sensors

MDPI

Article

Applying Hybrid Lstm-Gru Model Based on Heterogeneous Data Sources for Traffic Speed Prediction in Urban Areas

Noureen Zafar [1,2], Irfan Ul Haq [1,*], Jawad-ur-Rehman Chughtai [1] and Omair Shafiq [3]

[1] Pakistan Institute of Engineering and Applied Sciences, Islamabad 44000, Pakistan; noureen_zafar@uaar.edu.pk (N.Z.); jawadchughtai@gmail.com (J.-u.-R.C.)
[2] University Institute of Information Technology, Pir Mehr Ali Shah University of Arid Agriculture, Rawalpindi 46000, Pakistan
[3] School of Information Technology, Carleton University, Ottawa, ON K1S 5B6, Canada; OmairShafiq@cunet.carleton.ca
* Correspondence: irfanulhaq@pieas.edu.pk

Abstract: With the advent of the Internet of Things (IoT), it has become possible to have a variety of data sets generated through numerous types of sensors deployed across large urban areas, thus empowering the notion of smart cities. In smart cities, various types of sensors may fall into different administrative domains and may be accessible through exposed Application Program Interfaces (APIs). In such setups, for traffic prediction in Intelligent Transport Systems (ITS), one of the major prerequisites is the integration of heterogeneous data sources within a preprocessing data pipeline resulting into hybrid feature space. In this paper, we first present a comprehensive algorithm to integrate heterogeneous data obtained from sensors, services, and exogenous data sources into a hybrid spatial–temporal feature space. Following a rigorous exploratory data analysis, we apply a variety of deep learning algorithms specialized for time series geospatial data and perform a comparative analysis of Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), and their hybrid combinations. The hybrid LSTM–GRU model outperforms the rest with Root Mean Squared Error (RMSE) of 4.5 and Mean Absolute Percentage Error (*MAPE*) of 6.67%.

Keywords: IoT; LSTM; GRU; CNN; ITS

Citation: Zafar, N.; Haq, I.U.; Chughtai, J.-u.-R.; Shafiq, O. Applying Hybrid Lstm-Gru Model Based on Heterogeneous Data Sources for Traffic Speed Prediction in Urban Areas. *Sensors* **2022**, *22*, 3348. https://doi.org/10.3390/s22093348

Academic Editors: Dieter Schramm and Philipp Sieberg

Received: 7 October 2021
Accepted: 3 December 2021
Published: 27 April 2022

1. Introduction

Recently, the explosive growth of sensors, the internet, and huge data generation have provided new avenues of storage, execution, and implementation opportunities for IoT-based applications. Where the real-time availability of heterogeneous data generated from a huge number of sensors has brought about novel possibilities, a new set of challenges has also emerged, which is primarily related with the need of methodologies and capabilities to optimally harness the power of heterogeneity of data. Unveiling new interpretations by first integrating various forms of heterogeneous data and then applying intelligent algorithms to it is one of the most exciting possibilities of the notion of the Connected World [1].

Intelligent Transportation System (ITS) plays an enabling role in the realization of the concept of smart cities [2]. ITS has a huge demand to integrate data directly from various sensors, services built upon sensors, and from a variety of other exogenous data sources. ITS heavily relies on the power of IoT to build multisource, multisensor, and multimodel service systems for the prediction of traffic speed [3,4].

Major challenges faced during the activities of the data integration pipeline include sparsity handling, anomaly detection and rectification, normalization, map matching, and data resolution. The resolution quality of the fused and integrated traffic data plays a significant role when prediction algorithms are applied to it. The spatial temporal nature of traffic data defines the two corresponding facets of its resolution on GIS maps. Spatial resolution identifies the length of the road segment on which prediction is computed whereas the temporal resolution means the minimum time interval during which the prediction is made. The spatial temporal structure is used to integrate traffic data from different sources, e.g., Floating Car Data and Google in our case. Data elements from different sources at specific road segments during a specific time interval are fused together, thus adding value to accuracy. Necessary transformations from Estimated Time of Arrival (ETA) to speed or vice versa are performed before the fusion. This may result in spatiotemporal gaps due to the varying sparsity of data sources that need to be addressed. Data from the exogenous sources, e.g., weather, holidays, or peak hours, are also weaved upon the same spatial temporal framework in a similar fashion.

In this research, a detailed workflow is developed for the data integration pipeline that transforms raw data extracted from different heterogeneous data sources, passing through various preprocessing, and integration activities finally resulting into an integrated hybrid feature space. Open Street Maps (OSM) was selected as the GIS mapping technology. The integrated data are mapped on the road segments between every two adjacent OSM nodes within the time intervals of 15 min for the whole 24 h of the day. Keeping in mind the spatial temporal nature of the data, machine learning and deep learning algorithms specialized for time series data were applied, and their results were compared.

The main contributions of the paper can be summarized as follows:

- We integrated the heterogeneous data sources of Intelligent transportation systems for data collected from a particular city in Pakistan and built the hybrid LSTM-GRU model.
- We predicted the traffic speed on the basis of heterogeneous traffic data sources including exogenous data sources, e.g., weather, event and, peak hours.
- The Hybrid LSTM-GRU model has been applied on time intervals varying from 15 min to 1 h and the effectiveness of the model has been evaluated.

The remainder of the paper is organized as follows: Section 2 presents the related work. Section 3 presents the proposed methodology based on Hybrid LSTM–GRU model. Section 4 provides the results and discussions. Section 5 concludes the paper and identifies the future research direction.

2. Related Work

The related work was organized in context with the integration of heterogeneous data sources obtained from a variety of sensors leading to deep learning techniques applied on the spatial–temporal traffic data.

Salanova Grau et al. [5] presented a novel auto-DL approach for tuning the 145 errors and 22 warnings hyperparameter of the LSTM model in order to reduce time consumption. However, the author worked only on temporal features, not spatial features, and also did not focus on multidata sources.

Tang et al. [6] improved the fuzzy neural network model in order to enhance the learning capability of the model. The proposed model consists of a supervised and unsupervised learning process. In unsupervised learning, a k-means is used to partition input samples, and in the case of supervised learning, a weighted recursive least squares estimator is used for the optimization of hyperparameters. However, spatial patterns were not considered, while periodic features were extracted from traffic flow data. The authors did not deal with heterogeneous data sources of the arterial road network.

Ma, Tao, Wang, and Yu [7] proposed the novel approach LSTM-NN to overcome the issue of backpropagated error decay and automatically determine the optimal time window for the time series data set. This means the short-term prediction of speed deals with NN and long-term prediction of speed deals with LSTM. The authors used microwave detector data for the prediction of speed. However, they have only addressed the temporal features. There was a need to address the spatial features as well as a need to investigate with different data aggregated levels that would act as an additional input and contribute towards the prediction of speed in a more accurate consistent manner. Salavona Grau et al. [1] proposed the framework that collects data from conventional(cameras, radars, and loops) as well as innovative (FCD and Bluetooth devices) technologies and analyzes the ETA and traffic flow on road networks. They are focused on data collection, filtration, and fusion of big data. They analyzed the conventional and innovative types of data sources with the help of statistical approaches such as correlation coefficients, R-square, absolute value, and percentages ranges. However, there is a need to check the spatial–temporal features of big data on the road networks with the help of a machine and deep learning models that are specially proposed to handle the time series and spatial patterns. Bratsas et al. [8] applied different machine learning models and figured out the forecasting effectiveness of the machine learning model on randomly selected dates, randomly selected roads with duration over eight consecutive 15 min intervals, and the whole day. The experiments show that SVR performed well with stable conditions and MPL performed well on greater variations. However, the paper does not analyze the specialized deep learning models that are designed to deal with spatial–temporal trends of the data.

Yang et al. [9] present the effects of heterogeneous data sources (parking meter transactions, traffic conditions, and weather conditions) and spatial–temporal nature of data on the parking occupancy by linking Graph CNN and LSTM deep learning models. However, the authors did not incorporate ETA, event, holiday, calendar, OSM, and count data sources in their work. The authors focused on single-scale occupancy prediction rather than multiple scales, i.e., parking meters to the level of aggregated zones.

In order to cope with chronological deviations for traffic forecasting, spatial–temporal patterns have been extensively addressed [10–23] using dynamic tensor completion method, LSTM, temporal graph convolutional network, multivariate regression model, GraphCNN-LSTM model, attention graph convolutional sequence-to-sequence model, and spatial–temporal residual graph attention network. The authors, however, have not considered the effects of heterogeneous and nonrecurrent events data sources for forecasting road traffic.

Zhang et al. [24] proposed the deep learning-based multitask learning model to predict traffic speed on the road networks. The authors used the hybrid approach to increase the performance of the proposed model on taxis GPS data. They extracted the spatial–temporal patterns with the help of the nonlinear Granger causality analysis method, and for hyperparameter tuning, they used the Bayesian optimization technique. However, the authors did not consider other modes of traffic and their influence on the prediction of speed.

