate TT as a pure regression task and estimate TT of an entire path/route from historical data by implicitly modeling traffic complexities. Data-driven approaches are further divided into trajectory-based [5], [6] and Origin-Destination (OD) based approaches [7]. Trajectory-based approaches use road network and trajectory data, while origin-destination-based approaches only consider pickup and drop-off location data for travel time prediction.

Data-driven approaches, a substitute to conventional learning approaches, are often hybridized for different traffic-related problems to enhance overall prediction capabilities. There has been a rapid rise in the use of hybridized and ensemble approaches for TTP in recent years. As an example, eXtreme Gradient Boosting (XGB) approach is combined with GRU and Light Gradient Boosting Machine (LightGBM) for TTP in [8] and [9], respectively. Similarly, fusion of Multi-Layer Perceptron (MLP) and LightGBM is studied in [10]. The outputs of the base regressors are fed to a linear regression model, a Decision Tree model, and a linear weighted fusion model which act as meta-regressors in [8], [10], and [9], respectively. These ensemble approaches discussed above use the output (decision scores) of base regressors as features for meta-regressors to predict TT. However, the feature spaces of base regressors with a meta-regressor, both separately and jointly, are not studied in the literature.

This paper proposes a systematic solution based on state-of-the-art deep learning models. To overcome the shortcomings in existing literature, we modeled our TTP problem as a regression problem. Extensive exogenous and endogenous features like spatial features, temporal features, and weather features are augmented with traffic data for better prediction. Four deep learning models namely, Convolutional Neural Network (CNN), MLP, Long Short-Term Memory (LSTM), and GRU are studied as base regressors and their feature spaces are analyzed. We extracted and hybridized the feature spaces of the two best models obtained empirically i.e., LSTM and GRU. SVR is used as meta-regressor to predict TT on this boosted feature space.

The following text presents the synopsis of the scientific contribution of this study.

- We propose a novel Hybridized Deep Feature Space (HDFS) based ensemble for Travel Time Prediction (TTP), also collectively called as HDFS-TTP. Feature spaces of deep learning models are analyzed separately and in a hybridized manner and fed to SVR for TTP.
- We have augmented exogenous features with Floating-Cars Data (FCD) to enhance the overall performance of our proposed ensemble.
- We have also extracted Principal Component Analysis (PCA) features and encode GPS trajectories using Deep Stacked Auto-Encoder (DSAE) to boost feature space. The comparative analysis with baseline architectures shows considerable improvement in metrics like RMSE, MAE, Max. AE, MAPE and R^2 on the FCD dataset for our feature-based LSTM-GRU ensemble.

The remainder of this paper is organized as follows: Section II provides the historical background of the study. Section III presents the proposed methodology. Section IV elaborates on the results of our research. Section V contains the concluding remarks and the future direction.

II. BACKGROUND

Most of the earlier work on TTP employed segment-based approaches focusing on predicting TT on a selected set of routes or a specific freeway segment/region. Loop detector data has been extensively used to predict segment/link TT. Various approaches, including pattern matching [11], Least-Square minimization [12], Hidden Markov Model [13], Gradient Boosting Decision Tree [14], and XGB [15] have been proposed to model segment-based TT. Data fusion has also been studied to improve the prediction accuracy in [16]. However, the major drawback associated with segment-based approaches is that link delays at intersections or transition time from one link to another are not considered in the prediction process. This limitation makes the applicability of these approaches only limited to freeway scenarios.

Path-based approaches address the limitation of segment-based approaches to some extent by splitting the entire path into sub-paths and computing TT for each sub-path using historical trajectories to get the final prediction [2], [5], [17], [18]. Rahmani *et al.* in [2] presented an idea to concatenate sub-paths to estimate the entire path. The authors in [17] decomposed the entire trajectory path into a pathlet dictionary and then reconstructed the complete path with fewer pathlets, and estimation of TT is carried out from these pathlets. Li *et al.* in [18] extended the work towards personalized prediction of TT using pathlet dictionary and learned congestion patterns. However, the performance of these path-based studies could be impacted by the data sparsity problem.

In the last decade, data-driven approaches have been widely used in traffic forecasting with the surge in data collection technologies like hand-held devices, and vehicle navigation systems. These approaches solve the problem by learning the hidden spatiotemporal features of traffic data in an end-to-end fashion. For instance, Abdollahi *et al.* in [19] employed DSAE on an extensive feature set followed by an MLP for TTP. Similarly, Stacked Sparse Denoising Auto-Encoder (SSDAE) [20], Deep Belief Network (DBN) [21], CNN [22], GRU [23], LSTM [24], and Bidirectional LSTM (BiLSTM) [25] have been investigated for TTP in the recent past. Graph Neural Networks (GNNs) have recently shown state-of-the-art performance in various applications and are naturally suited for traffic-related problems [26]. GNNs have widely been adopted in different traffic problems, including traffic state prediction, traffic flow prediction, travel demand prediction, trajectory prediction, and many others [27]. The authors in [28] introduced attention-based GNN for TTP task in which Spatiotemporal correlation is learned by combining a gated convolutional neural network and GNN. Three components were used to learn spatiotemporal heterogeneous information corresponding to recent, daily, and weekly periods before the final prediction. A multi-layer Graph Convolutional Network (GCN) is proposed in [29]. An encoder-decoder module based on LSTM and BiLSTM is used to learn traffic patterns better. Instead of dealing with traffic prediction and contextual information separately in a prediction, Fang *et al.* in [30] proposed a Spatial-Temporal GNN (STGNN) that jointly models traffic prediction and contextual information using a 3D attention mechanism for better prediction accuracy. The presented solution is deployed on Baidu maps showing the proposed approach's effectiveness and robustness. Wang *et al.* [31] has also incorporated intersection direction and driver behavior with other features in an attention-based GCN for TTP and demonstrated improved performance.

For the traffic forecasting domain, data-driven approaches can be viewed as OD-based and trajectory-based approaches. OD-based approaches only consider the pickup location, drop-off location, and departure time from the historical trajectory to estimate TT [19], [32]. However, typical OD-based solutions suffer from data sparsity (Not every trip in the database matches the query departure time, pickup location and drop-off location). The authors in [7] proposed a way to handle missing trips (data sparsity) with the same origin, destination, and departure time in historical trajectories by exploiting neighboring trips. Jindal *et al.* [33] further improved the performance with a distance-based TTP. The authors in [32] augmented exogenous features such as weather and air quality with traffic data to enhance model performance. The computation of OD-based TTP is faster. This kind of prediction works well in situations when intermediate trajectories are not crucial such as freeway TT. However ignoring intermediate trajectory points in urbanized TTP leads to missing essential information like route choices made by the driver, the waiting time due to signalized arterial, etc. On the other hand, trajectory-based approaches use this information in the prediction process [25]. Fu *et al.* [34] applied classical CNN and time CNN on taxi trajectory data for spatial and temporal feature learning and augmented exogenous features to improve prediction. The authors in [35] employed CNN on a hybrid trajectory dataset (car and bus), and learned features are fed to LSTM for sub-path and whole path TT estimation. The authors in [36] and [37] transformed vehicle trajectories into images and employed CNN for spatial and temporal feature extraction on transformed images for TTP.

