IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 November 2022, accepted 14 December 2022, date of publication 21 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3231448

RESEARCH ARTICLE

Traffic Prediction in Smart Cities Based on Hybrid Feature Space

NOUREEN ZAFAR[1,2], IRFAN UL HAQ[1], HUNIYA SOHAIL[1], JAWAD-UR-REHMAN CHUGHTAI[1], AND MUHAMMAD MUNEEB[3]

[1]Pakistan Institute of Engineering and Applied Sciences, Islamabad 44000, Pakistan
[2]PMAS-AAUR, Rawalpindi 46000, Pakistan
[3]Department of Mathematics, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Noureen Zafar (noureen_zafar@uaar.edu.pk)

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

ABSTRACT In smart cities of the future, data will be generated, integrated, processed and utilized from heterogeneous sources and at varying levels of complexity. For urban traffic planning in smart cities, one of the biggest challenges is traffic congestion prediction and its avoidance. Traffic congestion is a complex phenomenon and it is a manifestation of various contributing factors. In addition to vehicular mobility, properties of road network, weather, holidays and peak hours play a significant role in traffic congestion especially on arterial roads within a city. In this paper, we proposed a hybrid GRU-LSTM based deep learning model and applied it on city-wide novel traffic data integrated from heterogeneous sources. We have devised our indigenous data pipeline that is composed of a set of algorithms dealing with map matching, sparsity handling, outlier removal, zero speed adjustments, Open Street Map (OSM) and segment mapping etc. Extensive experimentations have been carried out to demonstrate the improved performance of the proposed method. The comparative analysis reveals that our methodology yields 95 % accuracy that outperforms other deep neural network models.

INDEX TERMS Intelligent transportation systems, traffic congestion, LSTM, GRU, deep learning, neural networks.

I. INTRODUCTION

Smart Cities promise to improve quality of life by augmenting urban infrastructure with IT based utilities underpinned by Internet of Things and smart services. The sensors and services generate huge volume of data that may be tapped for further utilization in various urban planning activities. One of the biggest challenges in this regard is to aggregate and integrate diverse nature of data from heterogeneous sources. The integration process may contextually conform to compliance standards and results into hybrid feature space. Various applications may result from such hybrid feature spaces embodied with a variety of algorithms. These algorithm-driven applications may generate new data that may also become part of the ecosystem thus giving rise to domain-specific data pipelines and data lakes.

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

In the last two decades, growing population and rapid expansion of metropolitan cities have affected the economic growth, and development. The main restrain for the development was inefficient and inadequate transportation infrastructure which causes serious traffic-related problems. Traffic congestion is one of the main problem of people in modern cities that deteriorates their quality of life in addition to its marked contribution to environmental pollution while hindering urban development. It has caused problems for commuters and lengthened their commuting time [1]. Moreover, wasted fuel, time lost and excessive air pollution due to traffic congestion cause loss of billions of dollars to the economy every year. The 2019 Urban Mobility Report identified 179 billion dollar national congestion cost with approximately 9 billion hours spent on extra travel time, and 3.3 billion gallons of fuel wasted due to traffic congestion for the surveyed four hundred and ninety eight U.S. urban areas in 2017. According to this report, there will be a 32 percent

increase in national congestion cost, 9 percent increase in wasted fuel cost and 14 percent increase in wasted time by 2025 [2].

Machine-learning and Deep learning models can be used in the field of health, neurological states, cardiac monitoring system and Intelligent Transportation System (ITS) for classification of health related data as well as transportation data [3], [4], [5]. A well-planned ITS is desired that could notify the drivers and commuters about the congested road sections in a timely manner so that they can take the alternate routes and plan their journey in a better way to avoid congestion. Lower traffic congestion means lesser wastage of fuel, lower pollution and time savings. Many cities around the world have developed Global Positioning System (GPS)-based services like Google Map, Baidu Map, and deployed sensors including video image processors and inductive-loop detectors used in road networks to observe traffic conditions in real-time. The data collected from these devices and services have opened new avenues for researchers to design and implement intelligent transportation systems to monitor real-time traffic conditions, e.g., predicting city-scale congestion level, traffic volume and speed estimation [6]. This can also help government in decision making related to urban planning, e.g., new land development styles, managing traffic flow and applying new taxes etc. [7]. Besides, this can also facilitate traffic management agencies to execute and optimize their operations (e.g., traffic signal time optimization) in a better way. Recently, there have been numerous studies on real-time road traffic congestion level forecasting using Google Maps API [8], [9].

Traffic related data can be classified into three types including traffic data, road network data and associated data which may be associated with traffic or road network e.g. weather or peak times etc. Traffic data can be further divided into three broad categories including speed data, estimated time of arrival (ETA) data and vehicular count data. Road network data describes road network in the form of a graph and associates spacial characteristics with different roads such as number of lanes, junctions and pavements etc. We use the term traffic associated data for those set of attributes which can either be derived from the traffic data e.g. traffic congestion indices or can be correlated with the traffic data e.g. weather or peak hours. The road network data in the forms of various graphs constructs a GIS map and plays pivotal role to interpret traffic data during the process of data integration as well data visualization. Roads are subdivided into segments on a map. Open Street Maps (OSM) is an open source system that marks various nodes on a road. Two adjacent nodes may serve as delimiters of a segment on the road. Traffic data may be in the form of speed data, count data or ETA type. Hence, it can be associated with specific segments of roads thus giving them a spacial interpretation.

In this paper, we performed aggregation and integration of multiple sources of traffic related data through an elaborated data pipeline and then applied various machine learning and deep learning algorithms for predictive analysis. Starting from classical techniques we culminated at a hybrid GRU-LSTM model. We then optimized traffic congestion prediction results through parameter tuning process.

Following are the outlines of the main contributions of this paper:

- We described our indigenous data integration pipeline that integrated both traffic data and a variety of exogenous data from multiple sources. It eventually produced a hybrid feature space.
- We presented a parallelized and batch processed map matching mechanism of Floating Car Data (FCD) that was based on OSRM's nearest service. Further, a novel mechanism was introduced to handle ambiguities of missing nodes resulting from FCD data.
- This paper also contains algorithms for the preparation of city-wise spatio-temporal data up to the spatial resolution of road segment defined between two adjacent OSM nodes and a temporal resolution of 15 minutes.
- An elaborated Exploratory Data Analysis (EDA) was followed by a comparative performance analysis of different deep and machine learning algorithms. We demonstrated how our proposed technique outperforms the rest.
- We performed a comparative analysis of different deep and machine learning approaches on hybrid data sources and reported our results.

The remainder of the paper includes: Section II that throws light on the related research work whereas section III illustrates the proposed GRU-LSTM based methodology. Section IV depicts the experimental results followed by the discussions. Finally, Section V concludes the present research and also pinpoints future research directions.

II. RELATED WORK

Traffic congestion has become one of the most persistently growing problems across the globe. Predicting current or future traffic conditions on different road segments is a very challenging task due to irregularity of traffic flow patterns and road network complexity. Recent years have witnessed an extensive research in the domain of Intelligent Transportation Systems (ITS) to address the traffic congestion problem. In this regard, several machine learning models e.g. Naive Bayes, SVM, SVR, Logistic Regression, Extra Tree, PCA, FFT, Filter, Wrapper, Embedded, ada-Boost and deep learning models including wavenet, Google deep mind's neural network, autoencoder neural network, LSTM, GRU and LSTM-SPRUM e.t.c., numerous statistical models e.g. bayseian network, ARIMA and STARIMA have been employed to forecast traffic congestion [10], [11], [12], [13], [14], [15].

The limitations of a variety of previously described machine learning, deep learning, and statistical models are listed in Table1.

In these research studies, temporal correlations had been explored to predict traffic congestion by using different models including wave net network, google deep mind's neural network, LSTM, GRU, stacked auto-encoder and multi-step

TABLE 1. Comparison table for related current research.

Methodology	Limitations	Reference
Wavenet net- work, google deep mind's neural network, LSTM , GRU, stacked auto-encoder and multi-step forcasting models	not explored spatial-temporal patterns of the road traffic data.	[8], [16]–[22]
multivariate regression model,temporal graph convolutional network, LSTM, spatial-temporal residual graph attention network, sequence-to-sequence model, GraphCNN-LSTM model, dynamic tensor completion method and attention graph convolutional	not consider the impact of heterogeneous data sources	[23]–[36]
bidirectional LSTM	ignoring weather, ETA and special events data sources	[37]
average fusion, KNN fusion and weighted fusion	data driven fusion such as hyper feature space ignored	[38]
Queueing Theory for outlier detection and segmentation, SVRGSA	not considered integration of multiple data sources	[39]–[46]
hybrid LSTM and ResNet model	errors in sparse spatial areas lacks work on different nature of data sources	[51]
CNN, BILSTM	external factors like weather, holiday and emergency traffic data	[52]
GRU	did not consider other types of data sources.	[53]
stack LSTM and transfer-learning	not consider Heterogeneous data sources	[54]
hybrid CNN-LSTM	other exogenous data sources	[55]
Vehicular Ad hoc Networks	not discuss probabilistic traffic prediction techniques and hybrid classifiers	[56]
bagging, boosting, stacking, and random forest ensemble models	not pay attention to deep learning and missing data imputation methods.	[61]
Deep Belief Network (DBN)	works only ETA not work on heterogeneous data sources	[62]
LightGBM-GRU	for enhancing features list heterogeneous data sources are ignored	[63]
CNN	not work on hybrid deep learning models and exogenous data sources	[64]
Support Vector Machine (SVM) and Multinomial Naïve Bayes (MNB)	not work on hybrid deep learning models and exogenous data sources	[65]
LSTM + CNN + ATTENTION	not works on heterogeneous data sources	[66]

forcasting models [8], [16], [17], [18], [19], [20], [21], [22]. However, the authors didn't explored spatial-temporal patterns of the road traffic data.

