Case Studies on Transport Policy 10 (2022) 965–975

Contents lists available at ScienceDirect

Case Studies on Transport Policy

journal homepage: www.elsevier.com/locate/cstp

Forecasting public transit passenger demand: With neural networks using APC data

Shivaraj Halyal [a,*], Raviraj H. Mulangi [a], M.M. Harsha [b]

[a] Department of Civil Engineering, National Institute of Technology, Surathkal, Karnataka, India
[b] Siddaganga Institute of Technology, Tumkur, Karnataka, India

ARTICLE INFO

Keywords:
Intelligent Transport System
APC
Forecasting of Bus Passenger Demand
SARIMA
LSTM

ABSTRACT

The implementation of Intelligent Transportation Systems (ITS) as a part of smart mobility is crucial for solving the current problems of the transportation industry. The setting up and maintenance of ITS requires not only the current passenger demand but also the future passenger demand. The future passenger demand can be obtained with time-series forecasting carried out with different techniques. With the advancements in the technological field, modern and more advanced methods of time-series forecasting using deep learning are being preferred over traditional forecasting techniques. However, the research carried out in this regard is quite limited, particularly considering the Indian scenario. Hence this research work focuses on exploring the performance of deep learning forecasting techniques considering the aspects mentioned previously. Here, the forecasting of passenger demand was done with Long Short-Term Memory (LSTM) using the three months Automatic Passenger Counter (APC) data of the Hubballi-Dharwad Bus Rapid Transit System (HDBRTS) as part of a case study. Then the forecasting of passenger demand was also done with Seasonal Autoregressive Integrated Moving Average (SARIMA), and the comparison of the forecasting accuracy of both methods was made using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Furthermore, to validate the results, novel approach has been adopted for the process, by following some more time-series resampled with different time intervals. Study shows that LSTMs will be used satisfactorily in the traffic conditions of developing counties, for forecasting passenger demand using APC data. Study also provides detailed guiding methodologies of advanced methods of passenger forecasting along with conventional ones.

1. Introduction

The transportation industry is tightly integrated with the day-to-day life of majority of the population which constitutes a significant part of it. Therefore, any advancements and improvements in this sector will directly impact people's lives and almost all other industries. The transportation industry is one of the major contributing sectors to environmental pollution (Overview, 0000). Much of this can be attributed to the inefficient transportation system and the rapid increase in the number of vehicles (Sharma et al., 2011). The increased purchasing power of people is leading to the ever-increasing number of vehicles on the road, is also creating problems such as congestion. This congestion causes increased travel durations, increased waiting times and increased travel distances, all of which eventually results in an inefficient transportation system. This inefficient transportation leads to excessive consumption of valuable resources. The increased number of vehicles and congestion also causes an increased number of accidents (Satheesh Kumar, 0000). Hence, a complete overhaul of the whole industry becomes inevitable.

The concept of 'Smart Mobility' (Benevolo et al., 2011; Šemanjski et al., 2018) is an innovative solution to tackle this problem. Smart mobility connects various elements of technology and mobility, and ITS is a step towards implementing it. The ITS integrates users of the transportation system with vehicles and infrastructure using information and communication technology. The technology used can vary from a basic management system to more advanced applications that integrate real-time data and feedback from a number of other sources. Additionally, predictive techniques can also be used for advanced modelling, forecasting and comparison with historical baseline data. (Satheesh Kumar, 0000) ITS also helps in decreasing the travel durations and reducing congestion. With reduced congestion, the capacity of narrow roads will also be increased. One of the critical aspects of ITS is

* Corresponding author.
E-mail address: shivaraj.halyal@kletech.ac.in (S. Halyal).

https://doi.org/10.1016/j.cstp.2022.03.011
Received 29 November 2021; Received in revised form 11 February 2022; Accepted 18 March 2022
Available online 23 March 2022

its strong endorsement of mass transportation.

The mass transportation system of ITS has to be designed considering many factors, passenger demand being an important one. However, with the current passenger demand, the future passenger demand should also be considered while designing. Demand forecast for public transportation provides practical picture of its future usage and is essential for effective policy making and planning (Nguyen et al., 2020). In general, public transit operation conversion from heterogeneous to homogenous traffic condition, passenger demand management implementations, enhancing the main and feeder systems etc. More specifically, short-term forecasting of passenger demand is essential for hiring or scheduling carrier vehicles and personnel, maintenance of infrastructures, and allocation of other resources. Similarly, long-term forecasting of passenger demand is needed for the construction of permanent infrastructures such as roads or tracks, stops, stations, terminals, depots, administrative or departmental buildings, and procurement of carrier vehicles. Hence understandably, passenger demand forecasting is a crucial and an inevitable step.