When compared to individual models, which often perform well, hybridization or the use of an ensemble of these approaches could further improve performance [38]. Corridor-level TTP has been carried out in [39] using an integration of SVR and Particle Filtering (PF). Network-wide TTP has been investigated in [40] using local smoothing and Probabilistic Principal Component Analysis (PPCA). LSTM and CNN are integrated in [41] followed by a fully-connected layer to predict TT. Shen *et al.* [42] have employed LSTM as a prediction layer on features learned using CNN-RNN models. The authors in [43] have hybridized DBN with quantile regression for highway TT prediction. In addition to hybridized models, ensemble-based approaches have also been developed for TTP. The output of GRU and XGB is combined in [8]. In another study, Zou *et al.* [10] have combined the output of Light Gradient Boosting Machine (LightGBM) and MLP using a decision tree model for TTP. Likewise, the authors in [9] have reported better results of an ensemble involving LightGBM and XGB as base regressors for the urban road network. In [4], Wide-Deep-Recurrent (WDR) models have been proposed that combine three models, namely, linear, MLP, and LSTM models to predict TT. All the above ensemble approaches have analyzed the impact of decision scores of machine learning and deep learning models for TTP. However, the impact of feature spaces of deep learning models and prediction of an ML model i.e., SVR for TTP have not been studied in prior literature. Moreover, exogenous features are not so extensively examined for TTP on a network scale. In our current work, we have augmented exogenous features including weather conditions, calendar data, peak hours data and fastest route data to our map-matched trajectories. Moreover, PCA features are extracted from pickup and drop-off location features. Finally, DSAE is employed to learn and encode GPS trajectories in a lower dimension. On the final feature set obtained after augmentation of exogenous features, PCA features, and encoded trajectories, we trained our meta-model in which SVR is used as a meta regressor and the feature spaces generated by LSTM, and GRU are fed as input to SVR for final prediction. The results demonstrate the superior performance of our proposed meta-learning based approach.

III. PROPOSED METHODOLOGY FOR TRAVEL TIME PREDICTION

Travel time prediction is a challenging task as it is affected by several exogenous and endogenous factors like the choice of route, time of the day (peak/non-peak hour), day of the week (week/weekend), weather condition (usually more time is needed to reach a destination in a bad weather situation). Ensembles are now widely considered the most advanced solution to many machine learning problems and address the limitations of a single model by adding diversity using multiple base learners (either homogeneous or heterogeneous), ultimately improving overall predictive performance. This diverse learning leads to a more robust model that sufficiently captures data's variance (distribution). Different approaches

FIGURE 1. Islamabad, Pakistan (Study area).

like voting, ensemble selection, and stacking have been used to combine base learners to form an ensemble model [44]. Our work proposes a stacking-based heterogeneous ensemble approach that utilizes the well-known and existing machine learning and deep learning models. Feature spaces of four existing deep learning models (CNN, MLP, LSTM, and GRU) are analyzed with SVR as a meta-regressor. Fig. 1 and 2 show the study area of our proposed approach and a brief overview of the proposed HDFS-TTP, respectively. The following section explains the stages of our proposed HDFS-TTP in detail.

A. MAP MATCHING

We used Open Source Routing Machine (OSRM) to map GPS trajectories onto the OpenStreetMap (OSM) street network [45]. Online requests to OSRM were pretty slow, so we configured an offline OSRM server in a docker environment to address the response time issue. To further speed up the process, we employed a parallelized mechanism involving batch processing and multi-threading published in [46]. The issues related to off-road mapping of cars and trackers' zero speed are resolved using the algorithm presented in our previous work [47].

B. FEATURE AUGMENTATION

Exogenous features are not thoroughly analyzed for TTP on a network scale. For instance, Chen *et al.* [48] have incorporated calendar features and road types in the prediction. Likewise, some studies consider weather information but not take into account other exogenous features like peak hours, calendar information, fastest route data etc. [49], [50]. Moreover, some studies have incorporated weather information, calendar information for a freeway [51] or corridor [52] or it is an OD-based prediction [19], [32]. Travel time is affected by weather conditions, time of day, day of the week, route choice, peak or non-peak hour, etc. We extracted and aggregated various spatio-temporal, and weather-related features in our integrated dataset. For instance, trip geospatial area and vehicle route during a trip strongly impacts TT. We extracted geospatial features such as total distance, segments, and intersections traversed by a vehicle during a trip using map-matching. Similarly, another important type of feature that affects TT is temporal features. For example, the TT during non-peak hours is extremely different and longer than during peak/rush hours. For temporal information, we extracted the time of the day, day of the week, day of the month, and month of the year features. Weather conditions also affect TT [53], so we included 18 weather conditions[1] in our final features set. These features are listed in Table 1. Other useful features contributing to accurate TTP are is_peak_hour, is_holiday, fastest_route_distance and fastest_route_time. The fastest route features as described in [54] are extracted using OSRM fastest route Application Programming Interface (API).[2] With the help of the Directorate of Traffic Engineering and Transportation Planning

[1] https://www.worldweatheronline.com/developer/ (accessed on 07 October 2021)

[2] https://project-osrm.org/docs/v5.5.1/api/#route-service

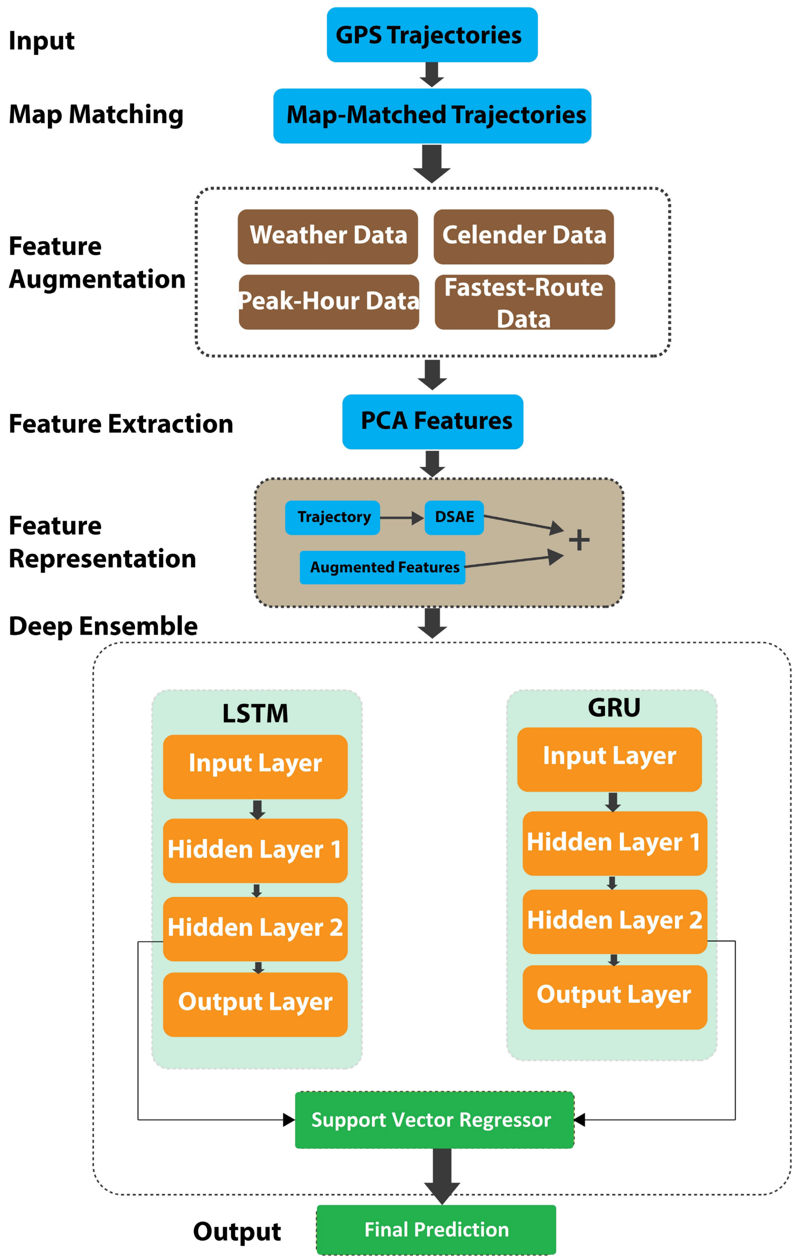


FIGURE 2. Overview of the proposed approach.