Spatial-temporal patterns [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36] were exploited to take into account the chronological deviations for traffic congestion using multivariate regression model,temporal graph convolutional network, LSTM, spatial-temporal residual graph attention network, sequence-to-sequence model, Graph -CNN-LSTM model, dynamic tensor completion method and attention graph convolutional. The limitation to these studies is that the authors did not take into consideration the impact of heterogeneous data sources on predicting traffic congestion on road network. The authors in [37] introduced the path based deep learning framework for capturing spatial-temporal features as well as effective speed prediction. In these studies, the authors used bidirectional LSTM for the prediction of speed considering spatial-temporal factors. However, they only focused on single type of data source (speed data) and ignored other data sources such as weather, ETA and special events etc which directly or indirectly influence the prediction of correct speed. In [38], the authors proposed multiple fuse predictor strategies such as average fusion, KNN fusion and weighted fusion in order to enhance overall performance of their individual model. However, they were unable to fuse feature driven hybrid feature space.

The authors of [39], [40], [41], [42], [43], [44], [45], and [46] used Queueing Theory for outlier detection and segmentation, SVRGSA for selection of appropriate hyper-parameters and hybrid model for the prediction of traffic patterns. They further applied different machine learning techniques to predict problems on the segmented data set posed by traffic congestion. They also experimented on numerous speed data sets and compared different machine learning and statistical models for congestion prediction. However, they did not attempted to predict ETA. Furthermore, they did not consider integration of multiple data sources like weather, special events and, road conditions. Additionally, they had also not taken into account long-term traffic patterns and road conditions like surface-turns and lane features.

The authors in [47], [48], [49], and [50] analyzed the impact of various factors including road intersections, number of market places, and rickshaw free roads on the traffic intensity. However, didn't deal with integrated data sources.

Reference [51] suggested a hybrid integrated–DL model. This model discussed both spatial-temporal dependencies in predicting city wide spatial-temporal traffic flow volume. The authors also presented a hybrid LSTM and ResNet model that deals the spatial-temporal effects on a given volume of traffic on the road. The demerits of the said proposed model were large errors in test data in sparse spatial areas and non-peak hours. Moreover, the research study also lacked work on different nature of data sources like count and ETA e.t.c.

The authors [52] predicted the traffic congestion by extracting spatial and temporal features from CNN and BIL-STM, respectively. However, they did not address the external factors like weather, holiday and emergency traffic data. The authors in [53] used the GRU model on speed data source and examined the weather impact on speed but did not consider road network and other types of data sources like ETA and count. The authors of [54] used stack LSTM and transfer-learning in order to tackle the problems of missing data, data insufficiency, and mitigated model over fitting. However, the authors didn't consider impact of different input attributes that were gathered from exogenous data sources, different traffic modes and traffic types(e.g. ETA, FCD and count). Heterogeneous data sources might be useful while applying transfer learning on the specific area. The authors of [55] used hybrid CNN-LSTM model encompassing predictions about both city wide traffic congestion data and its corresponding City wide pollution data source. They achieved 92.3 percent model accuracy. However, other exogenous data sources and other modes of traffic were not dealt in this study. The authors of [56] worked on Vehicular Ad hoc Networks for congestion detection and control line strategies. They did not discuss probabilistic traffic prediction techniques and hybrid classifiers.

Reference [58] worked on lane detection algorithm using Hough transform and vehicle detection using SSD at the beginning steps. After that, the violation-detection algorithm was used to identify traffic violations. However, the authors only focused on the data received from the camera and not worked on heterogeneous data sources for the prediction of traffic congestion on the road network.

The present study considered multiple heterogeneous data sources generating features such as ETA, speed, weather, Special events and road segments etc. We also worked on hybrid feature space and explored the spatial-temporal patterns using hybrid deep learning models i.e. GRU-LSTM and LSTM-GRU.

III. PROPOSED RESEARCH METHODOLOGY BASED ON HYBRID GRU-LSTM MODEL

Smart city speaks of various kinds of IoT devices, services and heterogeneous data sources. Following the notion of smart cities, we provided a mechanism to integrate heterogeneous data sources into a hybrid feature space for forecasting traffic patterns. Various features in the hybrid feature space played their contributing roles in analyzing and predicting traffic congestion patterns. Hybrid feature space included speed, estimated time of arrival (ETA), weather data and road network data. FCD was obtained from a tracker's company in raw and then it was synthesized. ETA data was gathered from Google Direction API. Weather data was extracted from Yahoo API and Dark Sky API whereas OSM's road information data was fetched through Turbo overpass API. Final integrated features were End-node, Start-node, Way-id, DateTime, Peak-Hour, Special-Condition, Weather, Speed, ETA and congestion level (Smooth, Congested, Higly Congested). Multiple data sources were integrated on the level of road segments as defined by OSM.

The flow of data integration and proposed approach is illustrated in Figure 1. It is necessary to process diverse nature of data into identical schema so that every data is uniformly mapped on the OSM. For this purpose, it should go through map matching process. Hence, the map matching was the very first step in our pipeline. The design and flow of complete Data Integration Pipeline and predictive model is shown in Figure 1. Various activities of the proposed approach are listed below:

1) Collection of requisite data from various Heterogeneous Data Sources is described in Section III-A.
2) Preprocessing included Map Matching algorithm and zero speed correction in Section III-B.
3) Time Series Aggregation and Integrating all Data Sources into a Hybrid Feature Space have been defined in Section III-C.
4) Integrated Data Analysis is described in Section IV-A
5) Predictive Modeling explained in Section III-D.
6) Parameter optimization and performance evaluation are discussed in Section IV.

A. DATA COLLECTION FROM HETEROGENEOUS DATA SOURCES

The heterogeneous data sources that contributed in the hybrid feature space included Floating Car Data (FCD), Estimated Time of Arrival (ETA) data from Google maps API, road network data from Open Street Maps (OSM), weather data from Yahoo Dark Sky API, calendar data from Date-Time API as well as special conditions and peak hours data. The features list of heterogeneous data sources is defined in Table 2.

Special conditions data refer to traffic schedules due to major events whereas peak hours are calculated from historical data based on Congestion Index (CI). CI is a feature derived based on average traffic flow on a specific road segment. Figure 1 depicts multiple data sources and various data pre-processing activities performed on them. Now we examine each data source one by one in detail.

Floating Car Data (FCD) for the complete month of September 2020 comprised of 2895 unique tracker IDs was fetched from a local tracker company. GPS Chipset(U-blox

TABLE 2. Heterogeneous data sources features list.

Data Sources	Features
FCD	Date Time , Unit ID, Latitude, Longitude, Speed, Reason, Direction, Altitude, Location.
Google	Date, MinSpeed, Source_Latitude_Longitude, Destination_Latitude_Longitude, ETA.
Calendar	Date Time, SystemTime, Day.
Weather	Date Time, Type, Name , Precipitation_Intensity_Max, Precipitation_Type.
OSM	MaxSpeed, Summary, StartNode, EndNode, SegmentId , WayId .

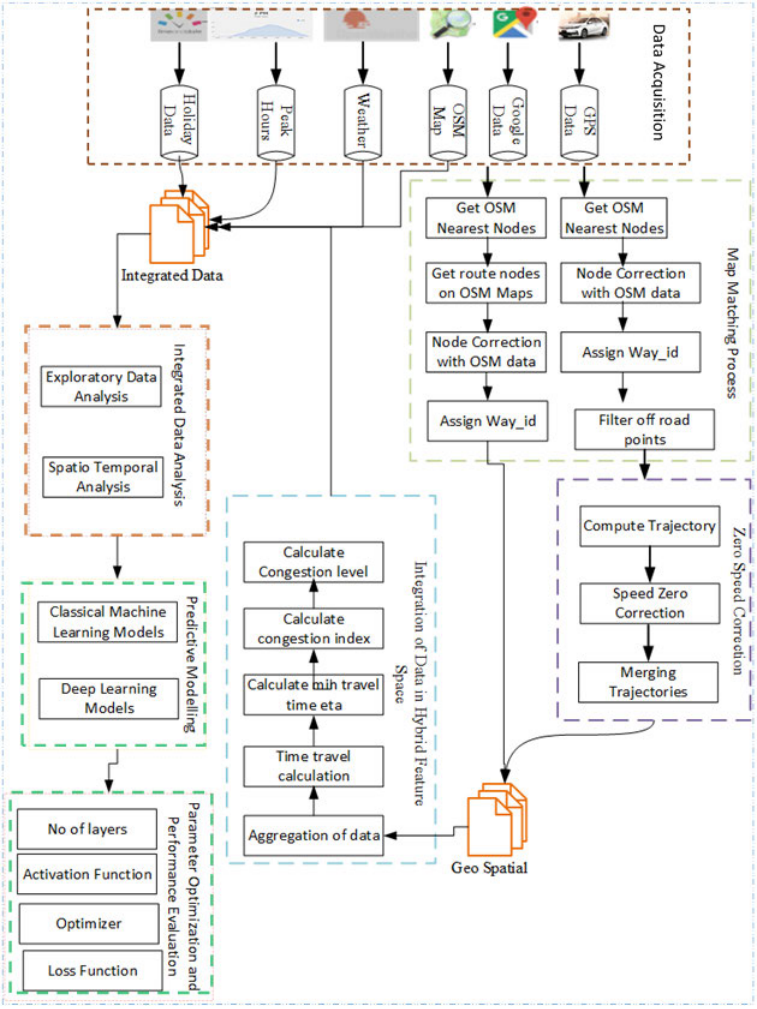

FIGURE 1. Proposed methodology for traffic congestion prediction with hybrid- deep learning models using heterogeneous data sources.