The passenger demand is represented through time series. Time series is a set of observations recorded sequentially over a period of time. The future values of a time series can be predicted based only on historical observations of the time series or based on external controlling factors, or by a combination of both. The time series forecasting can be done with traditional methods such as exponential smoothing and autoregressive integrated moving average (ARIMA). In addition, if seasonality is present, seasonal ARIMA can be used to handle the seasonality. In addition to this, the traditional methods can be improved and tweaked only to a certain extent to get better results.

With the exponential growth of the processing power of computers, more advanced and complex methods can also be used for forecasting. One of such methods is to use Artificial Neural Networks (ANNs). Furthermore, forecasting with Long Short-Term Memory (LSTM), a special case of Recurrent Neural Networks (RNN), is gaining popularity as it can efficiently learn long term dependencies. The primary purpose of this research work is to explore the application of LSTM for passenger demand forecasting in the Indian context. Mainly due to no studies reporting on productive usage of LSTM application in the passenger demand forecasting for traffic and transit characteristics pertaining to developing countries. Hence, to achieve this, three months of APC data has been used to forecast passenger demand by means of traditional seasonal ARIMA method and the advanced LSTM method. Then the forecasting efficacy of both the methods was compared to identify the better method. The comparison was also done employing time-series resampled with different time-frames to validate the results further, an approach not well researched on.

2. Literature review

The studies carried out on passenger demand, the factors affecting it, and its forecasting have been aplenty. However, most of the studies have used traditional methods of forecasting such as linear regression, trending models, exponential smoothing, moving average, and ARIMA to forecast bus, rail passenger demand and traffic flow prediction with unstructured data or data collected with traditional methods.

Banerjee et al. (Banerjee et al., 2020) did a review on the research performed on demand forecasting in the transportation industry. They listed a large number of articles and made an attempt to classify them systematically. Furthermore, they assessed each methodology and suggested different measures others can adopt during future researches.

Cyril et al. (Cyril et al., 2018) used seasonal ARIMA models to predict the bus passenger demand from Thiruvananthapuram to five other districts of Kerala using the Electronic Ticketing Machine (ETM) data of 2010 to 2013. Similarly, Cyril et al. (Cyril et al., 2019) used different Holt-Winters' modelling methods and compared the predictive accuracy of those for the same data used by Cyril, A et al.

Cyprich et al. (Cyprich et al., 2013) employed all of Box-Jenkins, exponential smoothing and multiple linear regression models to forecast the short-term bus passenger demand. Anvari et al. (Anvari et al., 2016) developed a time-series forecasting framework using the Box-Jenkins method. The developed framework, which is comprehensive, automated, accurate, and fast, was tested using the passenger data from Istanbul Metro.

Milenković et al. (Milenković et al., 2018) used seasonal ARIMA to forecast the monthly passenger flow of Serbian railways using the data from January 2004 to June 2014. Gummadi et al. (Gummadi and Edara, 2018) used ARIMA and seasonal ARIMA for bus passenger flow prediction using the data from April 2016 to December 2016 of Andhra Pradesh State Road Transport Corporation. Ye et al. (Ye et al., 2019) used the ARIMA model to predict the daily bus passenger flow volume and compared the prediction performances in the case using whole weekday data against the case using weekday-only data. The data used was from January 2018 to March 2018 of Jiaozuo in China.

Kumar et al. (Kumar and Vanajakshi, 2015) used seasonal ARIMA for the short-term prediction of the traffic with limited input data comprising of only three consecutive days from Rajiv Gandhi Road in Chennai, and Rabbani et al. (Rabbani et al., 2021) used both seasonal ARIMA and Exponential Smoothing techniques to forecast the accident rates in Pakistan and compared the predictive accuracy of both methods.

Xue et al. (Xue et al., 2015) proposed an Interactive Multiple Model (IMM) filter algorithm-based model which combines individual forecasting models to predict short-term bus passenger demand for the passenger boarding data from August 2013 to December 2013 of Shenzhen, China. Hyndman et al. (Hyndman and Athanasopoulos, 2018) provided an excellent and detailed account of the traditional forecasting techniques in the form of a book.

Now focusing on the advanced and modern methods of forecasting, there have been several works concentrating on forecasting passenger demand using ANNs.

Li et al. (Li et al., 2019) used LSTM to forecast the short-term departing passenger flow of the Wuhan-Guangzhou high-speed railway using data from January 2010 to December 2015. Ouyang et al. (Ouyang et al., 2020) used LSTM to predict bus passenger flow considering both historical data and real-time data using one-month data from 1st October 2015 to 30th October 2015.