Kong et al. [25] proposed the recommendation system of intelligent traffic based on LSTM deep learning model. However, they ignored the positive impact of performance by using a multimodel on intelligent traffic information and also did not make a comparison of the famous time-recurrent neural network models, e.g., CNN [26], GRU, and hybrid models.

Authors in [27–29] used hybrid graph convolutional neural networks, CNN-LSTM, and LSTM-NN models on real-time traffic speed. The hybrid model produced promising results; however, other data sources related to road traffic that have directly or indirectly influenced the prediction of speed on road network were not taken into account.

Authors in [30–32] used k-means clustering, PCA, SOM, SVM, and SVR for prediction of traffic speed on regular or irregular intervals. However, their prediction models can be improved with a fusion of nonrecurrent events(e.g., calendar, special events, accidents, and weather) and traffic flow analysis.

Li et al. [33] proposed the transfer-learning model to address the missing data, data insufficiency, and mitigate model overfitting problems and stack LSTM for considering the time series patterns. However, the authors did not analyze the exogenous data sources, traffic types, and different traffic modes data sets. Hybrid feature space may be helpful for the construction of rules for applying transfer learning on the specific area while considering spatial factors.

Mena-Oreja et al. [34] discussed the formation of congestion by using state-of-the-art error recurrent and deep learning models. The author conducted a survey in the field of transportation and identified which deep learning and error recurrent models are helpful while considering spatial–temporal factors and other traffic conditions. The author applied the error recurrent and deep learning models on real traffic data sets in order to generate the common benchmark under traffic congestion conditions and demonstrated that the error recurrent model shows better accuracy as compared to deep learning models. However, the authors do not focus on the impact of statistical and ensemble deep and machine learning models on traffic congestion conditions using real traffic data sets and also do not discus hybrid feature space and its impact on traffic congestion prediction.

Ren et al. [2] propose the hybrid integrated–DL model in order to capture both spatial–temporal dependencies on the prediction of citywide spatial–temporal flow volume. The authors proposed the hybrid LSTM and ResNet model in order to deal with spatial–temporal effects on traffic volume. However, the authors proposed that the model shows large prediction error in sparse spatial areas and sleeping hours. The authors do not focus on the applicability of other types of flows, e.g., passenger flow, bike flow, etc.

Yu et al. [35] introduced the piece-wise correlation function and Jenks clustering method with dynamic programming to fix relationship of segment intervals. They considered heterogeneous data sources, e.g., speed, traffic flow, density, and road occupancy for short-term speed prediction on 5 and 10 min time intervals. They prepared the results on the basis of only three days, i.e., 1 February to 3 February 2015 (from Sunday to Tuesday). They did not discuss the effects of the weekdays and nonweekdays on road traffic. The amount of data is very low, and it is difficult to explore the correlation function across the whole week.

Liu et al. [36] used attention CNN to forecast traffic speed. They used 29,952 records as a training set and 5760 records as the test set. The amount of data is very low due to coverage of single road and difficult to explore traffic trends on their adjacent roads. In an earlier work [37], we built a solution that used traffic and weather data to predict traffic congestion using Estimated Time of Arrival (ETA).

We proposed a hybrid LSTM–GRU approach on heterogeneous data sources which comprises 7,343,362 records of September 2020. We made a comprehensive comparison among famous time-recurrent neural network advanced deep learning models, e.g., CNN, GRU, and their combinations.

3. Proposed Methodology Based on Hybrid LSTM–GRU Model

We obtained Floating Car data from a regional trackers' company. A data integration pipe-line was developed to clean, preprocess, and integrate this data with other data sources. Data integration pipe-line caters data pertaining to FCD, Google, weather, peak hour, holiday, and OSM data sources. We addressed the issues related to the FCD data sources—some of which include zero speed adjustments, outlier removal, and map matching. A feature in FCD data provides the reason for a signal generation from the tracker. The reason may be regular time interval, ignition on/off, turn, etc. A map-matching technique was used for node correction whereas threshold value, parking info, and ignition on/off details were used for zero speed adjustment. Finally, all the data sources were aggregated on the basis of wayid attribute of OSM and time attribute at a regular interval of 15 min. This was followed by the normalization of road speed through Permissible Speed Limit (SPL) attribute and Speed Performance Index (*SPI*).

3.1. Data Sources

Data from the following data sources were obtained and integrated.

3.1.1. FCD Data Source

FCD data were obtained from a regional tracker company. The data set contained events generated by 2895 unique tracker ids for the whole month of September 2020. The tracker units were mounted with the sensors GSM Modem(Quectel M95) and GPS Chipset(U-blox EVA-M8M). The key features of Floating Car Data (FCD) include latitude, longitude, date time, address, location, direction, speed, reason, and unit id. We faced multiple issues related to data preparation of FCD data source. The following issues were identified in FCD data:

- There was an off-road mapping of cars. This could be due to two reasons. Either the car appears offroad because of inherent GPS error or because the car was actually parked somewhere off the road.
- A large number of speed values generated by trackers were zero. This again could be due to two reasons: either the car is parked or stuck in congestion. The congestion data needed to be distinguished from the data related to the parked cars.
- There was duplication of tuples.
- There are missing values causing spatial sparsity. This is because the FCD does not cover all segments of roads of the road network.

The issues pertaining to the missing values and off-road mapping of cars are resolved through the map-matching process described in Algorithm 1. When the nodes are placed correctly on the map, the in-between missing values can be suggested. The zero speed issue is resolved through the zero speed adjustment section described in Algorithm 1. Outliers are identified and rectified through the max speed attribute obtained from OSM. Any speed exceeding the max speed on a segment of the road is replaced with the max speed value. Speed Performance Index is used for the normalization of data.

Algorithm 1 Preprocessing and Data Integration Algo

```
Read coordinate point [latitude, longitude] from data
Initialize nodes-seg with empty start-point and end-point
for t = 1 to m do
    Initialize way-id to zero
    nodes-seg["start-point","end-point"]:=get_nearest_seg(latitude,longitude)
    if nodes-seg ["start-point"] is zero then
        Check end-point value and update the start-point value with previous node value
    else
        nodes-seg ["start-point"] and nodes-seg ["end-point"] represents to same road points
    end if
end for
Update way-id
data.append(way-id)
Update data
take input records from data
F = 0
for t = 1 to n do
    Initialize speed = data["FCD-speed"]
    Initialize elapsed time = data["FCD-elapsed-time"]
    if reason is "ignition-ON" then
        F = 1
        if next reason is "ignition-OFF" then
            if speed is zero then
                if  elapsed time >threshold then
                    continue F = 1
                end if
                if next speed is zero then
                    F = 0
                    delete record
```

Algorithm 1 *Cont.*

```
                else
                    keep record
                end if
            else
                F = 0
                keep record
            end if
        end if
        remove record
    end if
end for
Initialize the data to empty
for each seg do
    for each agg-min each day do
        Compute avg-speed = $\frac{1}{n}\sum x_i$
        Initialize s = length of the segment
        Compute eta = $L \div$ avg-speed
        data[“avg-speed”].append(“avg-speed”)
        data[“eta”].append(“eta”)
    end for
    if current_day is “Sunday” or “Saturday” then
        data[“holiday”] = 1
    else
        data[“holiday”] = 0
    end if
    data[“weather”] = get weather of the current day with respect to current location
end for
```

3.1.2. ETA Data Source

We decided to obtain Google Map’s data from more than 500 points of interest on important roads of Islamabad. We acquired data by sending start and endpoints to Google Maps API. The Google data source is an authentic data source and provides an estimated time of arrival information. Data obtained from Google can be easily mapped on OSM Maps. The key features of Google data are Source Latitude, Source Longitude, Max Speed, Date, Time, Destination Latitude, Destination Longitude, and Estimated Time Arrival.

3.1.3. OSM Data Source

The road network attributes were fetched from OSM’s Turbo Overpass API. It provides the tags of Islamabad which include start node, end node, highway types, max speed, min speed, way_id, and max length. Max speed is a feature of special importance that is utilized in the normalization of data and outlier identification and removal. Some roads do not have a max speed feature associated with them; therefore, we had to insert it manually.

3.1.4. Calendar Data Source

In order to identify the effect of traffic on holidays at a particular location, we need a calendar data source. Behaviors and patterns of traffic are highly dependent on holiday data. The calendar data source features include DateTime, Name, and Type.

3.1.5. Weather Data Source

Real-time data source is gathered from Yahoo and Dark Sky API on the basis of time and latitude and longitude. Table 1 shows the most relevant features of the proposed hybrid model. The maxspeed-real attribute is used to detect anomalies and normalized the speed.

From feature engineering process, we exploited the input feature set and then applied machine and deep learning models that are capable enough to consider spatial–temporal effects on transportation data sources. We also applied hybrid approaches such as LSTM-GRU, GRU-LSTM, CNN-LSTM, and CNN-GRU as these hybrid approaches enhance the performance of models and provide more accurate prediction of the speed of a specific road at a specific time.

$$SPI = (S_i^t / S_{PL}) * 100 \tag{1}$$

Table 1. Hybrid Feature Space.