EVA-M8M) and GSM Modem (Quectel M95) sensors were used in trackers units. Trackers produced regular signals of various alerts which constituted FCD. The feature-set of FCD included unit ID, DateTime, latitude, longitude, speed, reason, direction, altitude, address and location. Following issues have been identified in FCD while the data cleaning process:

- GPS data has an inherent spatial error which causes off-road mapping of cars.
- There are garbage records in latitude and longitude fields.
- Distances and time stamps are changed even if speed values generated by trackers are zero.
- There is redundancy in records.
- There is incomplete information that causes problem to identify the exact position of the car.
- The resolution of individual triggering time is very high.
- There are missing Values in the map segment data due to unavailability of FCD records on some segments of roads.
- There is a huge number of zero values of speed parameter due to parked cars. Zero values due to parked cars need to be distinguished from the zero values due to Highly Congested roads.

We resolved the first problem by Map Matching Algorithms. The second issue was resolved by discarding the garbage

TABLE 3. Total size of heterogeneous data sources.

Label Classes	Size
smooth[0]	3,944,218
congested[1]	1,361,499
highly congested[2]	1,981,125

values. The third problem was tackled by building the trajectory of each user separately. The fourth problem was addressed by discarding the repeating records. The fifth problem was resolved by adding the direction. By using multiple trajectories of different trackers, sixth issue was resolved. To handle missing values in segments, values from the neighboring segments lying on the same trajectory were utilized. We tackled the last problem by using the reason feature available in FCD. The reason feature contained events such as ignition_on, ignition_off, power_on, power_off, timer, and turn etc. Figure 2 depicts the map of Islamabad and one month's processed data plotted on it. Red colour shows the density of the data. Data has been collected from Kashmir Highway, Constitution Avenue, Margalla Road, Jinnah Avenue, Faisal Avenue, 7th Avenue, 9th Avenue, Ahmed Faraz Road, Service Road E (F-10 and G-10), Service Road W (F-11 and G-11), IJP Road, Ibn-e-Sina Road,, Route for Metro Bus Service Islamabad, and main roads of sectors F-7, F-10, F-8, F-11, G-8, G-9, G-10, G-11. FCD covers all of these roads as well as the remaining roads.

We fetched data from Google Maps API in real-time and and compiled our data by consigning start and endpoints to the requisite Google Maps API [9]. We marked more than 500 points on the Islamabad map which covered all important roads of Islamabad. Google data consisted of Date, System Time, Day, Source Longitude, Destination Longitude, Destination Latitude, Source Latitude, and ETA.

OSM data was gathered from Turbo Overpass API. The OSM data was accessed by specifying a bounding box in terms of latitudes and longitudes. OSM provided the geometry and important features of road network. The main features extracted from OSM were way_id, end node, start node, Max Speed, Min Speed, Max length and Motorway highway types.

OSM road network comprises of a set of large number of nodes. The distance between any two adjacent node can be treated as a segment which provide an excellent mean for narrowing spatial resolutions. Though the distance between any two adjacent nodes is not always equal. OSM provides node_id and way_id but does not have a concept of segment. We have created a customized map structure called City Map Structure(CMS) that defines a road network in a city as a set of road segments. In the present study, the concept of segment was used to resolve issues related to map matching. CMS is a table having fields of segment_id, start node, end node, way_id and Road network.

Since Behavior and pattern of traffic was strongly dependent on holiday data on a particular road network at a specific moment therefore, the holiday data was extracted from date and time.com website. Calendar Data consisted of Date, time, type and name.

FIGURE 2. Islamabad road map.

Algorithm 1 Map Matching Process

```
Read coordinate point [lat, lon] from data
Initialize nodes-segment with empty start-point-list and end-point-list
for t=1 to n do
    Initialize way-id to zero
    nodes-segment[“start-point”,”end-point”]:=get_nearest_segment(lat,lon)
    if nodes-segment [“start-point”] is zero then
        Check end-point value and update the start-point value with previous node value
    else
        nodes-segment [“start-point”] and nodes-segment [“end-point”] represents to same road points
    Update way-id
```

Weather data ws fetched from Dark Sky API and Yahoo on the basis of latitude and longitude of the location.

Table 3 data set comprises of 7,286,842 records from September 2020. In the data set, there were 3,944,218 smooth conditions, 1,361,499 congested conditions, and 1,981,125 highly congested conditions.

B. DATA PRE-PROCESSING

Data Pre-processing is primarily a technique that works on raw data. It is used to organize and clean data to make it compatible with training machine learning models. In transportation domain, map matching, sparsity handling, outlier removal, zero speed adjustments and Open Street Map(OSM) segment mapping, etc were addressed in data pre-processing.

GPS data usually has an error of up to several meters. That error is resolved through map matching techniques in order to accurately map an FCD data point on the exact road. We developed a novel map matching mechanism based on OSRM.

We used Open Source Routing Machine (OSRM) which supported OSM for the purpose of map matching. A speed or ETA data point was mapped on a road segment. We defined

Algorithm 2 Map Matching Process

```
Split file into m units
Assign units to r processors
 range = 10

for n records in each unit do group the data by unit_id sorted by date time to form trajectories
      Read coordinate point [lat, lon, bearingangle] from data
    nodes-segment[“startnode”,”endnode”]:
=get_nearest_segment(lat,lon, bearingangle,range)

    if nodes-segment [“startnode”] is zero then retrieve way-id list of endnode from CMS
wayid= pick the matching way-id as per trajectory
nodelist = geometry of way-id from CMS
startnode = previous node of endnode in nodelist
      nodes-segment [“endnode”] is zero retrieve way-id list of startnode from CMS

wayid= pick correct way-id as per trajectory
nodelist = geometry of way-id from CMS
endnode = next node of startnode in nodelist

    else
        Assign way-id to coordinate point =0
```

Algorithm 3 Zero Speed Correction Based on Events

```
threshold = 2500
while next_record != NULL do
    Read records from data
    Initialize speed= data[“avgspeed”]
    Initialize elapsed time1= data[“elapsed time1”]
    if speed > max_speed then
        speed = max_speed

        if speed = = zero then
            if reason ==“ignition OFF” and reason == “Power_OFF” then
                delete record
                if elapsed time > threshold then
                    delete record
                     if reason ==“ignition ON”and reason == “Power_ON” then
                        delete record

                        if speed > zero and node.highway = “Residential” then
                            delete record

            keep record
        =0
```

road segments as a polyline on the map between two adjacent OSM nodes. Therefore, our map matching algorithms did not require a precise point rather a precise road segment on the map. Hence a speed or ETA point was associated with a specific road segment. We noticed that the online API of OSRM had a very low response time so we setup an offline OSRM server. The response time improved but still it was very slow and there were memory leakage issues when huge volumes of data was being processed. Therefore, we created a multi-threaded and batch processing script that drastically reduced the processing time. The parallelized mechanism had already been documented in [57]. Map matching procedure is represented by the following Algorithm 2. The data comprised of specific coordinate points possessing longitude, latitude and the bearing angle of the specific located geographical position. These coordinate points were sent to the nearest API of OSRM server to obtain required pair of nodes (marking the delimiters of the specific road segment) depicting the location of the driving vehicle. Consequently, the order of nodes on OSM maps differentiated between the incoming and outgoing traffic bearing roads. Occasionally, OSRM nearest API reached zero value at start or end node. This was due to multiple available options at the nearest end/start node owing to junctions on roads. For this purpose, an algorithm for the correction of zero values was written.

Startnode and endnode contain zero values. To correct the zero value pertaining to start or end node returned by OSRM, the trip trajectory containing a sequence of OSM nodes was obtained by retrieving the way_id of the trailing point. A road can have a sequence of way_ids due to its changing attributes e.g. number of lanes, condition of pavements etc. Similarly a node can also be associated with more than one way_ids if it is on a junction.

Identifying the right way_id of the node along the progressing trajectory is the trick to fix the zero values. Zero value associated with the start node was replaced with the value of the node previous to the end node located on trajectory segment described by the geometry of the retrieved way_id. To fix the zero value of the end node, the same algorithm was applied and zero value was replaced by the value of the node next to the start node on the identified trajectory. A road on OSM maps was divided into several different types of sections, and each section was identified by the way_id of the road. OSM did not define the concept of road segments. As part of this research, we created road segments as the segments between two adjacent OSM nodes on a way_id of the OSM. We maintained the whole list of road segments of a city (Islamabad in this case) and term it as the City Map Structure (CMS).

Algorithm 3 addresses zero speed issues based on events. Vehicles which have trackers installed on them spend a significant amount of time staying idle i.e. in the parking state. Therefore, almost 40 percent of the records contained zero speed values. The zero speed values can not be discarded right away as some of them may represent high congestion.