Gallo et al. (Gallo et al., 2019) proposed the use of Artificial Neural Networks (ANNs) to forecast metro onboard passenger flow as a function of passenger counts at station turnstiles and tested using the data of Line 1 of the Naples metro system of Italy. Xiong et al. (Xiong et al., 2019) used two deep learning neural networks, an LSTM and convolutional neural network (CNN), to predict the passenger flow time series and spatiotemporal series respectively for Beijing Metro using the data from January 2015 to February 2016.

Han et al. (Han et al., 2019) proposed a hybrid, optimised LSTM network based on Nesterov accelerated adaptive moment estimation (Nadam) and the stochastic gradient descent algorithm (SGD) and tested it using the bus passenger flow data of Qingdao, China of March 2016. Farahani et al. (Farahani et al., 2020) proposed a deep learning model called Variational LSTM Encoder (VLSTM-E) to predict short-term traffic flow. The authors used the data from 1st January 2019 to 30th May 2019 of district seven California.

Yang et al. (Yang et al., 2019) proposed an enhanced long-term feature based on LSTM (ELF-LSTM) neural network, which strengthens the long temporal dependency features embedded in passenger flow data and incorporates the short-term features to predict passenger flow in the rail transit system. The authors tested this using the data from 1st January 2014 to 30th March 2015 of Chongqing, China. Shahriari et al. (Shahriari et al., 2020) developed an ensemble of ARIMA models (E-ARIMA) by randomly subsampling the data and compared the validity of E-ARIMA with ARIMA and LSTM.

However, the literature review shows, studies concentrating on the modern techniques of forecasting in developing countries like India is substantially less. Research work carried out on passenger demand forecasting with deep learning techniques that too using APC data is very minimal. This can be attributed to both deep learning being a relatively new concept and the non-availability of a considerable amount of accurate, systematic and reliable data till the recent past. Other than these, effect of different time frames where time series have been reframed on forecasted results also remains an unexplored area. This work embarks to address the above-mentioned gaps clearly, through passenger demand forecasting using reliable APC data of an entire BRTS network. Results of study clearly shows that LSTMs could be satisfactorily used for forecasting passenger demand with APC data to obtain better results in comparing with conventional methodologies. Study also provides guiding methodologies to be benefited by more advanced methods of passenger forecasting in place with conventional ones.

A tabular representation of the previously explained literature review carried out on traditional methods of forecasting is organised in a table format and given here as Table 1. Method of calculation for different measures of forecasting accuracy used is given below the table.

Tabular representation of the previously explained literature review carried out on modern, advanced methods of forecasting using deep learning techniques is organised in a table format and given here as Table 2. Expansions for acronyms used are given below the table.

3. Study area and data source

Hubballi-Dharwad is located in Dharwad district, situated in the north-western part of Karnataka state of India. Dharwad district lies between 15°02′ and 15°51′ North latitudes and 73°43′ and 75°35′ East longitudes which comprises of an area of 4230 km^2. The cities have a combined population of 9,43,857 as per the 2011 census. The BRTS successfully functioning between twin cities, Hubballi-Dharwad was considered for this research work. The Hubballi-Dharwad BRTS was established to serve public transport as a part of the Sustainable Urban Transport Project (SUTP) and funded by the Government of Karnataka, Ministry of Housing and Urban Affairs (MHUA), World Bank and Global Environment Facility (GEF).

The data collection for transportation and traffic-related studies can be done either by manual methods (field-based observations) or by automated technology-based methods (Tourangeau et al., 1997; Hummer, 1994; Zheng et al., 2016). The traditional manual methods of data collection necessitate the investment of a substantial amount of human resources, efforts and time. These methods are also subjected to human errors to an extent. With the advancements in the technological field, automated technology-based data collection methods, which are accurate, reliable and readily available without the previously mentioned disadvantages, are being preferred these days. Automated Vehicle Location (AVL) (Du and Barth, 2008) and Automatic Fare Collection System (AFCS) (Alfred Chu et al., 2009; Jia et al., 2019) are some of the examples of these technology-based methods.

The system operation was started on 2nd October 2018. For the current research work, the passenger data from 1st December 2019 to 29th February 2020 (91 days) of all the stations of the HDBRTS network obtained by the APC was considered. The system was designed such that, every passenger should get a ticket (even if the passenger has a pass) which should be scanned both at the turnstiles installed at boarding and alighting station to enter or exit the station. Hence, the ticket data can be reliably considered to count passenger. The HDBRTS network consists of 35 stations, starting from City Bus Terminal (CBT) Hubballi and ends at Dharwad New Bus Stand and Dharwad BRTS Terminal. The route map of the HDBRTS network is given in Fig. 1. The original passenger data was

Table 1
Summary of literature review on traditional methods of forecasting.