Islamabad, the is_peak_hour is calculated. Our data is then used to validate the feature.

C. FEATURE EXTRACTION AND REPRESENTATION

The basic idea of PCA is to retain maximum variance while reducing dimensionality. We employed PCA on pickup and drop-off locations to enhance and boost the feature space to get the top two uncorrelated (orthogonal) principle components [55]. These features were added to the feature space and DSAE as shown in Fig. 3 is employed to get the encoded representation of our GPS trajectory to improve feature representation. The trajectory data was encoded into eight features and appended these features to the final feature set. After data aggregation and feature representation, we removed anomalous trips with duration less than a minute (extremely short) and greater than two hours before final experimentation. Our data comprises trips between 0.5 and 60 kilometers.

In our dataset, the longest trip contains 99 GPS locations (latitude longitude pairs) which corresponds to 198 latitude and longitude points. After feature augmentation as discussed in Section III, B, we have increased the dimensionality of our

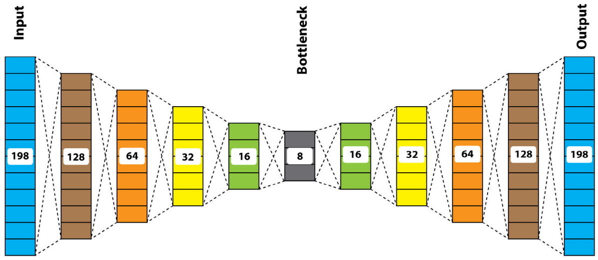


FIGURE 3. Proposed Deep Stacked Auto-Encoder.

TABLE 1. List of weather features used in the study, taken from https://www.worldweatheronline.com/developer/.

ID	Weather Conditions
1	Clear
2	Cloudy
3	Heavy rain
4	Light drizzle
5	Light rain
6	Light rain shower
7	Moderate or heavy rain shower
8	Moderate rain
9	Moderate rain at times
10	Overcast
11	Partly cloudy
12	Patchy light drizzle
13	Patchy light rain
14	Patchy light rain with thunder
14	Patchy rain possible
16	Sunny
17	Thundery outbreaks possible
18	Torrential rain shower

feature space. Our feature space comprises 253 features out of which 198 are GPS trajectory features. Deep auto-encoders are widely adopted as a data/feature compression technique in various domains [56]. A typical deep stacked auto-encoder consist of an encoder and a decoder with multiple layers each and a coded layer (also called bottleneck) as illustrated in Fig. 3. The basic idea is to learn the coded representation from the input first using encoder part and then reconstruct the input from the coded representation in the decoder part. This coded representation after training comprises maximum information needed to reproduce the input in a lower dimensional space. In our study, we learn this code representation of trajectory features into 8 features with DSAE. We added these encoded features to our final feature set.

We evaluated the feature importance of the final feature set we have obtained from the feature extraction and representation step before moving on to the implementation phase. One of the most widely used approach for feature importance is correlation coefficient which measures linear relationship between the features and target variable. However, mutual information regression has the capability to measure both linear and non-linear relationships between input features and the target variable [57]. Therefore, we have chosen mutual information regression for feature importance in this

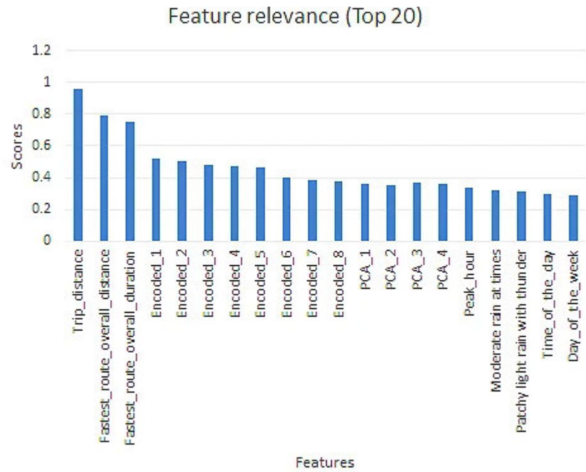


FIGURE 4. Feature importance based on mutual information regression.

study. Top 20 features with maximum gain are shown in Figure 4. As can be seen, trip distance, fastest route features, and encoded trajectory features (Encoded_1,…Encoded_8) haves shown the maximum relevance in the reduced space along with other features.

D. IMPLEMENTATION OF HDFS-TTP

Our implementation consists of two phases. 1) Investigating feature space of state-of-the-art Deep Learning models and 2) hybridizing the feature space of the best two models to boost feature space.

1) IMPLEMENTATION OF STATE-OF-THE-ART DEEP LEARNING MODELS

We trained four deep learning models (CNN, MLP, LSTM, and GRU) and used SVR [58] as a meta regressor. We extracted the feature spaces of individual models and fed them to SVR for the final prediction as it is based on structural risk reduction theory. SVR seeks to reduce test error and enhances the model's ability to generalise, in contrast to models based on empirical risk minimization theory [59]. To create a hybrid learning-based boosted feature space, we chose the two best-performing models i.e., LSTM and GRU, as our base feature extractors.

2) OUR PROPOSED HYBRID DEEP FEATURE SPACE-BASED TRAVEL TIME PREDICTION (HDFS-TTP)

LSTM and GRU performed best among the four models and are selected as our base regressors in the proposed HDFS-TTP. Also, these functionally similar network have been used in ensemble studies in literature e.g., [61] have shown promising results with LSTM and GRU when their intermediate layer activation is concatenated and fed to an MLP model for final prediction in [61]. Likewise, [62] have employed LSTM, GRU, Bi-LSTM, and Bi-GRU as base learners in their ensemble approach. Similarly, [63] and [64] have used these sequence learning models in their

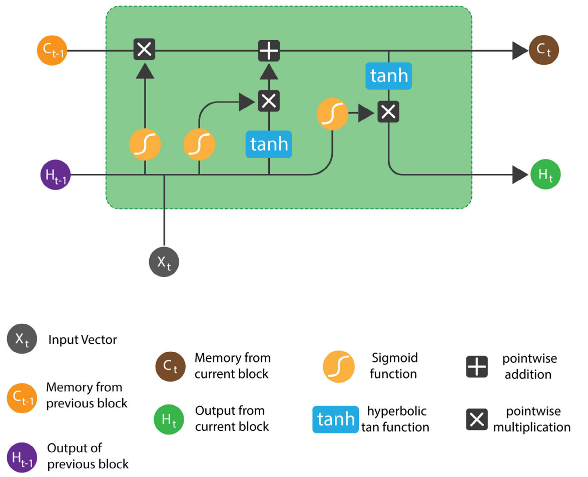


FIGURE 5. Structure of an LSTM cell [60].