Nature of Attributes	Data Type
Day	integer
Hour	integer
Startnode	integer
Endnode	integer
aggminutes	15 min time interval
Weather	char
maxspeed-real	integer
aggSpeed	integer
Holiday	boolean

3.2. Data Integration Pipeline

In the Integration Pipeline, we performed the transformation on FCD and Google data set. The map-matching process was used to obtain the data in spatial format. Hence, the first transformation in the pipeline is the map matching.

Figure 1 illustrates the data integration procedure. Data are collected from different data sources such as Google, tracking company, osm road, weather data, holiday data, and peak hour data. In addition to the map-matching procedure, the following activities were required for data transformation before the machine learning algorithms could be applied:

- Map matching of GPS points
- Handling the abnormal behavior of data
- Data generalization and transformation
- Calculating the average speed of road section.

In the Google data set, the data points needed to be mapped on the OSM nodes. In this way, we could divide long roads in smaller segments with each segment marked by two adjacent OSM nodes. For this purpose, the nearest API of OSRM was again used. We verified the mapping results by visualizing the nodes on OSM maps. The mapping of FCD data and Google traffic data on OSM nodes provided a mechanism to spatially unify both traffic data. For temporal aggregation, the data points of both data sources were aggregated for every 15 min for all the road segments defined on the OSM road network. The integrated data were then merged with holiday data based on the date field. The purpose of the abovementioned integration was to encounter the special effects of congestion on working hours and weekends during holidays. These merged data are further integrated with road attributes on the basis of wayid. Furthermore, to handle the environmental effects, we combined these integrated data with only those weather data parameters that affect the behavior of the traffic, i.e., rainy, visibility, etc.

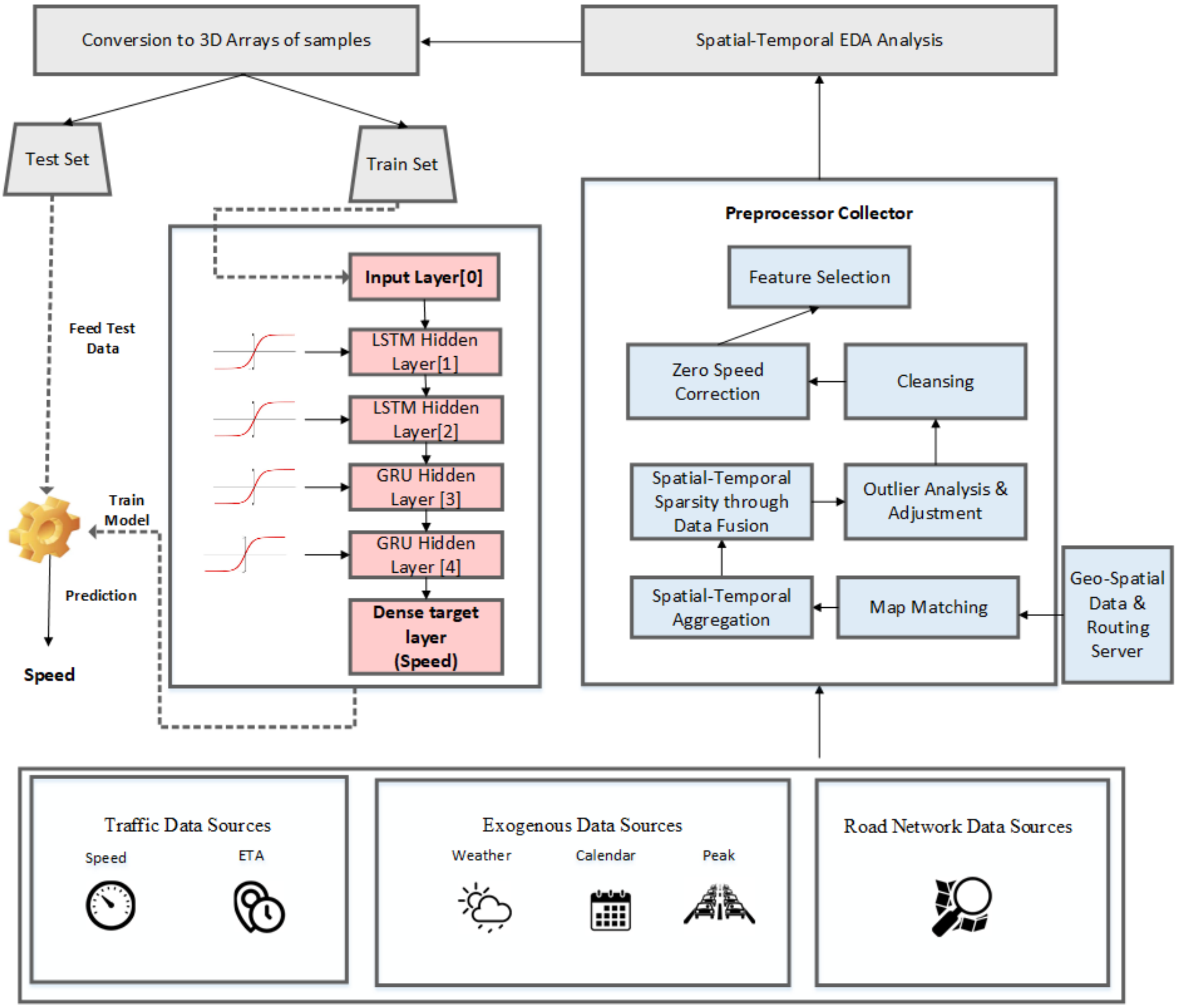


Figure 1. Urban traffic speed prediction based on hybrid LSTM–GRU model.

Algorithm 1 represents the procedure of map matching. The data contains the coordinate points with the latitude and longitude of the located geographical positions. Firstly, these coordinates were sent to the nearest API of OSRM server to obtain the pair of nodes of the segment containing the location of the driving vehicle that had already been determined from different sources. This was followed by the ordering of nodes on OSM maps to identify whether the road is incoming or outgoing. Occasionally, OSRM nearest API returns zero value in place of start node that might be due to multiple options for the nearest start node owing to the junctions on roads. For zero values, the end-node value was traced back on the OSM road information, and the immediate previous node was assigned to the start-node value.

3.3. Model Selection

We have applied the existing machine learning, classical deep learning and advanced deep learning models on the integrated traffic data. We have applied the well-known models such as XGBoost [38] (with Decision Tree [39] as weak learner), Linear Regression [40], K-Nearest Neigbor (KNN) [41], Multilayer Perceptron (MLP) based on MLPRegressor implementation in sklearn library (available at https://scikit-learn.org, accessed on 1 October 2021) [8], Artificial Neural Network (ANN) based on KerasRegressor implementation in keras library (available at https://keras.io, accessed on 1 October 2021) [1], Long Short Term Memory (LSTM) [42], Gated Recurrent Unit (GRU) [43], Convolutional Neural Networks

(CNN) [44], and well-known possible combinations of the models such as LSTM-CNN, CNN-LSTM, CNN-GRU, GRU-CNN, GRU-LSTM and LSTM-GRU models. Due to the spatial-temporal nature of the data models based on LSTM and GRU techniques generated better results. The Hybrid LSTM-GRU model has produced the most promising results.

In the following subsections we briefly describe the existing and well-known LSTM and GRU techniques and then move on to describe our hybrid model based on combining these existing techniques.

3.3.1. LSTM

Long Short Term Memory (LSTM) [42] is a variant of recurrent neural network (RNN) [5,45]. It is specialized for time series data. A generalized LSTM unit consists of three gates (i.e., input, output and a forget) and a cell. Cells are used to memorize the values of data and flow the information to output and forget gate. It is used to address vanishing gradient problem.

3.3.2. GRU

Gated Recurrent Unit (GRU) [43] is an advanced and more improved version of LSTM. It is also the type of recurrent neural network. It uses less hyper parameters because of reset gate and update gate as contrast to three gates of LSTM. Update gate and reset gate are basically vectors and are used to decide which information should be passed to the output.