Algorithm 4 Time-Series Aggregation and Integration

for each segment in CMS **do**
  read data

  for each quarter of hour for each day **do**
    if day is "Sunday(Sun)" or "Saturday(Sat)" or "Holiday" **then**
      data["Weekend"]=1

    else
      data["Weekend"]=0
      weighted_avg_speed $= \frac{1}{W_1+W_2}(W_1 * \sum_{i=1}^{n}(x_i/n) + W_2 * \sum_{j=1}^{m}(x_j/m))$
      data["weather_condition"]=get weather_condition of the current day =0

There is a parking event in the tracker for automatic transmission vehicles but for the manual transmission, the parking state needs to be determined by examining the sequence of ignition-off/power-off and ignition-on/power-on events. Even in that case it is a bit tricky to identify whether the vehicle is momentarily stuck in the congestion or parked for a longer period. The straightforward technique is to check the sequence of events between ignition-off and ignition-on events of the vehicle. When the car is parked, it still generates timer-on event but with huge gaps e.g. 2500 seconds or so. The exact duration of the gap depends upon the specific tracker. If that gap exceeds a threshold then the car is considered parked and the record is discarded. If the car is not parked then the gap is much shorter.

C. TIME SERIES AGGREGATION AND INTEGRATING ALL DATA SOURCES INTO A HYBRID FEATURE SPACE

Algorithm 4 addresses integration of heterogeneous data sources based on time-series. A single road segments captures different coordinate-points in different time windows. Our data fusion technique grouped speed at different spatial points lying on the same segment and computed their average in specific duration of time. In this method, we switched from latitude and longitude to more meaningful map attributes that are OSM nodes.

Once the data had been transformed on OSM standard nodes that ensured the uni-schema of each independent geo-spatial data set with temporal facts, the traffic information of each transformed data was aggregated into 15 minutes separately, between start-node and end-node of a particular way-ID. The aggregated data for both FCD and Google Maps was then integrated on the basis of the start-node and end-node between 15 minutes time-windows followed by merging of resultant integrated data with date based holiday data. The purpose of the same was to cope with the traffic congestion effects during both working hours and those due to off days. The integrated data was further aligned with various road attributes including ways_id to obtain the adaptive traffic patterns with respect to both time and space. Finally, we were able to get the GPS trackers parameters along with maximum speed, surface and number of lanes on the particular road. Moreover, for combating various environmental effects, we also merged weather data with integrated data e.t.c. as depicted in Table 6.

TABLE 4. Hybrid feature space text.

Features	The Value of Features
Day	1 to 7
Startnode	numeric
Endnode	numeric
aggminutes	0-15, 16-30, 31-45, 46-60
Weather	mostly sunny ,Sunny,rainy,cloudy
SpecialCondition	Yes, no
Time	Rushhour, non_Rush_hour
Holiday	1, 0
DateTime	Varchar

D. MODEL SELECTION

The spatial-temporal characteristics of the data led to the selection of algorithms specialized in spatial and temporal data. Our problem is inherently time series and multi-class problem. In the first instance, we applied classical classification techniques like Random forest, Support Vector Machine and XG Boost e.t.c. that failed to yield satisfactory results. In the second attempt, we tried to solve our problem via deep learning techniques and their ensembles such as LSTM, MLP, GRU, GRU-LSTM and LSTM-GRU. The latter yielded remarkable accuracy.

Stacked LSTM architecture consisted of hidden bilayers with each layer further comprising of 64 hidden units. Here, Tanh was used as an activation function in both hidden layers. Dropout layer with ratio 0.2 was used to regularize not only the network between the hidden and Dense (output) layer but also between two hidden layers. Dense layer with softmax activation function and 3 units activation function was applied in the output layer followed by holdout cross validation that divided data set into test and training sets. A batch learning approach was used to train the model on the training data set followed by checking of generalization of model on test data set.

Recurrent neural networks applied in this paper control the flow of the information. Gated Recurrent Network (GRU) is similar to LSTM but it uses two gates i.e. reset gate and update gate. The update gate decides whether previous information should be used or not. In other words, the update gate determines the previous information amount (prior time steps) needed to be passed along the next state whereas the reset gate decides the past information needed to be neglected.

Different combinations of LSTM and GRU were applied. First of all, we have discussed LSTM-GRU architecture and then applied the GRU-LSTM model. GRU-LSTM performed outstanding results as compared to LSTM, GRU, and LSTM-GRU.

TABLE 5. Testbed implementation details.

Features	Details
Software	Linux 64 bit operating system python version 3.7.16 simulation platform Keras(2.3.1) based on tensorflow(2.1.0)
Hardware	NVIDIA GeForce GTX 1070 Ti-equipped machine.

In LSTM-GRU, First of all input features were passed to two LSTM layers for extraction of temporal features and then, two layers of LSTM were incorporated and connected to the output layer. Output layer predicted congestion level i.e smooth, congested, and highly congested. In each layer, we used 128 units of neurons. The tanh activation function was applied in input to intermediate layers whereas softmax activation function was used in the output layer. Adam was applied as an optimizer whereas categorical cross-entropy was used as a loss function. Validation accuracy was 93.7 percent.

1) EXPERIMENTAL SETUP

See Table 5.

2) LONG SHORT TERM MEMORY

In this experiment, the proposed stacked LSTM architecture consisted of two hidden layers each containing 64 hidden units. Tanh is used as an activation function in both hidden layers. Dropout layer with ratio 0.2 was used to regularize the network between the two hidden layers and between hidden and Dense (output) layer. Dense layer with 3 units and softmax activation function was used in the output layer. We employed holdout cross validation to split data set into training and test sets. We firstly trained the model on the training data set by a batch learning approach and then checked generalization of model on test data set. The proposed LSTM model was applied to the data collected from Google and FCD which comprised 7,343,362 records of September 2020. The traffic condition data were collected every fifteen minutes covering 1649 segments of arterial roads in Islamabad, Pakistan. There were three categories of traffic conditions, i.e., smooth condition, congested condition and highly congested condition. In the data set, there were 4,218,750 smooth conditions, 1,480,417 congested conditions and 2,634,256 highly congested conditions. To evaluate the performance of the proposed deep architecture, we adopted accuracy, precision and recall, as a performance measure.

3) GATED RECURRENT UNIT (GRU)

Recurrent neural networks applied in this paper control the flow of the information. Gated Recurrent Network (GRU) is similar to LSTM but it uses two gates including update gate and reset gate. The update gate is responsible for deciding whether previous information should be used or not. In other words, the update gate is responsible for determining the amount of previous information (prior time steps) that needs to be passed along the next state whereas the reset gate is used from the model to decide how much of the past information is needed to neglect.

4) LSTM- GRU MODEL

First of all input features were passed to two LSTM layers for extraction of temporal features and then, two layers of LSTM were incorporated and connected to the output layer. Output layer predicted congestion level i.e smooth, congested, and highly congested. In each layer, we used 128 units in the layer, and tanh activation function was used in input to intermediate layers. In the output layer, softmax was used as an activation function. Adam was used as an optimizer and categorical cross-entropy was used as a loss function. Validation accuracy was 93.7 percent.

5) PROPOSED GRU-LSTM MODEL

We proposed hybrid GRU-LSTM model because GRU deals with the vanishing gradient problem. It also works on less memory by using less training parameters as compared to LSTM. LSTM has a capability to learn long term dependencies in addition to remembering long period of time using memory unit. In our proposed model, we used most promising time-series analyzers i.e. GRU and LSTM. GRU was applied at front layer. The output of the GRU was subsequently passed to LSTM. GRU has two gates e.g. update gate (ut_g) and reset gate (rt_g). The mathematical formula of update gate (ut_g) is explained in Equation 1:

$$ut_g = \sigma(we_u[x_t] + we_u[hd_{t-1}]) \tag{1}$$

Firstly, x_t passed as input to the first layer of GRU (ut_g) where x_t and $hidden_{t-1}$ were got multiplied to weight and then were added together. Then sigmoid activation function was used to convert results between 0 and 1. Update gate (ut_g) was aimed at deciding how much past information was passed to the future timestamp. Then this information was forwarded to reset gate (rt_g). The calculation of reset gate (rt_g) is expressed in Equation 2:

$$rt_g = \sigma(we_r[x_t] + we_r[hd_{t-1}]) \tag{2}$$

In reset gate(rt_g), calculation x_t and $hidden_{t-1}$ were multiplied by its own weight and were then sum up together followed by use of sigmoid activation function. rt_g decided which information needed to be stayed and which information be forgotten. It also stored stayed information by using the following Equation 3:

$$\sim hd_t = tanh(we_{xt}[x_t] + rt_g \circ we_{xt}[hd_{t-1}]) \tag{3}$$

x_t was multiplied by its weight we_{xt}. The element wise product was performed to the previous output hd_{t-1} an reset gate rt_g. Both results were added together and passed to tanh function. The unit computed the hd_t using following Equation 4:

$$hd_t = ut_g \circ hd_{t-1} + (1 - ut_g) \circ \sim hd_t \tag{4}$$

if u_g is near to 0, it means a big part of information was lost because current information was found irrelevant for the

prediction of traffic congestion. At the same moment, since u_g will be near to 0 at current time step, $1 - u_g$ will be near by 1 and most of the past information will stay in memory. The output of GRU(h_t) was then passed to the layer of LSTM as input. LSTM consists of three gates e.g. input gate(i_g), output gate (o_g) and forget gate (f_g). The behaviour of calculation of i_g is expressed in Equation 5:

$$i_g = \sigma(we_i[hd_{t-1}, hd_t] + b_i) \tag{5}$$

hd_t after passed through the network unit got multiplied by it's own weight(we_i). Like wise, hd_{t-1} was multiplied by it's own weight(we_i) followed by its addition to the bias(b_i). A hd_{t-1} had the details of previous units t-1. In order to produce the result in 0 and 1, sigmoid activation function was used. The input gate selected information that needed to be eliminated from the given cell state. In the second step forget gate(f_g) was used in sorting information needed to be stored in the cell state. Finally, output gate(o_g) decided value to be updated using these two mathematical Equations 6,7:

$$f_g = \sigma(w_f[hd_{t-1}, hd_i] + b_f) \tag{6}$$
$$o_g = \sigma(w_o[hd_{t-1}, hd_t] + b_o) \tag{7}$$