ensemble-based approaches to further improve performance in various tasks. The features space of LSTM and GRU is represented as f_l and f_g. Equation 2 illustrates how an SVR model using the learnt feature space of recurrent models gives the final prediction.

$$\hat{Y} = SVR(f_l, f_g) \tag{2}$$

LSTM: Proposed Base Regressor: LSTM is a specialized Recurrent Neural Network (RNN) developed to address the problem of long-term dependencies in standard RNN model [65]. In traffic data, LSTM can learn segment-level and long-term information about nearby segments [60]. The structure of an LSTM cell depicted in Fig. 5 is much more complex than an RNN cell due to its gating mechanism (forget, input, and output gate). This gating mechanism allows LSTM to solve long-term dependencies by extending the memory cycle of the network. We used a two-layered LSTM as one of our base regressors for travel time prediction. Forget, input, and output gate computations of an LSTM cell are expressed in Equations 3-5.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

where f_t denotes forget gate, i_t denotes input gate, and o_t denotes output gate at time t, σ is the gate activation function (sigmoid). W_f, W_i, and W_o are the respective weight matrices of the three gates, and b_f, b_i, and b_o are their biases. h_{t-1} denotes the output/hidden state from previous timestamp and x_t is the input at current timestamp. LSTM cell state C_t and hidden output h_t are computed using the Equations 6, and 7, respectively. LSTM equations are taken from [66].

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \mu(W_c[h_{t-1}, x_t] + b_c) \tag{6}$$
$$h_t = o_t \otimes \mu(C_t) \tag{7}$$

where $\otimes$ represents point-wise multiplication, and μ and σ are tanh and sigmoid activation functions.

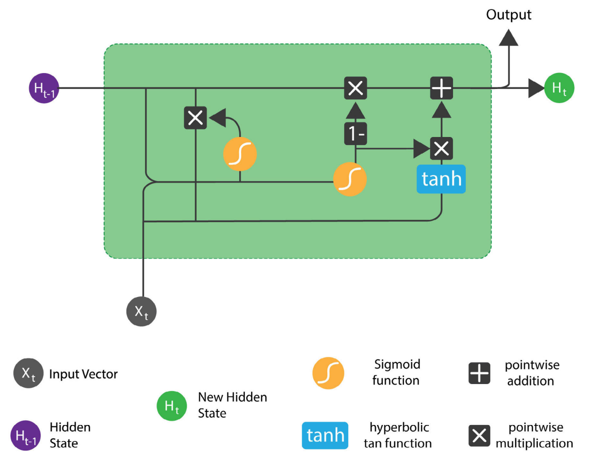


FIGURE 6. Structure of a GRU cell [67].

GRU: Proposed Base Regressor: Another enhanced RNN variation, the GRU uses two gates i.e., the update and reset gates, as compared to the three of the LSTM, making it more intuitive architecturally. [67]. GRU's overall efficiency is increased by the model's simplified architecture, which results in fewer parameters to train. The update gate in GRU replaces the input and forget gates of the LSTM. We employed a two-layer GRU model in this experiment. The structure of the GRU cell is depicted in Fig. 6, and the mathematics for the two gates of GRU to control the flow of information within the cell can be seen in Equations 8-11. The equations of GRU are taken from [68].

$$u_t = \sigma(W^u x_t + U^u h_{t-1}) \tag{8}$$
$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \tag{9}$$
$$h'_t = \mu(Wx_t + r_t \odot Uh_{t-1}) \tag{10}$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t \odot h'_t) \tag{11}$$

where u_t denotes update gate, r_t denotes reset gate, h'_t denotes memory content (current) and h_t denotes memory content (final) at time t, σ and μ are sigmoid and tanh activation functions. The symbol $\odot$ denotes element-wise multiplication whereas W^u, and U^u are the respective weight matrices of the two gates.

IV. RESULTS AND DISCUSSION

This section begins with a description of the data, followed by an explanation of the models that were used to analyse it and their results.

A. DATASET AND DATA DISTRIBUTION

We have collected and prepared a real-world anonymized FCD dataset obtained from a tracking company in Islamabad, Pakistan in the year 2019. In this experiment, we used

TABLE 2. Dataset statistics.

Attribute	Value
Trajectory count	724,402
Area	220 km^2
Sampling rate	15s-45s
Travel time mean	1109.50s
Travel time std	1173.51s
Travel distance mean	5986.96m
Travel distance std	6732.36m

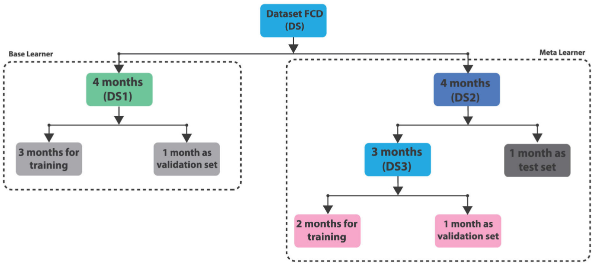


FIGURE 7. Data division between base and meta regressors.

data spanning eight months, from March 2019 to October 2019. For the specified time period, the dataset contains events recorded by 2895 unique tracker ids. Tracker units are mounted using the GSM Modem(Quectel M95) and GPS Chipset(U-Blox EVA-M8M). Our study uses data spanning the peak and off-peak hours, from 6:00 am to 11:00 pm. Details of the dataset are given in Table 2.

The data distribution scheme of our proposed HDFS-TTP approach is demonstrated in Fig. 7.

We have used 4 months of data (DS1) for base learners such that 3 months of data is used for training and validation is performed on the remaining 1-month data. Likewise, 4 months of data (DS2) is used for meta learner. From (DS2), 3 months of data (DS3) is used for training and validation of meta learner. The remaining 1-month data is used as a test set to report the error measures and evaluate the generalization of the proposed approach.

B. PERFORMANCE METRICS

We have evaluated our proposed model and baselines using five evaluation measures, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Maximum Absolute Error (Max. AE), Mean Absolute Percentage Error (MAPE), and Coefficient of determination (R^2). Let TT_i denote the actual travel time and $\hat{TT}_i$ denotes the predicted travel time, then RMSE can be expressed in Equation 12. These equations are taken from [73] and [74].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{TT}_i - TT_i)^2} \tag{12}$$

MAE refers to an average absolute error among the actual and estimated value and is given in Equation 13.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{TT}_i - TT_i| \tag{13}$$

Max. AE is the maximum absolute error among the actual and estimated value and is expressed in Equation 14.

$$Max.AE = max|\hat{TT}_i - TT_i| \tag{14}$$

MAPE is the average absolute percentage error among the actual and estimated value and is expressed in Equation 16.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{\hat{TT}_i - TT_i}{TT_i}| \tag{15}$$

The R^2 indicates how much of the variation is learned by the model and is shown in Equation 16.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}|(\hat{TT}_i - TT_i)|}{\sum_{i=1}^{n}|(\hat{TT}_i - TT_m)|} \tag{16}$$

Here TT_m refers to the mean travel time value. The ideal value for RMSE and MAE will be zero (or close to zero) and close to one for R^2 for the best prediction.