3.3.3. Hybrid LSTM-GRU Model Description

Since, our dataset is time series and regression problem, we generated the results by using both classical as well as deep learning techniques. Our hybrid approach techniques yielded the low *RMSE* and is thus more effective as compared to classical regression techniques e.g., KNN, XGBoost, Linear Regressor, ANN and MLP. In our hybrid LSTM-GRU model, we first applied LSTM. LSTM is used to tackled the problem of vanishing gradient in backpropagation. LSTM contain three gates e.g., input gate (i_g), forget gate (f_g) and output gate (o_g). Gates are used to store information in memory. It stores the information in analog format. These gates are element-wise multiplied by sigmoid function ranges between 0–1. If the value of the gate is zero then this information is ignored or discarded else remained in memory. Tanh [46] is a well-known non-linear activation function and ranges between -1 to +1. In order to avoid information fading, a second derivative is used. The sigmoid function [47] is also well-known as a non-linear activation function. A sigmoid function contains values between 0 to 1. It is basically used to suggest which information should stay or drop from memory units known as gates. The mathematical Equations of input gate(i_g), forget gate (f_g) and output gate (o_g) are taken and adapted from the literature and are explained in Equations (2)–(4). Whereas, GRU contains two gates e.g., update gate (u_g) and reset gate (r_g). The output of the LSTM was passed to GRU during this approach. x_i^t is the input feature set that contains hybrid feature space(start-node, end-node, way-id, day, hour, agg-minutes, quarter, holiday, peakhour, mazspeed-real) at specific time and location. The aggregate speed is the target or output label, similar to [48] in which authors used LSTM and Bi-Directional LSTM Models to predict stock price.

The details and equations presented are taken and adapted from the literature such as [42,43].

Input Gate: $i_g \rightarrow represents \quad input \quad gate$
Forget Gate: $f_g \rightarrow represents \quad forget \quad gate$
Output Gate: $o_g \rightarrow represents \quad output \quad gate$
Update Gate: $u_g \rightarrow represents \quad update \quad gate$
Reset Gate: $r_g \rightarrow represents \quad reset \quad gate$
$\sigma \rightarrow represents \quad sigmoid \quad function$
$w_x \rightarrow represents \quad weight \quad for \quad the \quad respective \quad gate\,(x)$
$h_{t-1} \rightarrow output \quad of \quad the \quad previous \quad lstm \quad block \quad at \quad timestamp \quad t-1$

$x_t \rightarrow input \;\; at \;\; current \;\; timestamp$
$b_x \rightarrow biases \;\; for \;\; the \;\; respective \;\; gates\,(x)$
Cell Output: $C_t \rightarrow memory \;\; at \;\; timestamp\,(t)$
Cell Input: $\sim C_t \rightarrow represents \;\; candidate \;\; for \;\; memory \;\; at \;\; timestamp\,(t)$

The Cell Input state is $\sim C_t$, Cell Output state is C_t, and LSTM consists of three gates i_g, f_g, and o_g. GRU consists of two gates u_g and r_g. The hidden layers of LSTM–GRU model are $\sim C_t$, $\sim h_t$, and h_t. The weights of LSTM are w_i, w_f, w_o, and w_c. The weights of GRU are w_u, w_r, w_o, and w_{C_t}. LSTM–GRU model have biases b_i, b_f, b_o, and b_c. tanh is known as the hyperbolic tangent function. The ratio of the hyperbolic sine and corresponding hyperbolic cosine functions is defined in terms of tanh function. The scalar products of two vectors are represented as $\circ$.

When x_t is passed to the input network unit, it is multiplied by its own weight (w_i), and h_{t-1} is also multiplied by its own weight (w_i) and then added the bias (b_i). A h_{t-1} holds the information of previous units $t-1$. It passes to the sigmoid function and converts values between 0 and 1 and updates the status of the cell. The details and equations presented are taken and adapted from the literature such as [42,48,49].

$$i_g = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$f_g = \sigma(w_f[h_{t-1}, x_i] + b_f) \tag{3}$$

$$o_g = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{4}$$

Equations (5) and (6) describes how to produce the result between 0 and 1 using sigmoid activation function . $\sim C_t$ and C_t are used to decide what information is kept in memory and what information is forgotten. $\sim C_t$ is multiplied by the tanh function and decides which value is more significant.

$$\sim C_t = tanh(w_c[h_{t-1}, x_t] + b_c) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \sim C_t \tag{6}$$

The details and equations presented are taken and adapted from the literature such as [43,48]. Equations (7) and (8) explain that C_t is passed as an input to the first layer of GRU (u_g), whereas u_g and h_{t-1} are multiplied to weight and this information is forwarded to reset gate (r_g).

$$u_g = \sigma(w_u[C_t] + w_u[h_{t-1}]) \tag{7}$$

$$r_g = \sigma(w_r[C_t] + w_r[h_{t-1}]) \tag{8}$$

h_t decides information to be kept. The stayed information is then attached to the output layer. Same layer contains tanh as an activation function that is used to predict road traffic speed at specific time and location. This is also discussed in Equations (9)–(11). We used adam as an optimizer and mean squared as loss function in this regression problem.

$$\sim h_t = tanh(w_{C_t} + r_g \circ w_{C_t}[h_{t-1}]) \tag{9}$$

$$h_t = u_g \circ h_{t-1} + (1 - u_g) \circ \sim h_t \tag{10}$$

$$h_t = o_g * tanh(h^t) \tag{11}$$

In this study, the proposed LSTM–GRU model was applied to the data collected from Google and FCD which comprises 7,343,362 records of September 2020. The traffic condition data were captured every fifteen minutes of arterial roads in Islamabad, Pakistan. The proposed stacked LSTM–GRU architecture consisted of four hidden layers with 256 hidden units each. tanh was used as an activation function in all hidden layers. In the dense layer, we used one unit. The linear activation function was used in the output layer. We employed holdout crossvalidation to split data set into training and test sets. Then, we trained the model on the training data set by a batch learning approach using batch size of

512. This was followed by checking generalization of model on test data set. To evaluate the performance of the proposed deep architecture, we adopted *RMSE*, *MAE*, and *MAPE* as performance measures. The final configuration of the proposed LSTM–GRU model is summarized in Table 2. In the proposed model LSTM–GRU, we used four hidden layers (two for LSTM and the rest for GRU). We tested various configurations of hidden units, i.e., 4, 16, 32, 64, 128, and 256. We achieved the optimal results with 256 hidden units. Likewise, tanh activation function in the hidden layers and linear activation function in the output layer gives optimal results. Similarly, we varied the number of epochs from 5 to 50 and choose 10 as the optimal value. Moreover, we used batch size 512, learning rate 0.001, and loss function adam as the optimal parameter values.

Table 2. Hyperparameters configuration for XGBoost, ANN, KNN, MLP, and Hybrid LSTM–GRU.

Model	Hyperparameters	Values
XGBOOST	objective	linear
	n-estimators	4000
ANN	input dimension	10
	activation function	relu
	loss function	*RMSE*
	optimizere	adam
	epoch	100
	batch-size	512
KNN	K	20
	loss	*RMSE*
MLP	activation function	relu
	loss function	*RMSE*
	hidden-layer-size	100
	optimizer	SGD
	learning rate	0.001
LSTM-GRU	Batch Size	512
	Learning Rate	0.001
	No of epochs	10
	No of Hidden Layers	04
	Hidden Units	256
	Dropout Ratio	0.2
	Activation Function	tanh
	Output-Units	1
	Output-Type	Single Label
	Output-Layer-Activation-Function	linear
	Optimizer	Adam
	Loss Function	mean squared error

3.3.4. Performance Measures for Proposed Hybrid LSTM–GRU Model

Well-known and existing performance evaluation metrics are used to judge or measure the model performance and pattern. It also indicates the best model in order to achieve the output label performance. To evaluate the solution, we have used the existing performance evaluation metrics [50] such as *RMSE* (Root Mean Square Error), *MAPE* (Mean Absolute Percentage Error) and *MAE* (Mean Absolute Error).

The details and equations for the existing performance evaluation metrics presented below are taken and adapted from the literature such as [50–52]. *RMSE* is also known as RMSD (root-mean-square deviation). It is the square root of the mean squared difference between desired output and predicted output. The same is explained in Equation (12).

$$RMSE = \sqrt{(\sum_{i}(Y_{dsi} - Y_{psi})^2 / number\,of\,observations)} \tag{12}$$

where,

Y_{dsi} = desired speed
Y_{psi} = predicted speed
n = number of observations

MAPE (mean-absolute-percentage-error) is robust to large outliers. It eliminates the scaling factor and explains the error in the form of percentage. The formula of *MAPE* is explained in Equation (13).

$$MAPE = median(|(Y_{dsi} - Y_{psi})/Y_{dsi}|) \tag{13}$$

In our scenario, *MAE* (Mean-Absolute-Error) is used to define how far our predicted output speed is from the desired output speed. Mathematically, we can explain this from the following Equation (14).

$$MAE = 1/n \sum_i |(Y_{dsi} - Y_{psi})| \tag{14}$$

4. Results and Discussion

In this study, we worked on the regression data set with the multitimestamp and single label. Some features related to traffic patterns were derived in each segments at specific interval of time such as minimum estimated time and maximum speed per segment from integrated data.

Data were captured at 15 min time resolution and taking the space resolution less than or equal to 1 km.

4.1. Exploratory Data Analysis

We analyzed the influence of weekdays and weekend on traffic patterns and speed fluctuation. Speed Performance Index (*SPI*) [53] is a derived feature and calculated by using the following formula in the Equation (15):

$$SPI = ((V_O)/V_m) * 100 \tag{15}$$

where,

V_O = *the current speed for the road segment;*
V_m = *the permissible max speed for the road segment.*

SPI also provides a normalized expected speed which prevents having extreme values.