Furthermore, $\sim_{Ct}$ being vector of candidate values was generated through tanh layer. $\sim_{Ct}$ is calculated by using the Equation 8:

$$\sim_{Ct} = tanh(w_c[hd_{t-1}, hd_t] + b_c) \tag{8}$$

Furthermore, the previous state of the C_t is updated by using following Equation 9:

$$C_t = f_t * C_{t-1} + i_t * \sim_{Ct} \tag{9}$$

$\sim_{Ct}$ and C_t differentiated between desirable information to be kept in memory and irrelevant information that needed to be forgotten. This was followed by the attachment to the dense layer. The dense layer contained tanh as an activation function that was predicted road traffic congestion at specific location in a given time frame. The tanh function was employed to transform the values lying between -1 to 1. It was then-multiplied by sigmoid layer ouput in order to acquire the desired output by using the Equation 10:

$$C_t = o_g * tanh(c^t) \tag{10}$$

We used adam as an optimizer and Categorical-Cross-entropy was employed as a loss function in present case of classification problem. Figure 3 shows the GRU- LSTM architecture in order to extract spatial-temporal features. Current Model was trained and validated on 3954765 and 1689234 samples, respectively. First of all input features were passed to two GRU layers for extraction of temporal features and then, two layers of LSTM were incorporated and connected to the output layer. Output layer predicted congestion level i.e smooth, congested, and highly congested. In each layer, we used 128 units, and tanh activation function was used in input to intermediate layers. In the output

TABLE 6. Proposed GRU-LSTM approach sumary report.

Layer(type)	Output Shape	parameter#
gru_1(GRU)	(None, 28, 128)	49920
gru_2(GRU)	(None, 28, 128)	98688
lstm_1(LSTM)	(None, 28, 128)	131584
lstm_2(LSTM)	(None,128)	131584
dense_1(Dense)	(None,3)	387

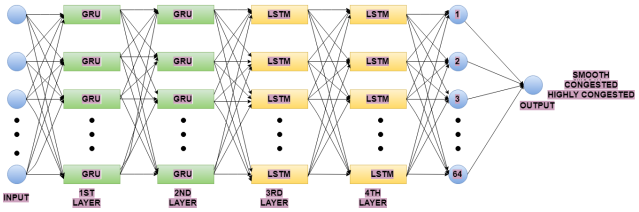

FIGURE 3. Architecture diagram of proposed GRU-LSTM model.

layer, softmax served as an activation function, Adam as an optimizer and categorical cross-entropy was used as a loss function. Validation accuracy is 95 percent. The proposed architecture is depicted in Table 6. The following step by step process explains the functionality of input, output and hidden layers:

Step 1: Cleaned the google and FCD data, and then read t_L and t_O.

Step 2: we defined congestion index(CI) using Equation 11:

$$CI = (t_L - t_O)/t_O \tag{11}$$

where t_L denotes current time required for the road segment and t_O represents the least time required for the road segment. CI is a derived attribute that is used to define thresholds of various categories of congestion. According to the congestion index CI, the traffic situation could be divided into A: smooth B: congestion C: highly congested as shown in Table 6.

Step 3: So we modeled the speed and traffic flow in time. For example: we can get the t+1 according to the in-front-of-several-moments t, t - 1, t - 2, t - 3, t - 4.

Step 4: Got the input layer x = [t+1].

Step 5: The model had 4 layers of hidden layers which used the tanh activation function, with 128 neurons.

Step 6: The output layer used the softmax activation function, and categorical-cross entropy as loss function.

We discussed the integrated data analysis in section 4 whereas a discussion on parameter optimization and performance evaluation was distributed among sections 4 and 5.

IV. RESULTS AND DISCUSSIONS

The limitations of a variety of previously described machine learning, deep learning, and statistical models are listed in Table1. We have described our research work results and their discussions below.

TABLE 7. Statistical measures of congestion index of heterogeneous data sources.

Classes	Mean	Median	Std	Min	Max
Smooth	0.01	0.0	0.035	0.0	0.20
Congested	0.495	0.499	0.179	0.20	0.92
Highly Congested	1.39	1	0.67	0.92	4

TABLE 8. Congestion state level of proposed approach target classes.

CI	Traffic State Level
(0 , 0.2)	Smooth
(0.2, 0.92)	Congested
(0.92,4)	Highly Congested

A. EXPLORATORY DATA ANALYSIS (EDA)

During this study, the traffic patterns on all week days including weekends were statistically analyzed. The means, medians and standard deviations of all three output labels of data are mentioned in the Table 7. Congestion Index (CI) was used to label these classes. CI, a derived attribute provided a normalized expected time of arrival. Thus it prevented from having extreme values. Algorithm 4 explained the procedure to calculate CI for all data points.

Algorithm 5 Segmentation Normalization of Congestion Index (CI)

```
Segmented distribution of records
Label recorded of each segment

for each Road Segment R_s i do
    t_m = min(ETA in R_s i)

    while i < len(Road Segment R_s i) do
        Computed CI_R si = (1/t_m) * t_i − t_m
        applied required label on the t_i containing records
    =0
```

The integrated data set was categorized into three classes namely smooth, congested and Highly Congested as depicted in Table 9.

Smooth class ranged between 0 to 0.20, nearest to zero and had almost perfect mean, median and std. However, both highly Congested and Congested classes ranging between 0.92 to 4 and 0.20 to 0.92 respectively had imperfect mean and median.

Figure 5 depicts the congestion index vs time in hour on various weekdays i.e. Different colour lines have been used to show same on Monday to Friday. The Figure 5 evidenced that a major deviation exists in the value of congestion index over different time (in hours) of the same day e.g. the congestion index was 0.69 at 8:00-9:00 am (a morning rush hour) and was 0.90 between 11:00-12:00 pm, higher than the average of the morning rush hours. During rush hours (4:00-5:00 pm), the congestion index raised to 0.95. However,

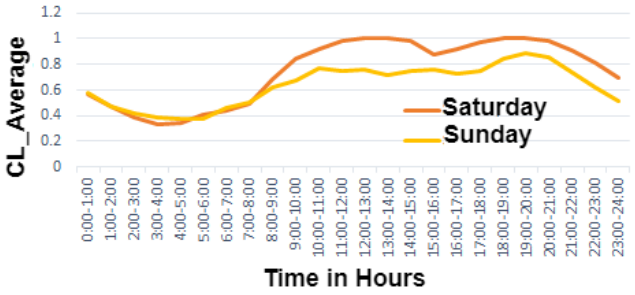

FIGURE 4. Heterogeneous data sources congestion index variation on weekends.

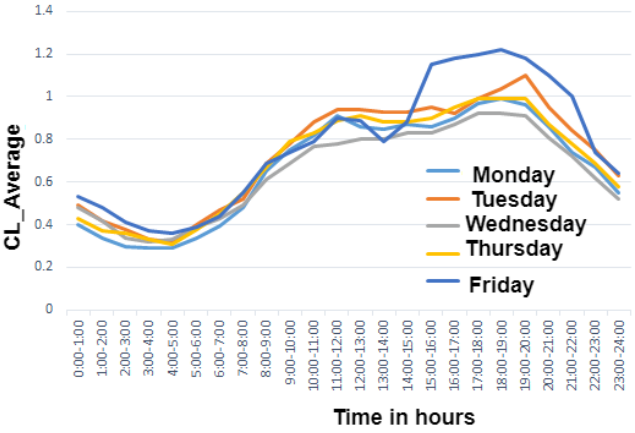

FIGURE 5. Heterogeneous data sources congestion index variation on week days.

the highest congestion index valuing to 1.1 was observed at 7:00 – 8:00 pm (the evening rush hour). It advocated the importance of congestion index behaviour at different time slots in our present model development. Keeping in view its importance, average of the time slot specific congestion index was used as a predictor in prediction of congestion. Time slots 8:00-9:00 am, 11:00-12:00 pm, 4:00-5:00 pm, and 7:00-8:00 pm thus showed peak traffic congestion on road. Friday's congestion index was however quite different from other weekdays due to official breaks in offices to offer Friday prayers, long weekend and half school timings. On Friday, the congestion index was initially estimated to be 0.5 at 8:00-9:00 am (the morning rush hou) that subsequently raised to 0.8 from (12:00-1:00 pm) owing to the offering of Friday prayer during 12:00 to 1:00 pm. Again the highest congestion index valuing to 1.2 was observed around 15:00 – 20:00 pm (the evening rush hours) which was the highest among the average of all morning and evening hours of all weekdays.