C. EXPERIMENTAL SETTINGS

All the simulations are performed using Python 3.7.16 and Keras (2.3.1), which is based on Tensor Flow (2.1.0). A machine equipped with an NVIDIA GeForce GTX 1070 Ti graphics card is used to train all the models.

D. BASELINES

As there is no prior research on our data, we have implemented four state-of-the-art deep learning architectures namely, MLP [70], CNN [69], LSTM [71], and GRU [72]. In addition to that, three related ensemble approaches [8], [10], and [9] were also implemented to compare with our proposed HDFS-TTP approach.

E. HYPER-PARAMETER SETTING

Table 3 lists the parameters we specified for our baseline NNs. Trial-and-error method is used to get these values. These best values for each parameter of the models presented in Table 3 were obtained after multiple experimental runs. For our base regressors, we've tweaked the learning rate, the number of hidden layers, the number of neurons in each hidden layer, and the batch size. The optimizer and activation function have been set to 'relu' and 'adam', respectively. Our suggested approach's results are validated using hold-out cross-validation (See Figure 7). Our proposed approach, in contrast to baselines, includes a machine learning-based meta model (SVR) with a pseudo-random behaviour (like other machine learning models). In order to demonstrate the robustness of our approach, we ran the experiment 10 times using the best parameters and the reported the results with confidence interval in Table 6, 7, 8, and 9.

F. PERFORMANCE EVALUATION OF STATE-OF-THE-ART DEEP LEARNING BASELINES AS FEATURE EXTRACTORS

The results of various deep learning models used as feature extractors for SVR on the entire dataset are presented in this section. Table 5 shows the outcome on overall data. As can be

TABLE 3. Parameters setting of baselines (Optimal).

Model	Base	Parameter	Value
CNN [69]	-	convolution layers	2
		max-pooling layers	1
		filter size	[64,32]
		kernel size	3
		pool size	3
		activation	relu
		optimizer	adam
		learning rate	0.0001
MLP [70]	-	layers	2
		no. of neurons	[64,64]
		activation	relu
		learning_rate	0.001
		optimizer	adam
		batch_size	256
LSTM [71]	-	layers	2
		no. of neurons	[64,64]
		activation	relu
		learning_rate	0.001
		optimizer	adam
		batch_size	128
GRU [72]	-	layers	2
		no. of neurons	[64,64]
		activation	relu
		learning_rate	0.001
		optimizer	adam
		batch_size	128
[8]	GRU	layers	2
		activation	relu
		learning_rate	0.001
		optimizer	adam
	XGB	learning_rate	0.05
		max_depth	7
		n_estimators	300
		objective	reg:squarederror
[10]	LightGBM	num_leaves	31
		learning_rate	0.05
		objective	rmse
	MLP	layers	2
		no. of neurons	[64,64]
		activation	relu
		learning_rate	0.001
		optimizer	adam
		batch_size	256
[9]	LightGBM	learning_rate	0.05
		objective	rmse
		num_leaves	31
	XGB	learning_rate	0.05
		max_depth	7
		n_estimators	300
		objective	reg:squarederror

observed, CNN is clearly not suitable for our data. The main reason is that CNN didn't take into account temporal aspects in making a prediction and failed to perform well on our data. Compared to a CNN model, MLP reduces the RMSE to 131.09 seconds and MAE to 25.77 seconds. However, these errors are still higher for practical purposes. In comparison to these models, LSTM and GRU, which are specialized time-series models performed significantly better on the same data. The RMSE with LSTM and GRU is reduced to 73.46 seconds and 69.81 seconds, respectively. It can be seen that SVR performed well on the feature spaces learned by LSTM and GRU models.

TABLE 4. Inference time of all the approaches.

Model	Overall(1k)	Without_ weather(1k)	Weekdays(1k)
CNN	0.476	0.324	0.341
MLP	0.160	0.145	0.142
LSTM	0.166	0.151	0.149
GRU	0.154	0.151	0.148
GRU+XGB [8]	0.181	0.172	0.173
MLP+LightGBM [10]	0.178	0.170	0.169
LightGBM+XGB [9]	0.177	0.169	0.174
Our Proposed HDFS-TTP	0.182	0.174	0.171

TABLE 5. Performance evaluation of baselines feature extractors.

Model	RMSE(s)	MAE(s)	MaAE(s)	MAPE(%)	R^2(%)
CNN	178.02	68.42	601.16	11.29	0.977268
MLP	131.09	25.77	351.32	4.80	0.987572
LSTM	73.46	24.50	142.93	4.72	0.996098
GRU	69.81	16.54	135.14	3.35	0.996475

TABLE 6. Performance evaluation of our Proposed HDFS-TTP on overall data.

Model	RMSE(s)	MAE(s)	MaAE(s)	MAPE(%)	R^2(%)
Our Proposed HDFS-TTP	62.27± 1.58	13.38± 1.09	104.66± 2.77	2.50 ± 0.03	0.99714± 0.00044

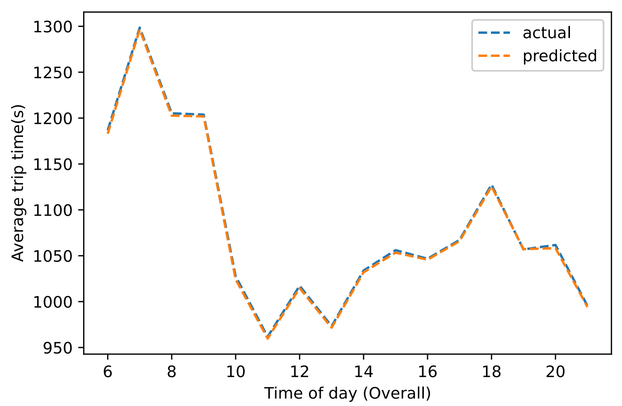


FIGURE 8. Normalized trip time using feature-based LSTM-GRU ensemble.

G. EVALUATING OUR PROPOSED HDFS-TTP ON OVERALL DATA

LSTM and GRU perform better as feature extractors than CNN and MLP, as was discussed in Section IV-F. The performance of these recurrent learning models will be further enhanced by concatenating their feature spaces [61]. Therefore, we have combined the feature spaces of LSTM and GRU to form a hybridized deep boosted feature space. Our proposed HDFS-TTP approach has shown further improvements in terms of RMSE (62.27 ± 1.58), MAE (13.38 ± 1.09), Max. AE (104.66 ± 2.77), MAPE (2.50 ± 0.03), and R^2 (0.99714±0.00044), as reported in Table 6. These approaches complemented each other when used as a merger for TTP and further improved the generalization of HDFS-TTP. Actual vs. expected normalized trip times for various times of the day, from 6:00 am to 11:00 pm, are shown in Fig. 8.

In order to demonstrate the generalizability of our proposed HDFS-TTP, we performed two experiments. In the first experiment, we analyzed the impact of weather features. In the second experiment, we tested our model on weekdays data only. In both scenarios, only a slight degradation in model performance is reported. In the next sections, we have discussed the details.

TABLE 7. Investigating the impact of weather conditions with baselines and proposed HDFS-TTP.