Figure 2 presents the average speed performance index (*SPI*) vs. time of day on weekdays. Different working days, i.e., Monday to Thursday, are shown with different color lines. In Figure 2, it can be visualized that there is a major change in the *SPI* over different hours of the day. In the morning rush hours (5:00–6:00 a.m., 9:00–10:00 a.m.,12:00–13:00 p.m.), the *SPI* is 66%, 64.6%, and 63%, respectively, which is higher than the average of morning hours. During the evening rush hour (around 8:00–9:00 p.m.), the *SPI* is 55%. The different trend of the *SPI* in the different time slots is not only compelling for model training but also in taking the average of the *SPI* for each time slot and for the prediction of speed on the basis of historical data. Various time slots including 5:00–6:00 a.m., 9:00–10:00 a.m., 12:00–13:00 p.m., and 8:00–9:00 p.m. show high traffic congestion. Friday's *SPI* trend is different from other working days. On Friday, during the morning rush hour (10:00 a.m.–12:00 p.m.), the *SPI* is 65.7%, which is higher than the average of morning hours. Friday traffic is different from other weekdays due to the Friday prayer, which is offered during 1–2 p.m. During the evening rush hour (9:00–10:00 p.m.), *SPI* is 60%, which is higher than all the time of day.

SPI variations on weekends are addressed in Figure 3. This figure depicts that weekend *SPI* trend is extremely opposite from weekdays *SPI*. On Saturday, rush hours are 10:00 a.m.–1:00 p.m. in the morning and 3:00–4:00 p.m., 9:00–10:00 p.m. in the evening, whereas on Sunday, rush hours are 12:00 p.m. to 1:00 p.m. in the morning and in the evening, 9:00 p.m. to 10:00 p.m., respectively.

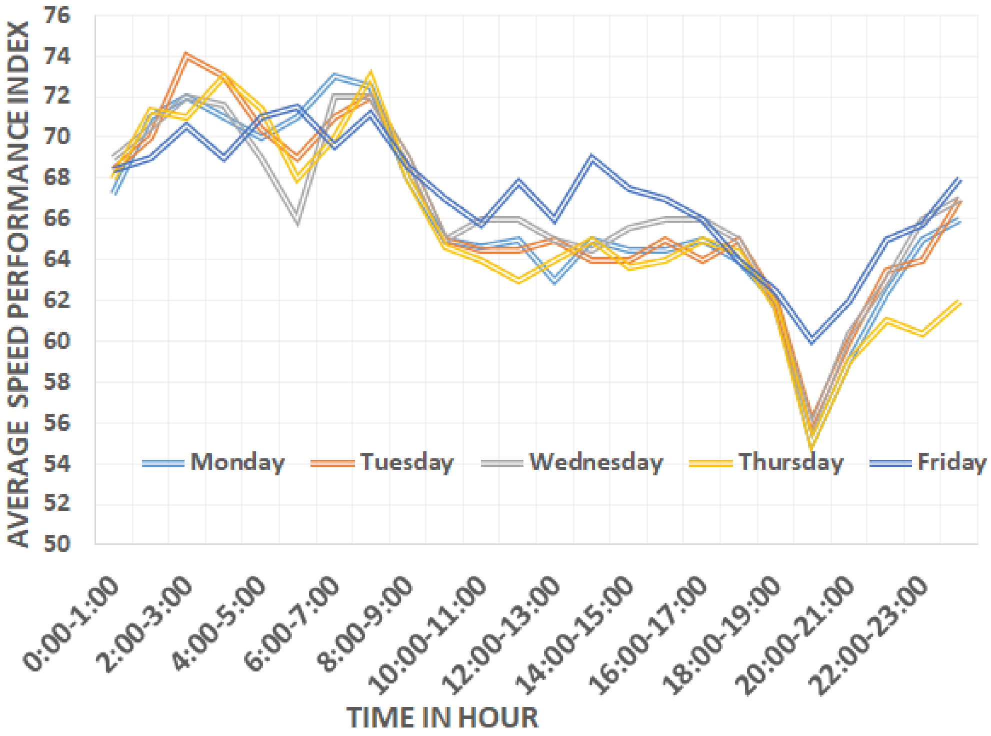


Figure 2. Speed Performance index variation on weekdays.

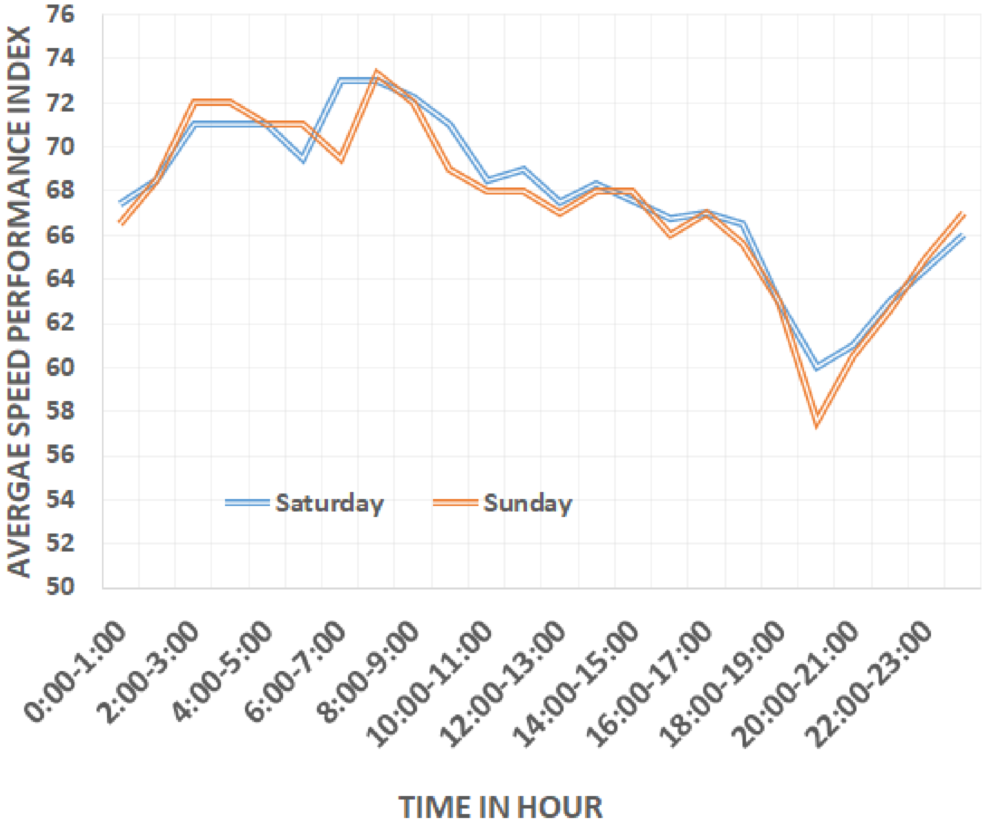


Figure 3. Speed performance index variation on weekends.

4.2. Feature Selection

Feature selection is a method that was used to explore the most relevant feature set by using correlation. The correlation of each feature was calculated and compared with the target variable. The correlation ranged between −1 and +1. +1 shows the positive correlation or perfect correlation, whereas −1 shows negative correlation and zero value means nonexistence of correlation. Basically, we were dealing with numeric input and target features. As the most relevant and famous techniques that deal with numeric data are correlation feature selection and mutual feature selection techniques, we used the same ones in our study.

4.2.1. Correlation Feature Selection Technique

Correlation is a statistical measure used to identify how two features changed together. We used Pearson correlation coefficient(PCC) [54] in our scenario.

Equation (16) explains that PCC is a standard measure of linear correlation between the two features. The formula of PCC is the ratio between the covariance of two features divided by their standard deviations. It deals between normalized data that range between −1 and +1.

$V[:,j]$ = feature of V_j in hybrid data set.
j = is a variable.
$mean(V[:,j])$ = mean of the V_j feature of hybrid data set.
W = feature of W in hybrid data set.
$mean_W$ = mean of the W feature of hybrid data set.
$std(V[:,J]$= Standard deviation of $V[:,j]$.
$std(W)$ = Standard deviation of W.

$$((V[:,j]-mean(V[:,j]))*(W-mean_W))/(std(V[:,J])*std(W)) \tag{16}$$

Table 3 depicts that start-node, way-id, hour, peak hour, and max speed-real are most relevant features as they are positively correlated, because in transportation domain start-node, way-id shows spatial impact, whereas hour and peak hour show temporal impact. Max speed-real shows the permissible speed of the road. We selected the 9 most correlated features from 29 combined features of all heterogeneous data sources. The correlation shows how all features are close to the target feature.

Table 3. Feature selection through correlation feature selection technique.