Figure 4 depicts congestion index variation on weekends i.e. Saturday and Sunday. On Saturday, congestion index remained uniform during 10:00 am to 03:00 pm valuing to 1. The value of CI decreased during time slot 3:00 to 5:00 pm and again achieved value of 1 between 5:00 pm to 8:00 pm being recreational timing during weekends. A different trend of CI was observed on various time slots of Sunday e.g.

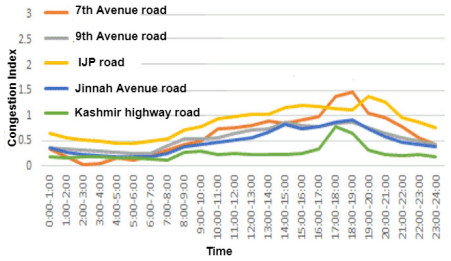

FIGURE 6. Heterogeneous data sources spatial impact on road traffic.

0.8 CI was estimated during 10 am to 11 am. Similarly CI was 0.85 during 06:00 pm to 08:00 pm. Hence, the data showed that traffic congestion trend was quite different for both weekend and weekdays time slots.

Figure 7 shows the spatial impact of traffic on different rushy roads. e.g. Seventh Avenue remained congested from 10:00 am to 05:00 pm and blocked from 05:00 pm to 07:00 pm. The 9th Ave road showed congestion from 8:00 am to 9:00 am, 2:00 pm to 3:00 pm, and 6:00 pm to 7:00 pm. IJP road being a rushy road due to load of logistic trucks was highly loaded from 10:00 am to 10:00 pm. Jinnah Ave showed congestion from 2:00 pm to 3:00 pm and 6:00 pm to 7:00 pm as it ran through the business hub. Kashmir highway remained congested for atleast two hours between 5:00 pm to 7:00 pm due to official off timings. It also showed how spatial trends affect traffic congestion. After detailed analysis of traffic data set, it is revealed that traffic congestion is directly affected by spatial as well as temporal aspects. Therefore both time features and spatial features can be used to predict true traffic congestion phenomenon.

B. FEATURE SELECTION TECHNIQUES

In order to reduce computation in the data obtained from heterogeneous data sources, we selected the most relevant and significant feature set by using Heat Map techniques.

1) CORRELATION MATRIX WITH HEATMAP

Correlation matrix technique was used for advanced and detailed analysis of features set. A correlation matrix consists of table showing correlation coefficients between two features with the Correlation value lying between positive 1 to negative 1. Positive correlation means input feature is more relevant to the target and vice versa. The visual effects were further strengthened by using heatmap. Figure 7 shows the correlation coefficients of all features along with their correlation with the target variable i.e. Congestion-level. end-node, start-node, day, way ID, hour, eta, agg-minutes, peak-hour, CI and maxspeed-real had positive correlation with the target where as quarter, agg-speed, holiday, min-time had negative correlation with the target. Agg-speed had negative correlation because we converted speed into eta standard whereas min-time and max speed-real attributes were used to detect outlier. Furthermore, CI was not only used to normalize eta but also to derive congestion level.

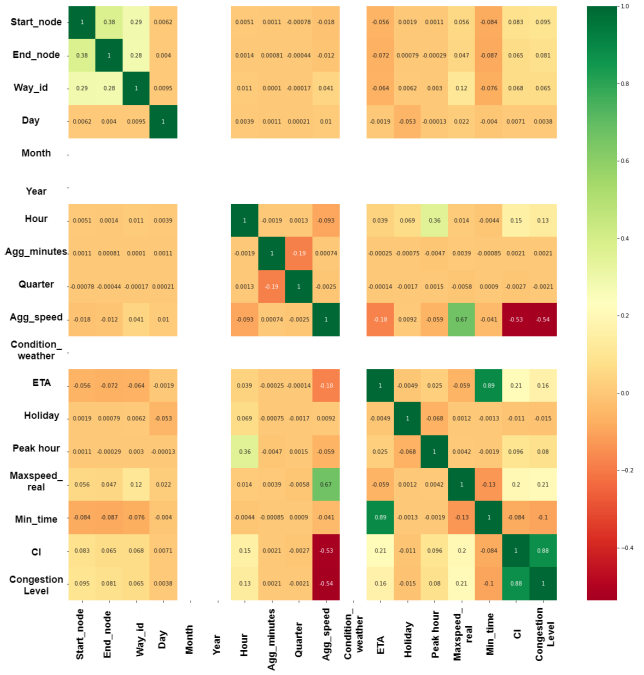

FIGURE 7. Correlation matrix of heterogeneous data sources.

2) RECURSIVE FEATURE ELIMINATION

Recursive Feature Elimination (RFE) is a wrapper selection method that recursively removes the attributes while training the model on the basis of remaining most relevant feature set. The algorithm assigned the weights through the coefficients of a linear model to feature set as an external estimator. External estimator then prunes less weight features and keeps the most significant features. Thus, RFE assigned the rank in such a way that the most relevant features were assigned rank 1 and true value. In our case, the heterogeneous data sources input feature set is (start_node, end_node, way_id, day, month, year, hour, agg_minutes, quarter, agg_speed, Condition_Weather, eta, holiday, peak hour, maxspeed_real, min_time, CI).

After applying RFE, 10 input features out of a pool of 17 features namely start_node, end_node, way_id, day, month, year, hour, quarter, agg_speed, maxspeed_real were selected.

3) RANDOM FOREST FOR FEATURE IMPORTANCE

Random Forest is a combination of multiple decision tree that are used to improve the accuracy by taking averaging of the data set. It is also used to extract important features by using scores. Figure 8 shows the features set and their respective scores in x-axis and y-axis, respectively. Figure 8 depicts that start-node, end-node, way-id, hour, aggregate speed, eta, max speed, min time, and congestion index have highest score as compared to others features.

We worked on multi time stamp, multi class and single label classification data set. Our data set consisted of three classes including smooth, congested and Highly Congested.

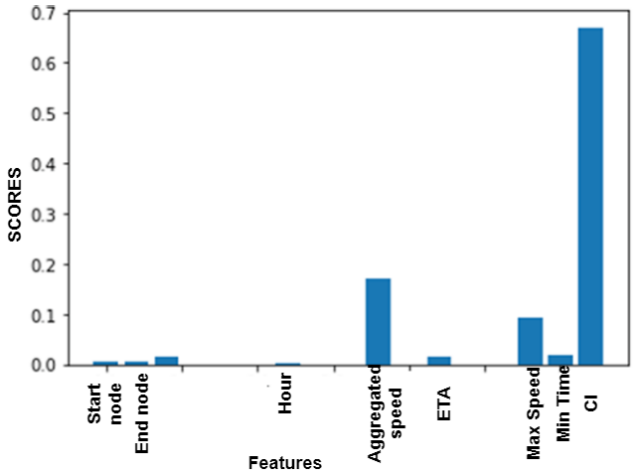

FIGURE 8. Random forest for feature importance applied on hybrid feature space.

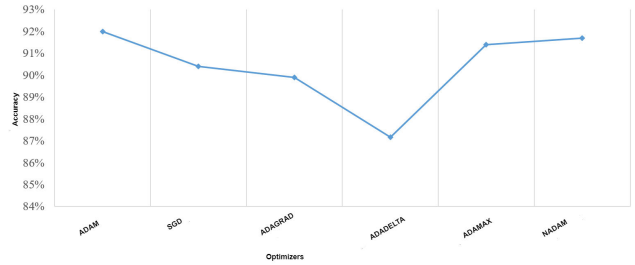

FIGURE 9. Accuracy values of the different Optimizers based on proposed GRU-LSTM model.

TABLE 9. Accuracy table of different ML and deep learning models.

Model	Accuracy
XGBOOST	71 percent
SVM	39 percent
RANDOMFOREST	83.9 percent
LSTM	92 percent
GRU	92 percent
MLP	67.4 percent
LSTM − GRU	93.7 percent
GRU − LSTM	**95 percent**

TABLE 10. Proposed GRU_LSTM model hyper-parameters configuration.

Model Hyper Parameters	Values
LearningMode	Batch Learning
BatchSize	512
LearningRate	0.001
Noofepochs	25
NoofHiddenLayers	04
HiddenUnits	128
DropoutRatio	0.2
ActivationFunction	tanh
OutputUnits	3
OutputType	Single Label, Multiple Classes
OutputLayerActivationFunction	Softmax
Optimizer	Adam
LossFunction	Categorical Crossentropy

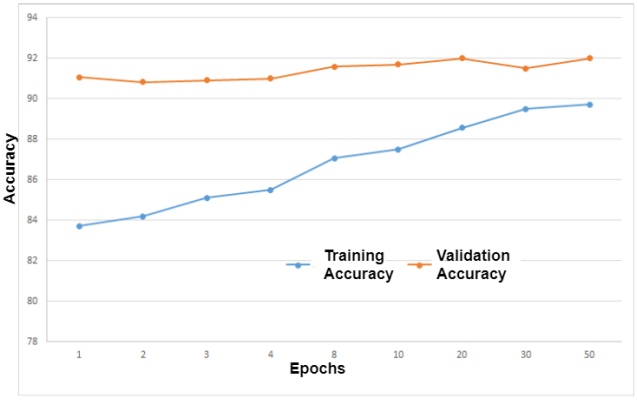

FIGURE 10. Impact of timestamp on LSTM.