Model	RMSE(s)	MAE(s)	MaAE(s)	MAPE(%)	R^2(%)
CNN	195.52	85.90	669.83	14.14	0.972355
MLP	134.48	26.95	358.55	4.03	0.987029
LSTM	81.89	19.50	189.47	3.86	0.995190
GRU	77.33	17.86	178.91	2.90	0.995675
GRU+XGB [8]	84.96	33.96	198.27	5.36	0.994780
MLP+LightGBM [10]	67.95	24.91	132.54	3.33	0.996661
LightGBM+XGB [9]	67.91	24.99	131.75	3.64	0.996665
Our Proposed HDFS-TTP	64.60± 1.68	15.47± 1.12	112.35± 4.38	2.88 ± 0.05	0.99696± 0.00062

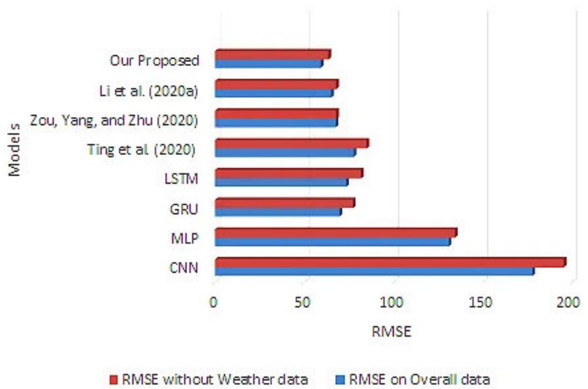


FIGURE 9. RMSE comparison with and without weather data.

1) IMPACT OF WEATHER ON MODEL PERFORMANCE

To show the significance of augmenting weather conditions with traffic data and the robustness of our model, we have evaluated the performance of our proposed ensemble and baselines on overall data without weather features. To examine the impact of weather data on overall performance, we excluded weather features from the data. The Table 7 summarises our findings. As demonstrated, removing weather data degrades the performance of deep models i.e., CNN, MLP, LSTM, GRU and ensemble models. RMSE of our proposed HDFS-TTP ensemble is increased from 62.27 ± 1.58 seconds to 64.60 ± 1.68 seconds. This indicates that weather features have a considerable effect on the overall prediction of TT. Fig. 9 shows the RMSE of our proposed feature-based LSTM-GRU ensemble and baselines. The impact of weather features on our proposed models and baselines is readily apparent.

TABLE 8. Performance evaluation of baselines & Proposed ensembles on weekdays data.

Model	RMSE(s)	MAE(s)	MaAE(s)	MAPE(%)	R^2(%)
CNN	223.14	85.02	708.04	11.30	0.964145
MLP	139.78	41.41	373.01	6.93	0.985930
LSTM	81.47	25.88	188.51	4.73	0.995269
GRU	74.90	19.31	169.23	3.14	0.996001
GRU+XGB [8]	74.11	31.94	191.95	5.11	0.996045
MLP+LightGBM [10]	78.87	30.26	189.92	2.80	0.995521
LightGBM+XGB [9]	65.24	23.78	126.89	3.26	0.996935
Our Proposed HDFS-TTP	64.02± 1.14	15.12± 0.89	111.56± 4.49	2.58 ± 0.02	0.997142± 0.00026

2) PERFORMANCE ON WEEKDAYS DATA

This experiment is performed on weekdays data. The results are reported in Table 8. There is a slight degradation in overall performance, which could be caused by the reduction in data size. An RMSE of 64.02 ± 1.14 is reported for our proposed HDFS-TTP ensemble. The RMSE of our proposed feature-based ensemble and baselines is demonstrated in Fig. 10. Even with weekend data removed, our model performs better than its counterparts.

The performance of [8], [10] and [9] deteriorates slightly on weekdays data. An ensemble proposed [8] in has RMSE and MAE of 74.11 and 31.94, respectively. The ensemble proposed in [10] has an RMSE and MAE of 78.87 and 30.26, respectively. Similarly, the RMSE and MAE of the ensemble proposed in [9] are 65.24 and 23.78, respectively.

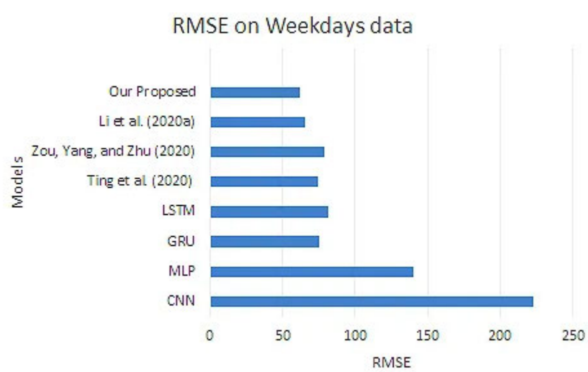


FIGURE 10. RMSE comparison of proposed ensembles & baselines on Weekdays data.

H. PERFORMANCE EVALUATION OF OUR PROPOSED HDFS-TTP WITH REPORTED ENSEMBLE APPROACHES ON OVERALL DATA

It is evident from the results shown in Table 9 that our proposed boosted feature space-based ensemble (HDFS-TTP)

TABLE 9. Performance comparison of proposed HDFS-TTP with reported ensemble approaches on overall data.

Model	RMSE(s)	MAE(s)	MaAE(s)	MAPE(%)	R^2(%)
GRU+XGB [8]	77.75	33.90	190.13	4.53	0.995629
MLP+LightGBM [10]	67.71	22.78	129.95	2.96	0.996685
LightGBM+XGB [9]	65.05	23.34	124.62	3.12	0.996940
Our Proposed HDFS-TTP	62.27± 1.58	13.38± 1.09	104.66± 4.77	2.50 ± 0.03	0.99714± 0.00044

performed considerably well compared to existing ensemble baseline approaches reported in the literature. Ting *et al.* in [8] combined the scores of XGBoost and GRU, yielding RMSE = 77.75 and MAE = 33.90. The authors in [10] combined the scores of LightGBM (another lightweight Gradient boosting tree model) and a deep learning model (MLP) and reported RMSE = 67.71 and MAE = 22.78. Li *et al.* [9] combined the scores of two decision tree-based ensemble models to improve the performance. The ensemble of LightGBM and XGBoost has an RMSE = 65.05 and MAE = 23.34. However, none of these approaches analyzed feature spaces of deep learning models with the capabilities of ML models separately or in a hybridized manner.