Features	Scores
$start-node$	**14,218.665540**
$end-node$	**52.788980**
$way-id$	**19,974.123749**
day	**487.561578**
$hour$	**24,238.112125**
$agg-minutes$	**0.380622**
$quarter$	**25.339737**
$holiday$	**620.959742**
$peakhour$	**9876.836458**
$maxspeed-real$	**3,227,692.593161**

4.2.2. Mutual Information Regression Feature Selection Technique

The mutual information feature selection technique is a method that works on information gain [55]. It works on the decision tree principle and calculates the entropy of each feature by calculating the information gain of each feature. Entropy helps the decision tree draw boundaries and measures disorder and uncertainty in the available data set. The most relevant feature has the highest information gain. Figure 4 indicates that start-node, end-node, way-id, and max speed have the highest information gain and are therefore the most relevant feature set.

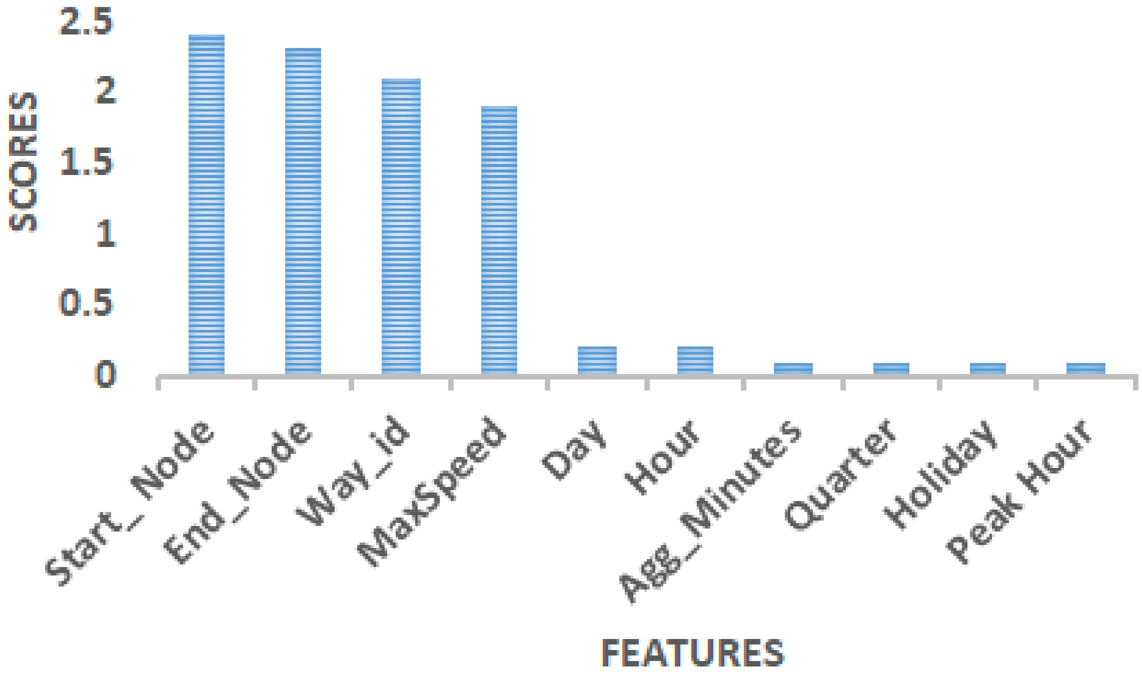


Figure 4. Feature selection using mutual information regression feature selection technique.

4.2.3. Heat Map of Hybrid Feature Space

Heat Map is a visualization style for analyzing the intensity, density, patterns, outliers, and variance of the feature set. It provides a correlation among all features. Figure 5 shows that maxspeed-real have the highest correlation with target agg-speed followed by way-id, day, and holiday features among the other heterogeneous features. Here, maxspeed-real denotes the permissible road speed limit. It helps to identify outliers and trends and patterns of road speed. In this way, we can tackle the missing speed and also normalize the data set in order to predict the true speed of a specific road. Figure 6 depicts that LSTM–GRU yields the lowest *RMSE*, i.e., 4.5 as compared to deep learning technique LSTM with a *RMSE* yield of 4.86 and classical regression techniques such as KNN with an *RMSE* yield of 6.03. Because the LSTM–GRU model handles both spatial–temporal effects, LSTM–GRU is specialized in time series data set. LSTM contains the temporal effects, and GRU contains the spatial effects. Because we have worked on transport data set and in the domain of transportation prediction of speed highly depends on specific time and location.

4.3. Hybrid LSTM–GRU Results

Table 4 elaborates the three performance metrics, i.e., *RMSE*, *MAE*, and *MAPE*, with respect to lowest *RMSE*-generated model, i.e., hybrid LSTM–GRU and GRU–LSTM models. In Figure 7, the prediction horizon indicates the capturing of data intervals such as 15, 30, 45, and 60 min. This helps in analyzing the time resolution impact on the model training. As we increased the sliding window, our test data performance improves, which proves a direct proportion of sliding window with the performance metrics. Table 4 contains the different performance metrics behavior on multiple individuals as well as hybrid deep learning models. As per *RMSE* and *MAE* evaluation metrics, hybrid LSTM–GRU produced the lowest *RMSE* of 4.5, *MAE* of 2.03, and *MAPE* of 6.67% being the lowest error.

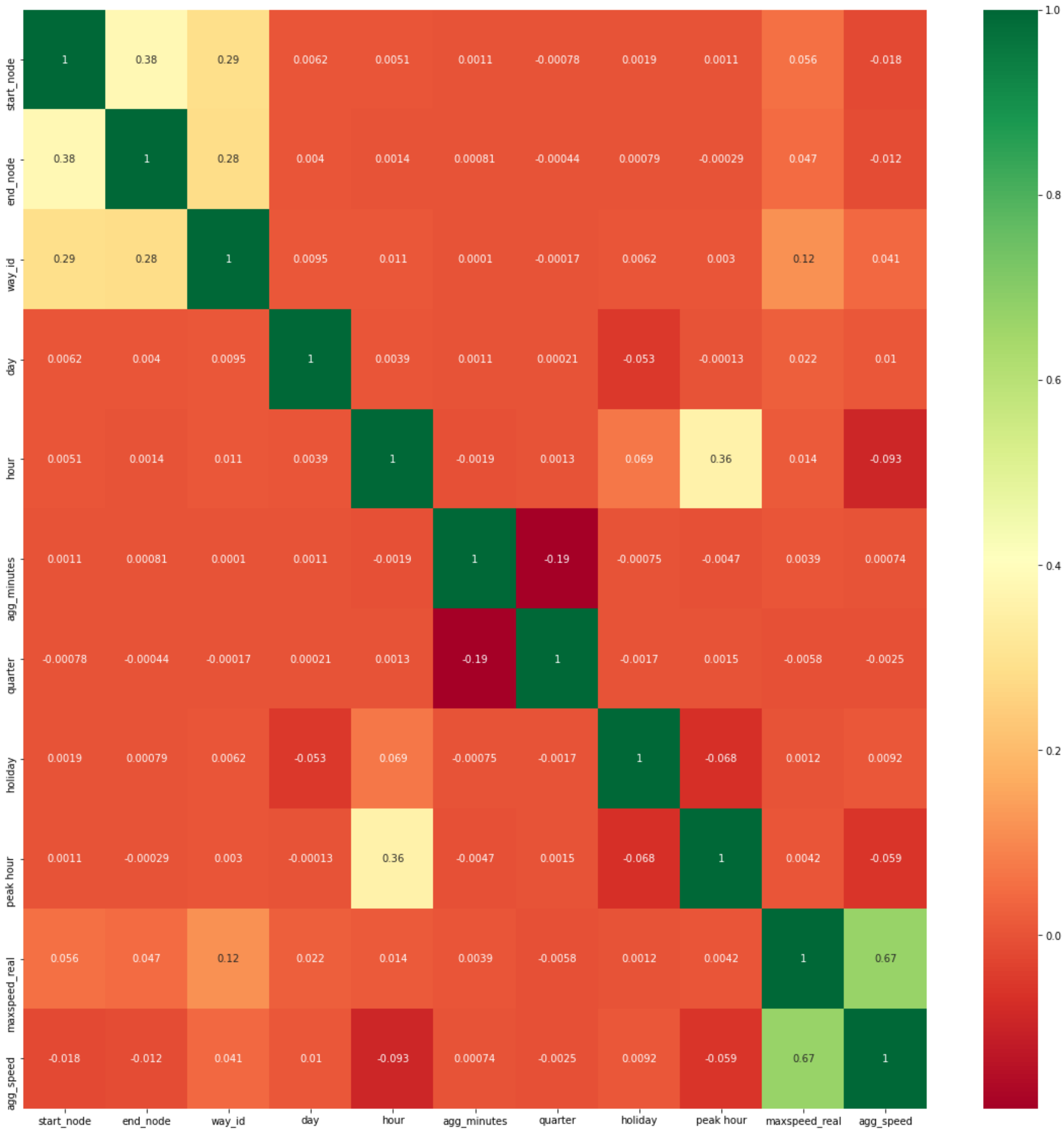


Figure 5. Heat map of hybrid feature space.