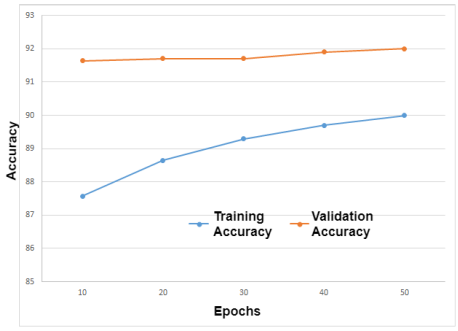

FIGURE 11. Impact of timestamp on GRU.

For applying the classification models, some features related to traffic patterns including max speed per segment and minimum estimated time, were derived in each segments at specific interval of time from our integrated data. This derived feature was then used in computing the congestion index which is given in the equation 11. Algorithm 5 calculated different segments of the road network Congestion Indices. Same was then applied on various congestion labels in accordance with the thresholds depicted in the Table 9. The final form of proposed GRU_LSTM model is summarized in the Table 10.

Data was recorded at 15 minutes of time resolution and less than or equal to 1 km of the space resolution.

10 shows the timestamp positive trends towards training accuracy as well as validation accuracy. X-axis shows the timestamp and Y axis shows the accuracy. Figure 11 shows that by increasing timestamp, training accuracy of LSTM and GRU improved from 84 percent to 89.9 percent and 85 percent to 90 percent respectively. Figure 12 visualizes the two graphs. Left side shows the learning curve where as right side shows the cost curve. X axis indicates the epochs and Y axis shows the accuracy. In the learning curve, validation accuracy touches the 93.17 percent and training accuracy reaches approximately 92 percent. In the cost curve, X axis indicates the epochs and Y axis shows the loss. In cost curve training

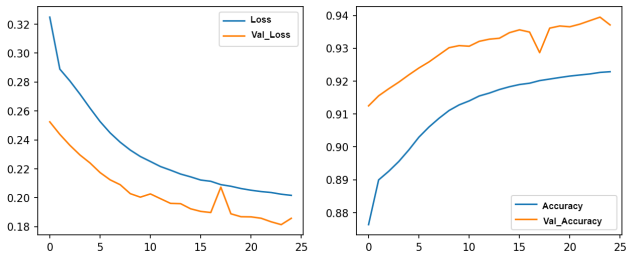

FIGURE 12. Learning vs cost curve for LSTM_GRU.

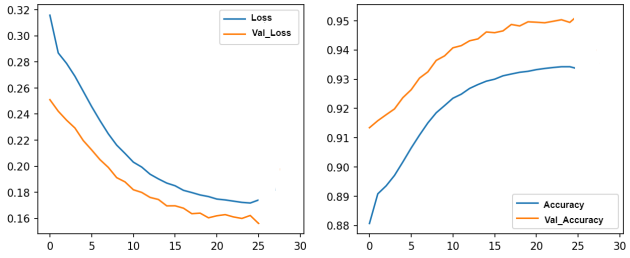

FIGURE 13. Learning vs cost curve for GRU_LSTM.

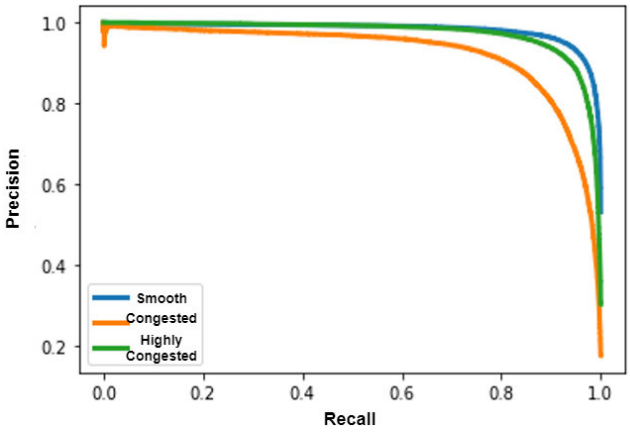

FIGURE 14. PR curve for our proposed GRU_LSTM model.

loss depreciates from 0.33 to 0.21. Where as validation loss dropped from 0.25 to 0.19. Figure 13 describes that validation loss decreases in GRU-LSTM from.25 to .16 with an average validation accuracy of 95.19 percent. However GRU-LSTM provided the most promising results in our scenario Table 9 depicts that GRU-LSTM yielded promising results with an accuracy of 95 percent where as classical classification techniques were unable to yield the suitable results. In classical techniques, we applied XGBoost, SVM and Random Forest. Among classical techniques, random forest provided results with 83.9 percent accuracy. on the other hand, among deep learning techniques including MLP, LSTM and GRU, GRU-LSTM produced best accuracy.

Figure 14 shows the Precision Recall(PR) near to 1. X-axis represents the recall and Y axis depicts the precision near to 1. However GRU-LSTM provided the most promising results in our case.

Firstly, we fixed the time of epoch to 25 and tested the accuracy behaviour when the optimizer changed. Optimizer was used in the current model to minimize the loss and maximize the accuracy. From Figure 9, the highest accuracy value of GRU-LSTM was achieved from Adam when only the optimizer function changed and other parameters remained same as default values.

V. CONCLUSION

This paper describes a mechanism to integrate multiple sources of data into a hybrid feature space. Basically, It utilizes an ETA based congestion index as a road network state evaluation indicator that distributes the traffic state primarily into three categories ranging from smooth to congested to Highly Congested class. We also integrated the traffic load, GPS, weather, special conditions with the OSM data set and employed different deep learning and machine learning algorithms. Among classical learning techniques, Random Forest provided the best results whereas in deep learning algorithms, GRU-LSTM proved to be the best with the highest accuracy. From current study, it can be concluded that deep learning techniques are the most reliable learning techniques providing maximum accuracy and better yield than classical learning techniques when are applied in traffic congestion problems on arterial roads. This further paves way towards automatic labelling of the classes instead of using congestion indexes and automatic optimization of hyper parameters using adaptive techniques in future studies. In the future, we will work on real-time traffic data on Vehicular Ad Hoc Networks (VANET) [59], [60] that permit vehicles to communicate with each other and improve traffic safety.

REFERENCES

[1] J. Guo, Y. Liu, Q. K. Yang, Y. Wang, and S. Fang, "GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model," *Transportmetrica A, Transp. Sci.*, vol. 17, no. 2, pp. 190–211, 2021.
[2] P. Lasley, *Urban Mobility Report*. College Station, TX, USA: Texas Transportation Institute, 2019.
[3] I. Hussain and S. J. Park, "HealthSOS: Real-time health monitoring system for stroke prognostics," *IEEE Access*, vol. 8, pp. 213574–213586, 2020.
[4] I. Hussain, M. A. Hossain, R. Jany, M. A. Bari, M. Uddin, A. R. M. Kamal, Y. Ku, and J.-S. Kim, "Quantitative evaluation of EEG-biomarkers for prediction of sleep stages," *Sensors*, vol. 22, no. 8, p. 3079, Apr. 2022.
[5] I. Hussain and S. J. Park, "Big-ECG: Cardiographic predictive cyber-physical system for stroke management," *IEEE Access*, vol. 9, pp. 123146–123164, 2021.
[6] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.
[7] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability*, vol. 12, no. 11, p. 4660, Jun. 2020.
[8] Y.-Y. Chen, Y. Lv, Z. Li, and F.-Y. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 132–137.
[9] N. Zafar and I. U. Haq, "Traffic congestion prediction based on estimated time of arrival," *PLoS ONE*, vol. 15, no. 12, Dec. 2020, Art. no. e0238200.
[10] M. Chikaraishi, P. Garg, V. Varghese, K. Yoshizoe, J. Urata, Y. Shiomi, and R. Watanabe, "On the possibility of short-term traffic prediction during disaster with machine learning approaches: An exploratory analysis," *Transp. Policy*, vol. 98, pp. 91–104, Nov. 2020.

[11] J. Mena-Oreja and J. Gozalvez, "A comprehensive evaluation of deep learning-based techniques for traffic prediction," *IEEE Access*, vol. 8, pp. 91188–91212, 2020.

[12] T. Afrin and N. Yodo, "A probabilistic estimation of traffic congestion using Bayesian network," *Measurement*, vol. 174, Apr. 2021, Art. no. 109051.

[13] J. Wang, I. Tsapakis, and C. Zhong, "A space–time delay neural network model for travel time prediction," *Eng. Appl. Artif. Intell.*, vol. 52, pp. 145–160, Jun. 2016.

[14] R. Yu, G. Wang, J. Zheng, and H. Wang, "Urban road traffic condition pattern recognition based on support vector machine," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 13, no. 1, pp. 130–136, Feb. 2013.

[15] J. Miao and L. Niu, "A survey on feature selection," *Proc. Comput. Sci.*, vol. 91, pp. 919–926, Jan. 2016.

[16] S. Zhang, Y. Yao, J. Hu, Y. Zhao, S. Li, and J. Hu, "Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks," *Sensors*, vol. 19, no. 10, p. 2229, May 2019.

[17] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[18] L. Liu and R.-C. Chen, "A novel passenger flow prediction model using deep learning methods," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 74–91, Nov. 2017.

[19] M. N. Sweet, "Do firms flee traffic congestion?" *J. Transp. Geogr.*, vol. 35, pp. 40–49, Feb. 2014.

[20] S. Majumdar, M. M. Subhani, B. Roullier, A. Anjum, and R. Zhu, "Congestion prediction for smart sustainable cities using IoT and machine learning approaches," *Sustain. Cities Soc.*, vol. 64, Jan. 2021, Art. no. 102500.