V. CONCLUSION

Travel time prediction (TTP) is an integral component of ITS as trip time is influenced by various factors such as weather and peak hours, demanding a multi-model to capture non-linearities in traffic data for accurate travel time prediction. We developed a novel Hybridized Deep Feature Space (HDFS) based TTP ensemble model (HDFS-TTP) based on a hybrid feature learning strategy. Various endogenous and exogenous data sources affecting travel time like peak hours, weather conditions, and calendar features are augmented with FCD data. We also included PCA features along with this data to enhance the feature space and used DSAE for dimensionality reduction. This data is fed to four models, CNN, MLP, LSTM, and GRU, and their feature spaces are analyzed with an SVR as a meta-regressor for TTP. Next, we concatenated the feature spaces of the best performing models like LSTM and GRU to form hybridized deep boosted feature space and used SVR on this boosted feature space for final prediction. We achieved a Root Mean Square Error of 62.27 ± 1.58, a Mean Absolute Error of 13.38 ± 1.09, a Maximum Absolute Error of 104.66 ± 2.77, a Mean Absolute Percentage Error (2.50 ± 0.03), and a Coefficient of determination of 0.99714 ± 0.00044 with our proposed hybrid learning-based ensemble. To further investigate the robustness of our proposed model, we removed weather features and weekend data from the dataset. The results demonstrated better performance of our proposed feature-based LSTM-GRU ensemble compared to baselines and a slight deterioration in overall performance. Our proposed approach is different as compared to ensemble approaches presented in the literature that rely on the decision score of the base regressors. Investigating other ML models as meta-regressors can further enhance the results. In addition to that, variants of DSAE such as variational AE, and denoising AE can be used to enhance the feature spaces prior to model training. In the future, we plan to incorporate decision scores with feature spaces of recurrent learning models. We also plan to evaluate the performance of Graph-based NNs on the same dataset.

ACKNOWLEDGMENT

The authors acknowledge the support of the Khalifa University of Science and Technology.

REFERENCES

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[2] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Route travel time estimation using low-frequency floating car data," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2292–2297.

[3] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Floating car and camera data fusion for non-parametric route travel time estimation," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1286–1291.

[4] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 858–866.

[5] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 25–34.

[6] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1695–1704.

[7] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, 2019.

[8] P.-Y. Ting, T. Wada, Y.-L. Chiu, M.-T. Sun, K. Sakai, W.-S. Ku, A. A. K. Jeng, and J.-S. Hwu, "Freeway travel time prediction using deep hybrid model—Taking sun yat-sen freeway as an example," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8257–8266, Aug. 2020.

[9] D. Li, D. Yu, Z. Qu, and S. Yu, "Feature selection and model fusion approach for predicting urban macro travel time," *Math. Problems Eng.*, vol. 2020, pp. 1–13, Jun. 2020.

[10] Z. Zou, H. Yang, and A.-X. Zhu, "Estimation of travel time based on ensemble method with multi-modality perspective urban big data," *IEEE Access*, vol. 8, pp. 24819–24828, 2020.

[11] H. Chen, H. A. Rakha, and C. C. McGhee, "Dynamic travel time prediction using pattern recognition," in *Proc. 20th World Congr. Intell. Transp. Syst.*, 2013, pp. 1–17.

[12] X. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, "Urban link travel time estimation using large-scale taxi data with partial information," *Transp. Res. C, Emerg. Technol.*, vol. 33, pp. 37–49, Aug. 2013.

[13] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 95–111, Jun. 2014.

[14] C.-M. Chen, C.-C. Liang, and C.-P. Chu, "Long-term travel time prediction using gradient boosting," *J. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 109–124, 2018.

[15] Z. Chen and W. Fan, "A freeway travel time prediction method based on an XGBoost model," *Sustainability*, vol. 13, no. 15, p. 8577, 2021.

[16] M. Martínez-Díaz and F. Soriguera, "Short-term prediction of freeway travel times by fusing input-output vehicle counts and gps tracking data," *Transp. Lett.*, vol. 13, no. 3, pp. 193–200, 2021.

[17] Y. Li, D. Gunopulos, C. Lu, and L. Guibas, "Urban travel time prediction using a small number of GPS floating cars," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2017, pp. 1–10.

[18] Y. Li, D. Gunopulos, C. Lu, and L. J. Guibas, "Personalized travel time prediction using a small number of probe vehicles," *ACM Trans. Spatial Algorithms Syst.*, vol. 5, no. 1, pp. 1–27, Jun. 2019.