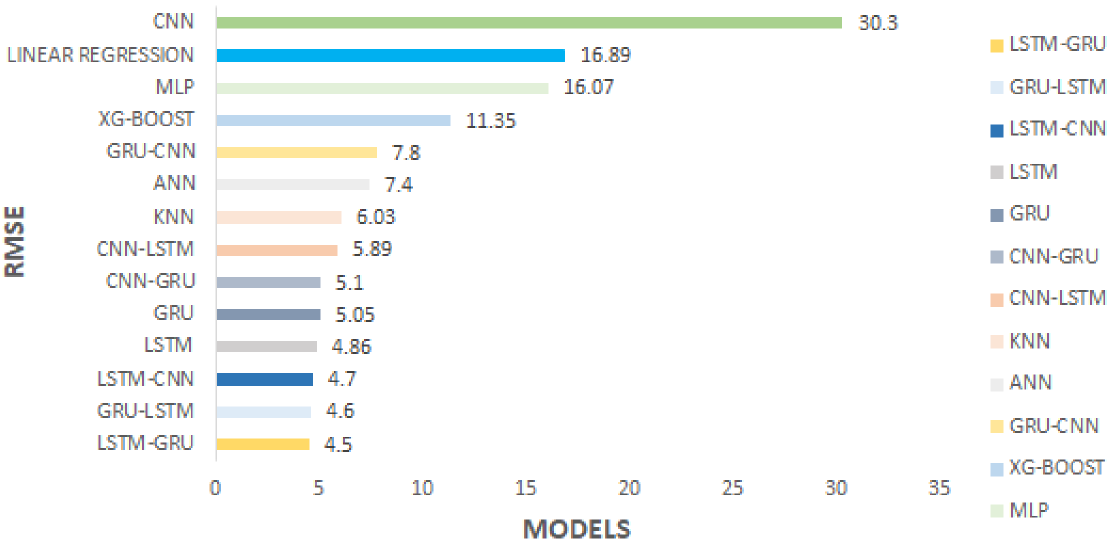
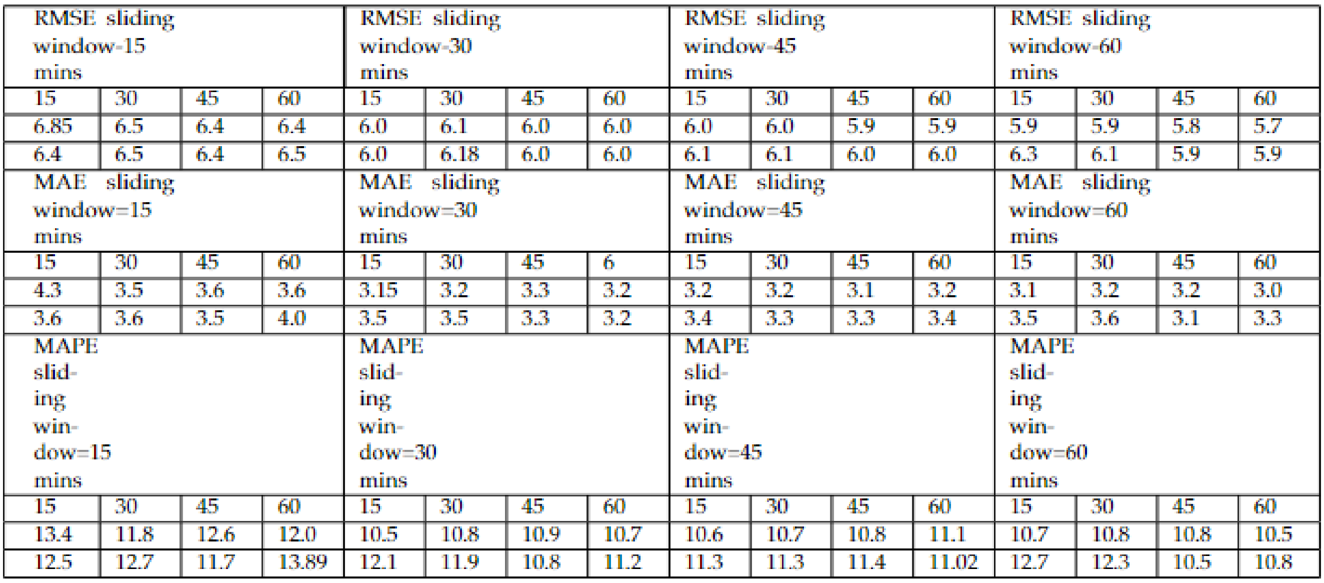


Figure 6. Evaluation metrics of prediction on test data.



RMSE sliding window-15 mins				RMSE sliding window-30 mins				RMSE sliding window-45 mins				RMSE sliding window-60 mins			
15	30	45	60	15	30	45	60	15	30	45	60	15	30	45	60
6.85	6.5	6.4	6.4	6.0	6.1	6.0	6.0	6.0	6.0	5.9	5.9	5.9	5.9	5.8	5.7
6.4	6.5	6.4	6.5	6.0	6.18	6.0	6.0	6.1	6.1	6.0	6.0	6.3	6.1	5.9	5.9
MAE sliding window=15 mins				MAE sliding window=30 mins				MAE sliding window=45 mins				MAE sliding window=60 mins			
15	30	45	60	15	30	45	6	15	30	45	60	15	30	45	60
4.3	3.5	3.6	3.6	3.15	3.2	3.3	3.2	3.2	3.2	3.1	3.2	3.1	3.2	3.2	3.0
3.6	3.6	3.5	4.0	3.5	3.5	3.3	3.2	3.4	3.3	3.3	3.4	3.5	3.6	3.1	3.3
MAPE sliding window=15 mins				MAPE sliding window=30 mins				MAPE sliding window=45 mins				MAPE sliding window=60 mins			
15	30	45	60	15	30	45	60	15	30	45	60	15	30	45	60
13.4	11.8	12.6	12.0	10.5	10.8	10.9	10.7	10.6	10.7	10.8	11.1	10.7	10.8	10.8	10.5
12.5	12.7	11.7	13.89	12.1	11.9	10.8	11.2	11.3	11.3	11.4	11.02	12.7	12.3	10.5	10.8

Figure 7. Evaluation metrics of prediction on test data.

Table 4. Performance metrics of deep learning models.

Model	*RMSE*	*MAE*	*MAPE*
LSTM	4.86	2.13	6.95
GRU	5.05	2.29	7.7
CNN	30.3	25.96	64.10
LSTM − *CNN*	4.7	23.9	7.9
CNN − *LSTM*	5.89	3.53	11.47
GRU − *LSTM*	4.6	2.08	6.85

Table 4. *Cont.*

$\boldsymbol{LSTM-GRU}$	**4.5**	**2.03**	**6.67**
$CNN-GRU$	5.1	2.4	8.4
$GRU-CNN$	7.8	4.5	14.6

5. Conclusions

This paper utilizes the speed performance index (*SPI*) as the road network state evaluation indicator. We integrated heterogeneous data, i.e., traffic, GPS, weather, special condition, and OSM obtained from a variety of sensors and services. We analyzed the behavior of transportation data sources with the help of different machine and deep learning algorithms. The LSTM–GRU model proved to be the most effective hybrid model among all time series deep learning and classical machine learning models with a net *RMSE* yield of 4.5. In the future, we plan to automatically label the classes using fuzzy logic and k-means clustering, followed by analyzing the results automatically by using optimization of hyper parameters and statistical models.