[21] A. Mondschein and B. D. Taylor, "Is traffic congestion overrated? Examining the highly variable effects of congestion on travel and accessibility," *J. Transp. Geogr.*, vol. 64, pp. 65–76, Oct. 2017.

[22] X. Han and Y. Shi, "Online traffic congestion prediction based on random forest," in *Proc. 4th Int. Conf. Mechatronics, Mater., Chem. Comput. Eng.*, 2015, pp. 1–7.

[23] D. Impedovo, V. Dentamaro, G. Pirlo, and L. Sarcinella, "TrafficWave: Generative deep learning architecture for vehicular traffic flow prediction," *Appl. Sci.*, vol. 9, no. 24, p. 5504, Dec. 2019.

[24] L. Li, H. Lin, J. Wan, Z. Ma, and H. Wang, "MF-TCPV: A machine learning and fuzzy comprehensive evaluation-based framework for traffic congestion prediction and visualization," *IEEE Access*, vol. 8, pp. 227113–227125, 2020.

[25] X. M. Chen, M. Zahiri, and S. Zhang, "Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 76, pp. 51–70, Mar. 2017.

[26] A. Candel, V. Parmar, E. LeDell, and A. Arora, "Deep learning with H_2O," H2O AI, Mountain View, CA, USA, Tech. Rep., 2016.

[27] M. Fang, L. Tang, X. Yang, Y. Chen, C. Li, and Q. Li, "FTPG: A fine-grained traffic prediction method with graph attention network using big trace data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5163–5175, Jun. 2022.

[28] S. Zhang, S. Li, X. Li, and Y. Yao, "Representation of traffic congestion data for urban road traffic networks based on pooling operations," *Algorithms*, vol. 13, no. 4, p. 84, Apr. 2020.

[29] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2123–2133, Aug. 2016.

[30] D.-H. Shin, K. Chung, and R. C. Park, "Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data," *IEEE Access*, vol. 8, pp. 150784–150796, 2020.

[31] H. Liu, H. Xu, Y. Yan, Z. Cai, T. Sun, and W. Li, "Bus arrival time prediction based on LSTM and spatial–temporal feature vector," *IEEE Access*, vol. 8, pp. 11917–11929, 2020.

[32] H. Xu and J. Ying, "Bus arrival time prediction with real-time and historic data," *Cluster Comput.*, vol. 20, no. 4, pp. 3099–3106, Dec. 2017.

[33] N. Servos, X. Liu, M. Teucke, and M. Freitag, "Travel time prediction in a multimodal freight transport relation using machine learning algorithms," *Logistics*, vol. 4, no. 1, p. 1, Dec. 2019.

[34] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 297–322, Oct. 2019.

[35] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transp. Res. C, Emerg. Technol.*, vol. 112, pp. 62–77, Mar. 2020.

[36] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, 2016.

[37] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 372–385, Mar. 2019.

[38] F. Guo, J. W. Polak, and R. Krishnan, "Predictor fusion for short-term traffic forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 90–100, Jul. 2018.

[39] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Inf. Syst.*, vol. 64, pp. 266–280, Mar. 2017.

[40] J. Grengs, J. Levine, Q. Shen, and Q. Shen, "Intermetropolitan comparison of transportation accessibility: Sorting out mobility and proximity in San Francisco and Washington, D.C," *J. Planning Educ. Res.*, vol. 29, no. 4, pp. 427–443, Jun. 2010.

[41] N. D. Chan and S. A. Shaheen, "Ridesharing in North America: Past, present, and future," *Transp. Rev.*, vol. 32, no. 1, pp. 93–112, Jan. 2012.

[42] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *Proc. 10th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2017, pp. 361–364.

[43] H. Yi and K.-H.-N. Bui, "An automated hyperparameter search-based deep learning model for highway traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5486–5495, Sep. 2021.

[44] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.

[45] L. Cai, Q. Chen, W. Cai, X. Xu, T. Zhou, and J. Qin, "SVRGSA: A hybrid learning based model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1348–1355, Sep. 2019.

[46] L. Han and Y. Huang, "Short-term traffic flow prediction of road network based on deep learning," *IET Intell. Transp. Syst.*, vol. 14, no. 6, pp. 495–503, Jun. 2020.

[47] E. Talen and J. Koschinsky, "The walkable neighborhood: A literature review," *Int. J. Sustain. Land Use Urban Planning*, vol. 1, no. 1, pp. 42–63, Mar. 2013.

[48] M. M. Rahman, M. M. M. Shuvo, M. I. Zaber, and A. A. Ali, "Traffic pattern analysis from GPS data: A case study of Dhaka city," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Mar. 2018, pp. 1–6.

[49] V. Kapoor, D. Saxena, V. Raychoudhury, and S. Kumar, "Real time building and maintaining causal congestion graph for intelligent traffic management," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 770–775.

[50] M. N. Borhan, D. Syamsunur, N. M. Akhir, M. R. M. Yazid, A. Ismail, and R. A. Rahmat, "Predicting the use of public transportation: A case study from Putrajaya, Malaysia," *Sci. World J.*, vol. 2014, pp. 1–9, Oct. 2014.

[51] Y. Ren, H. Chen, Y. Han, T. Cheng, Y. Zhang, and G. Chen, "A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes," *Int. J. Geogr. Inf. Sci.*, vol. 34, no. 4, pp. 802–823, Apr. 2020.

[52] T. Li, A. Ni, C. Zhang, G. Xiao, and L. Gao, "Short-term traffic congestion prediction with conv–BiLSTM considering spatio-temporal features," *IET Intell. Transp. Syst.*, vol. 14, no. 14, pp. 1978–1986, Dec. 2020.

[53] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: A GRU-based deep learning approach," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 578–585, Sep. 2018.

[54] J. Li, F. Guo, A. Sivakumar, Y. Dong, and R. Krishnan, "Transferability improvement in short-term traffic prediction using stacked LSTM network," *Transp. Res. C, Emerg. Technol.*, vol. 124, Mar. 2021, Art. no. 102977.

[55] S. Khan, S. Nazir, I. García-Magariño, and A. Hussain, "Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion," *Comput. Electr. Eng.*, vol. 89, Jan. 2021, Art. no. 106906.

[56] K. N. Qureshi, A. H. Abdullah, O. Kaiwartya, S. Iqbal, R. A. Butt, and F. Bashir, "A dynamic congestion control scheme for safety applications in vehicular ad hoc networks," *Comput. Electr. Eng.*, vol. 72, pp. 774–788, Nov. 2018.

[57] 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.
[58] M. M. Rathore, A. Paul, S. Rho, M. Khan, S. Vimal, and S. A. Shah, "Smart traffic control: Identifying driving-violations using fog devices with vehicular cameras in smart cities," *Sustain. Cities Soc.*, vol. 71, Aug. 2021, Art. no. 102986.
[59] H. Shahwani, S. A. Shah, M. Ashraf, M. Akram, J. Jeong, and J. Shin, "A comprehensive survey on data dissemination in vehicular ad hoc networks," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100420.
[60] M. M. Rathore, S. A. Shah, A. Awad, D. Shukla, S. Vimal, and A. Paul, "A cyber-physical system and graph-based approach for transportation management in smart cities," *Sustainability*, vol. 13, no. 14, p. 7606, Jul. 2021.
[61] T. Bokaba, W. Doorsamy, and B. S. Paul, "A comparative study of ensemble models for predicting road traffic congestion," *Appl. Sci.*, vol. 12, no. 3, p. 1337, Jan. 2022.
[62] G. Chen and J. Zhang, "Applying artificial intelligence and deep belief network to predict traffic congestion evacuation performance in smart cities," *Appl. Soft Comput.*, vol. 121, May 2022, Art. no. 108692.
[63] W. Cheng, J.-L. Li, H.-C. Xiao, and L.-N. Ji, "Combination predicting model of traffic congestion index in weekdays based on LightGBM-GRU," *Sci. Rep.*, vol. 12, no. 1, pp. 1–13, Feb. 2022.
[64] M. Z. Mehdi, H. M. Kammoun, N. G. Benayed, D. Sellami, and A. D. Masmoudi, "Entropy-based traffic flow labeling for CNN-based traffic congestion prediction from meta-parameters," *IEEE Access*, vol. 10, pp. 16123–16133, 2022.
[65] A. Izhar, S. M. K. Quadri, and S. A. M. Rizvi, "Hybrid feature based label generation approach for prediction of traffic congestion in smart cities," in *Proc. 3rd Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2020, pp. 991–997.
[66] Y. Zhang, S. Yang, and H. Zhang, "Research on urban traffic industrial management under big data: Taking traffic congestion as an example," *J. Adv. Transp.*, vol. 2022, Jun. 2022, Art. no. 1615482.

NOUREEN ZAFAR is currently pursuing the Ph.D. degree with the Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan. She is doing research in traffic congestion prediction on heterogeneous data sources. She is also a Lecturer with PMAS-AAUR. Her research interests include traffic congestion prediction, machine learning, and deep learning.

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 research interests include intelligent transportation systems, logistics, and autonomous vehicles.

HUNIYA SOHAIL received the M.S. degree in computer science from the Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan. She works as a Data Scientist in a multinational software company. She is also an Active Researcher. Her research interests include data science, machine learning, and deep learning.

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 Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad. His research interests include machine learning, deep learning, trajectory pattern mining, intelligent transportation systems, and traffic data analysis.

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.