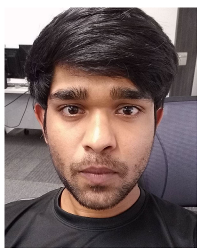
[19] M. Abdollahi, T. Khaleghi, and K. Yang, "An integrated feature learning approach using deep learning for travel time prediction," *Exp. Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112864.
[20] K. Tang, S. Chen, A. J. Khattak, and Y. Pan, "Deep architecture for citywide travel time estimation incorporating contextual information," *J. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 313–329, 2021.
[21] M. Wang, W. Li, Y. Kong, and Q. Bai, "Empirical evaluation of deep learning-based travel time prediction," in *Knowledge Management and Acquisition for Intelligent Systems* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2019, pp. 54–65.
[22] X. Ran, Z. Shan, Y. Shi, and C. Lin, "Short-term travel time prediction: A spatiotemporal deep learning approach," *Int. J. Inf. Technol. Decis. Making*, vol. 18, no. 4, pp. 1087–1111, 2019.
[23] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2659–2666, Apr. 2020.
[24] X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-based method with attention mechanism for travel time prediction," *Sensors*, vol. 19, no. 4, p. 861, 2019.
[25] H. Zhang, H. Wu, W. Sun, and B. Zheng, "DeepTravel: A neural network based travel time estimation model with auxiliary supervision," 2018, *arXiv:1802.02147*.
[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[27] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.
[28] H. Hong, Y. Lin, X. Yang, Z. Li, K. Fu, Z. Wang, X. Qie, and J. Ye, "Heteta: Heterogeneous information network embedding for estimating time of arrival," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 2444–2454.
[29] W. Zhang, Y. Wang, X. Xie, C. Ge, and H. Liu, "Real-time travel time estimation with sparse reliable surveillance information," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–23, 2020.
[30] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "ConST-GAT: Contextual spatial–temporal graph attention network for travel time estimation at Baidu maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 2697–2705.
[31] Z. Wang, H. Fu, G. Liu, and X. Meng, "Attention-based sequence learning model for travel time estimation," *IEEE Access*, vol. 8, pp. 221442–221453, 2020.
[32] T. Xu, X. Li, and C. Claramunt, "Trip-oriented travel time prediction (TOTTP) with historical vehicle trajectories," *Frontiers Earth Sci.*, vol. 12, no. 2, pp. 253–263, 2018.
[33] I. Jindal, X. Chen, M. Nokleby, and J. Ye, "A unified neural network approach for estimating travel time and distance for a taxi trip," 2017, *arXiv:1710.04350*.
[34] L. Fu, J. Li, Z. Lv, Y. Li, and Q. Lin, "Estimation of short-term online taxi travel time based on neural network," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Cham, Switzerland: Springer, 2020, pp. 20–29.
[35] X. Lin, Y. Wang, X. Xiao, Z. Li, and S. S. Bhowmick, "Path travel time estimation using attribute-related hybrid trajectories network," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1973–1982.
[36] T.-Y. Fu and W.-C. Lee, "DeepIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 69–78.
[37] W. Lan, Y. Xu, and B. Zhao, "Travel time estimation without road networks: An urban morphological layout representation approach," 2019, *arXiv:1907.03381*.
[38] I. Lana, J. D. Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, Apr. 2018.
[39] R. B. Sharmila, N. R. Velaga, and A. Kumar, "SVM-based hybrid approach for corridor-level travel-time estimation," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1429–1439, Sep. 2019.
[40] M. Cebecauer, E. Jenelius, and W. Burghout, "Integrated framework for real-time urban network travel time prediction on sparse probe data," *IET Intell. Transp. Syst.*, vol. 12, no. 1, pp. 66–74, Feb. 2018.
[41] Z. Zhang, P. Chen, Y. Wang, and G. Yu, "A hybrid deep learning approach for urban expressway travel time prediction considering spatial–temporal features," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 795–800.
[42] Y. Shen, C. Jin, J. Hua, and D. Huang, "TTPNet: A neural network for travel time prediction based on tensor decomposition and graph embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4514–4526, Sep. 2022.
[43] L. Li, B. Ran, J. Zhu, and B. Du, "Coupled application of deep learning model and quantile regression for travel time and its interval estimation using data in different dimensions," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106387.
[44] D. Gupta and R. Rani, "Improving malware detection using big data and ensemble learning," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106729.
[45] J. J. Arsanjani, A. Zipf, P. Mooney, and M. Helbich, "An introduction to openstreetmap in geographic information science: Experiences, research, and applications," in *OpenStreetMap GIScience* (Lecture Notes in Geoinformation and Cartography). Cham, Switzerland: Springer, 2015, pp. 1–15.
[46] A. Shamshad and I. U. Haq, "A parallelized data processing algorithm for map matching on open source routing machine (OSRM) server," in *Proc. 14th Int. Conf. Open Source Syst. Technol. (ICOSST)*, Dec. 2020, pp. 1–6.
[47] N. Zafar, I. U. Haq, J.-U.-R. Chughtai, and O. Shafiq, "Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas," *Sensors*, vol. 22, no. 9, p. 3348, 2022.
[48] M.-Y. Chen, H.-S. Chiang, and K.-J. Yang, "Constructing cooperative intelligent transport systems for travel time prediction with deep learning approaches," *IEEE Trans. Intell. Transp. Syst.*, early access, May 4, 2022, doi: 10.1109/TITS.2022.3148269.
[49] X. Zhu, Y. Fan, F. Zhang, X. Ye, C. Chen, and H. Yue, "Multiple-factor based sparse urban travel time prediction," *Appl. Sci.*, vol. 8, no. 2, p. 279, 2018.
[50] H. Yuan, G. Li, Z. Bao, and L. Feng, "Effective travel time estimation: When historical trajectories over road networks matter," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Jun. 2020, pp. 2135–2149.
[51] B. Qiu and W. Fan, "Machine learning based short-term travel time prediction: Numerical results and comparative analyses," *Sustainability*, vol. 13, no. 13, p. 7454, 2021.
[52] S. Yang and S. Qian, "Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 3, pp. 12–28, Jun. 2019.
[53] G. Qi, A. A. Ceder, Z. Zhang, W. Guan, and D. Liu, "New method for predicting long-term travel time of commercial vehicles to improve policy-making processes," *Transp. Res. A, Policy Pract.*, vol. 145, pp. 132–152, Mar. 2021.
[54] H. Huang, M. Pouls, A. Meyer, and M. Pauly, "Travel time prediction using tree-based ensembles," in *Proc. Int. Conf. Comput. Logistics*. Cham, Switzerland: Springer, 2020, pp. 412–427.
[55] C. Zhu, C. U. Idemudia, and W. Feng, "Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques," *Informat. Med. Unlocked*, vol. 17, Jan. 2019, Art. no. 100179.
[56] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.
[57] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, 2015.
[58] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1996, pp. 1–7.
[59] A. Roy, R. Manna, and S. Chakraborty, "Support vector regression based metamodeling for structural reliability analysis," *Probabilistic Eng. Mech.*, vol. 55, pp. 78–89, Jan. 2019.
[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[61] S. Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, "All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework," *IEEE Trans. Affect. Comput.*, vol. 13, no. 1, pp. 285–297, Jan./Mar. 2022.
[62] W.-I. Lai, Y.-Y. Chen, and J.-H. Sun, "Ensemble machine learning model for accurate air pollution detection using commercial gas sensors," *Sensors*, vol. 22, no. 12, p. 4393, 2022.
[63] N. Rahimi, F. Eassa, and L. Elrefaei, "One- and two-phase software requirement classification using ensemble deep learning," *Entropy*, vol. 23, no. 10, p. 1264, Sep. 2021.

[64] I. M. Kamal, H. Bae, S. Sunghyun, and H. Yun, "DERN: Deep ensemble learning model for short- and long-term prediction of Baltic dry index," *Appl. Sci.*, vol. 10, no. 4, p. 1504, 2020.
[65] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[66] K. Qiao, J. Chen, L. Wang, C. Zhang, L. Zeng, L. Tong, and B. Yan, "Category decoding of visual stimuli from human brain activity using a bidirectional recurrent neural network to simulate bidirectional information flows in human visual cortices," *Frontiers Neurosci.*, vol. 13, p. 692, Jul. 2019.
[67] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
[68] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *J. Amer. Med. Informat. Assoc.*, vol. 24, no. 2, pp. 361–370, Mar. 2017.
[69] Y. Hou and P. Edara, "Network scale travel time prediction using deep learning," *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 115–123, 2018.
[70] S. Innamaa, "Short-term prediction of travel time using neural networks on an interurban highway," *Transportation*, vol. 32, no. 6, pp. 649–669, 2005.
[71] Y. Duan, L. V. Yisheng, and F.-Y. Wang, "Travel time prediction with LSTM neural network," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1053–1058.
[72] J. Zhao, Y. Gao, Y. Qu, H. Yin, Y. Liu, and H. Sun, "Travel time prediction: Based on gated recurrent unit method and data fusion," *IEEE Access*, vol. 6, pp. 70463–70472, 2018.
[73] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.
[74] L. Qing, T. Yongqin, H. Yongguo, and Z. Qingming, "The forecast and the optimization control of the complex traffic flow based on the hybrid immune intelligent algorithm," *Open Electr. Electron. Eng. J.*, vol. 8, no. 1, pp. 1–7, 2014.

JAWAD-UR-REHMAN CHUGHTAI received the B.S. degree in computer science from Azad Jammu and Kashmir University, Muzaffarabad, Pakistan, in 2011, and the M.S. degree in software engineering from Bahria University, Islamabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree with the Faculty Development Program, Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Women University of Azad Jammu and Kashmir, Bagh. His research interests include machine learning, deep learning, trajectory pattern mining, intelligent transportation systems, and traffic data analysis.

IRFAN UL HAQ received the M.Sc. degree in physics from Government College University, Lahore, Pakistan, and the Ph.D. degree in cloud computing from the University of Vienna, Austria. He is currently a Principal Scientist at the Pakistan Institute of Engineering and Applied Sciences. His industrial experience includes development of industrial automation systems and large-scale distributed systems. His current research interests include big data analytics, cloud computing, the IoT, blockchain, and smart contracts, with applications in transportation, logistics, supply chain, and industry 4.0.

OMAIR SHAFIQ (Member, IEEE) is currently an Associate Professor with the School of Information Technology, Carleton University, Canada. He has published several peer-reviewed research papers in journals, book chapters, conferences, and workshops, and served in technical program committee for several conferences and workshops, co-organized conference and workshops. His research interests include data modeling, big data analytics, services computing, machine learning, and cloud computing.



MUHAMMAD MUNEEB received the M.Sc. degree in computer science from Khalifa University, Abu Dhabi, United Arab Emirates. He is currently working as a Research Associate with Khalifa University, under the supervision of Dr. Samuel. He works on inter-discipline problems. His research interests include algorithms, automation, genetics, medical image analysis, and optimization.