Author Contributions: Conceptualization, N.Z. and I.U.H.; methodology, N.Z.; software, N.Z.; validation, N.Z., J.-u.-R.C. and I.U.H.; formal analysis, N.Z.; investigation, I.U.H., J.-u.-R.C. and O.S.; resources, N.Z. and O.S.; data curation, N.Z.; writing—original draft preparation, N.Z. and J.-u.-R.C.; writing—review and editing, N.Z., I.U.H. and O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the grant TDF02-261 under Technology Development Fund (TDF) program of Higher Education Commission of Pakistan(HEC).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Salanova Grau, J.M.; Mitsakis, E.; Tzenos, P.; Stamos, I.; Selmi, L.; Aifadopoulou, G. Multisource data framework for road traffic state estimation. *J. Adv. Transp.* **2018**, *2018*, 9078547. [CrossRef]
2. Ren, Y.; Chen, H.; Han, Y.; Cheng, T.; Zhang, Y.; Chen, G. A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 802–823. [CrossRef]
3. Aitkenhead, M.; Cooper, R. Neural network time series prediction of environmental variables in a small upland headwater in NE Scotland. *Hydrol. Process. Int. J.* **2008**, *22*, 3091–3101. [CrossRef]
4. Liu, X.; Liu, Y.; Song, H.; Liu, A. Big data orchestration as a service network. *IEEE Commun. Mag.* **2017**, *55*, 94–101. [CrossRef]
5. Yi, H.; Bui, K.H.N. An automated hyperparameter search-based deep learning model for highway traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5486–5495. [CrossRef]
6. Tang, J.; Liu, F.; Zou, Y.; Zhang, W.; Wang, Y. An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2340–2350. [CrossRef]
7. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
8. Bratsas, C.; Koupidis, K.; Salanova, J.M.; Giannakopoulos, K.; Kaloudis, A.; Aifadopoulou, G. A comparison of machine learning methods for the prediction of traffic speed in urban places. *Sustainability* **2020**, *12*, 142. [CrossRef]
9. Yang, S.; Ma, W.; Pi, X.; Qian, S. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 248–265. [CrossRef]
10. Impedovo, D.; Dentamaro, V.; Pirlo, G.; Sarcinella, L. TrafficWave: Generative deep learning architecture for vehicular traffic flow prediction. *Appl. Sci.* **2019**, *9*, 5504. [CrossRef]
11. Li, L.; Lin, H.; Wan, J.; Ma, Z.; Wang, H. MF-TCPV: A Machine Learning and Fuzzy Comprehensive Evaluation-Based Framework for Traffic Congestion Prediction and Visualization. *IEEE Access* **2020**, *8*, 227113–227125. [CrossRef]
12. Chen, X.M.; Zahiri, M.; Zhang, S. Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach. *Transp. Res. Part C Emerg. Technol.* **2017**, *76*, 51–70. [CrossRef]

13. Candel, A.; Parmar, V.; LeDell, E.; Arora, A. *Deep Learning with H_2O*; H_2O.ai Inc.: Mountain View, CA, USA, 2016.
14. Fang, M.; Tang, L.; Yang, X.; Chen, Y.; Li, C.; Li, Q. FTPG: A Fine-Grained Traffic Prediction Method With Graph Attention Network Using Big Trace Data. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
15. Zhang, S.; Li, S.; Li, X.; Yao, Y. Representation of traffic congestion data for urban road traffic networks based on pooling operations. *Algorithms* **2020**, *13*, 84. [CrossRef]
16. Tan, H.; Wu, Y.; Shen, B.; Jin, P.J.; Ran, B. Short-term traffic prediction based on dynamic tensor completion. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2123–2133. [CrossRef]
17. Shin, D.H.; Chung, K.; Park, R.C. Prediction of Traffic Congestion Based on LSTM Through Correction of Missing Temporal and Spatial Data. *IEEE Access* **2020**, *8*, 150784–150796. [CrossRef]
18. Liu, H.; Xu, H.; Yan, Y.; Cai, Z.; Sun, T.; Li, W. Bus arrival time prediction based on LSTM and spatial-temporal feature vector. *IEEE Access* **2020**, *8*, 11917–11929. [CrossRef]
19. Xu, H.; Ying, J. Bus arrival time prediction with real-time and historic data. *Clust. Comput.* **2017**, *20*, 3099–3106. [CrossRef]
20. Servos, N.; Liu, X.; Teucke, M.; Freitag, M. Travel Time Prediction in a Multimodal Freight Transport Relation Using Machine Learning Algorithms. *Logistics* **2020**, *4*, 1. [CrossRef]
21. Zhang, Z.; Li, M.; Lin, X.; Wang, Y.; He, F. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transp. Res. Part C Emerg. Technol.* **2019**, *105*, 297–322. [CrossRef]
22. Bogaerts, T.; Masegosa, A.D.; Angarita-Zapata, J.S.; Onieva, E.; Hellinckx, P. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* **2020**, *112*, 62–77. [CrossRef]
23. Cai, P.; Wang, Y.; Lu, G.; Chen, P.; Ding, C.; Sun, J. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. Part C Emerg. Technol.* **2016**, *62*, 21–34. [CrossRef]
24. Zhang, K.; Zheng, L.; Liu, Z.; Jia, N. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing* **2020**, *396*, 438–450. [CrossRef]
25. Kong, F.; Li, J.; Lv, Z. Construction of intelligent traffic information recommendation system based on long short-term memory. *J. Comput. Sci.* **2018**, *26*, 78–86. [CrossRef]
26. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
27. Cao, M.; Li, V.O.; Chan, V.W. A CNN-LSTM model for traffic speed prediction. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
28. Essien, A.; Petrounias, I.; Sampaio, P.; Sampaio, S. Improving urban traffic speed prediction using data source fusion and deep learning. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
29. Zhang, T.; Jin, J.; Yang, H.; Guo, H.; Ma, X. Link speed prediction for signalized urban traffic network using a hybrid deep learning approach. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2195–2200.
30. Asif, M.T.; Dauwels, J.; Goh, C.Y.; Oran, A.; Fathi, E.; Xu, M.; Dhanya, M.M.; Mitrovic, N.; Jaillet, P. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *15*, 794–804. [CrossRef]
31. Ye, Q.; Szeto, W.Y.; Wong, S.C. Short-term traffic speed forecasting based on data recorded at irregular intervals. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1727–1737. [CrossRef]
32. Yao, B.; Chen, C.; Cao, Q.; Jin, L.; Zhang, M.; Zhu, H.; Yu, B. Short-term traffic speed prediction for an urban corridor. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 154–169. [CrossRef]
33. Li, J.; Guo, F.; Sivakumar, A.; Dong, Y.; Krishnan, R. Transferability improvement in short-term traffic prediction using stacked LSTM network. *Transp. Res. Part C Emerg. Technol.* **2021**, *124*, 102977. [CrossRef]
34. Mena-Oreja, J.; Gozalvez, J. A Comprehensive Evaluation of Deep Learning-Based Techniques for Traffic Prediction. *IEEE Access* **2020**, *8*, 91188–91212. [CrossRef]
35. Yu, D.; Liu, C.; Wu, Y.; Liao, S.; Anwar, T.; Li, W.; Zhou, C. Forecasting short-term traffic speed based on multiple attributes of adjacent roads. *Knowl.-Based Syst.* **2019**, *163*, 472–484. [CrossRef]
36. Liu, Q.; Wang, B.; Zhu, Y. Short-term traffic speed forecasting based on attention convolutional neural network for arterials. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 999–1016. [CrossRef]
37. Zafar, N.; Ul Haq, I. Traffic congestion prediction based on Estimated Time of Arrival. *PloS ONE* **2020**, **15**, e0238200. [CrossRef] [PubMed]
38. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 785–794.
39. Song, Y.; Ying, L. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130. [PubMed]
40. Seber, G.; Lee, A. Linear Regression Analysis; John Wiley & Sons: Hoboken, NJ, USA, 2012
41. Peterson, L. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
42. Zaccone, G.; Karim, M.R.; Menshawy, A. *Deep Learning with TensorFlow*; Packt Publishing Ltd.: Birmingham, UK, 2017.
43. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing Ltd.: Birmingham, UK, 2017.

44. Albawi, S.; Mohammed, T.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference On Engineering And Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
45. George, S.; Santra, A. An improved long short-term memory networks with Takagi-Sugeno fuzzy for traffic speed prediction considering abnormal traffic situation. *Comput. Intell.* **2020**, *36*, 964–993. [CrossRef]
46. Weisstein, E. Hyperbolic Functions. 2003. Available online: https://mathworld.wolfram.com/ (accessed on 1 October 2021).
47. Han, J.; Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In Proceedings of the International Workshop On Artificial Neural Networks, Perth, Western Australia, 27 November–1 December 1995; pp. 195–201.
48. Sunny, M.; Maswood, M.; Alharbi, A. Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model. In Proceedings of the 2020 2nd Novel Intelligent And Leading Emerging Sciences Conference (NILES), Giza, Egypt, 24–26 October 2020; pp. 87–92.
49. Zhang, R.; Bai, L.; Zhang, J.; Tian, L.; Li, R.; Liu, Z. Convolutional LSTM networks for vibration-based defect identification of the composite structure. In Proceedings of the 2020 International Conference On Sensing, Measurement & Data Analytics In The Era Of Artificial Intelligence (ICSMD), Xi'an, China, 15–17 October 2020; pp. 390–395.
50. Hyndman, R.; Koehler, A. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
51. Chai, T.; Draxler, R. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
52. Su, Y.; Zhang, Q. Quality Assessment of Speckle Patterns by Estimating RMSE. In *International Digital Imaging Correlation Society*; Springer: New York, NY, USA, 2017; pp. 71–74.
53. He, F.; Yan, X.; Liu, Y.; Ma, L. A traffic congestion assessment method for urban road networks based on speed performance index. *Procedia Eng.* **2016**, *137*, 425–433. [CrossRef]
54. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
55. Ponomarev, S.; Dur, J.; Wallace, N.; Atkison, T. Evaluation of random projection for malware classification. In Proceedings of the 2013 IEEE Seventh International Conference On Software Security And Reliability Companion, Gaithersburg, MD, USA, 18–20 June 2013; pp. 68–73.