Author and Year	Field of Study	Study Area	Method	Time Period of Data	Measure of Forecasting Accuracy
Cyril et al. (2018) (2018)	Bus Passenger Demand Forecasting	Kerala, India	Seasonal ARIMA	Ticket Data between 2010 and 2013	MAPE
Cyril et al. (2019) (2019)	Bus Passenger Demand Forecasting	Kerala, India	Holt-Winters'	Ticket Data between 2010 and 2013	MAPE
Cyprich et al. (2013) (2013)	Bus Passenger Demand Forecasting	Slovakia	ARIMA, Box-Jenkins, Exponential Smoothing, Multiple Linear Regression	January 2000 – December 2007	MAPE, MPE, RMSE
Anvari et al. (2016) (2016)	Metro Passenger Demand Forecasting	Istanbul, Turkey	Box-Jenkins	1st July 2011 – 11th November 2011	MAPE, RMSPE, TS
Milenković et al. (2018)	Railway Passenger Demand Forecasting	Serbia	Seasonal ARIMA, Seasonal Exponential Smoothing	January 2004 – June 2014	MAPE, RMSE
Gummadi and Edara (2018)	Bus Passenger Demand Forecasting	Andhra Pradesh, India	Seasonal ARIMA	1st April 2016 – 31st December 2016	–
Ye et al. (2019) (2019)	Bus Passenger Demand Forecasting	Jiaozuo, China	ARIMA	1st January 2018 – 31st March 2018	MAPE, SDE, MAD
Kumar and Vanajakshi (2015)	Traffic Flow Prediction	Chennai, India	Seasonal ARIMA	20th September 2012 – 22nd September 2012	MAPE
Rabbani et al. (2021)	Road Accidents Forecasting	Hayatabad, Peshawar, Pakistan	Seasonal ARIMA, Exponential Smoothing	January 2009 – May 2020	MAPE, MAE, RMSE
Xue et al. (2015) (2015)	Bus Passenger Demand Forecasting	Shenzhen, China	Seasonal ARIMA	August 2013 – November 2013	MAPE, MAE, RMSE

MAE – Mean Absolute Error, $MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t|$, where e_t is the error and d_t is the actual value at any time t within the testing set containing n observations.

MAPE – Mean Absolute Percentage Error, $MAPE = 100 \times \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{d_t}$

MPE – Mean Percentage Error, $MPE = 100 \times \frac{1}{n}\sum_{t=1}^{n}\frac{e_t}{d_t}$

RMSPE – Root Mean Squared Percentage Error, $RMSPE = 100 \times \sqrt{\frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{d_t}}$

RMSE – Root Mean Squared Error, $RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}e_t^2}$

TS – Tracking Signal, $TS = \frac{Bias}{MAD}$, where $Bias = \sum_{t=1}^{n}e_t$, and.$MAD = MAE$

SDE – Standard Deviation Error, $SDE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(e_t - \overline{e_t})^2}$ where.$\overline{e_t} = \frac{1}{n}\sum_{t=1}^{n}e_t$

Table 2
Summary Of Literature Review On Modern, Advanced Methods Of Forecasting Using Deep Learning.

Author and Year	Field of Study	Study Area	Method	Time Period of Data	Measure of Forecasting Accuracy	Remarks
Li et al. (2019) (2019)	Railway Passenger Demand Forecasting	Guangzhou, China	LSTM, SVM, KNN, RF.	1st January 2010 – 31st December 2015	MAPE, RMSE	Beijing-Guangzhou High-speed Rail (HSR) line, which passes through Wuhan, was considered for the study.
Ouyang et al. (2020) (2020)	Bus Passenger Demand Forecasting	Beijing, China	LSTM	1st October 2015 – 30th October 2015	MAE, RMSE	–
Gallo et al. (2019) (2019)	Metro Passenger Demand Forecasting	Naples, Italy	ANN	Simulation	R^2	The time-series data was obtained by simulating the passenger flow of Naples Metro.
Xiong et al. (2019) (2019)	Metro Passenger Demand Forecasting	Beijing, China	SARIMA, STARIMA, LSTM, CNN	January 2015 – February 2016	MAE, MRE, RMSE	–
Han et al. (2019) (2019)	Bus Passenger Demand Forecasting	Qingdao, China	SARIMA, LSTM, SVR.	1st March 2016 – 31st March 2016	MAPE, MAE, RMSE	–
Farahani et al. (2020) (2020)	Traffic Flow Prediction	California, USA.	VLSTM-E, LSTM, MCNNM, SAEs	1st January 2019 – 30th May 2019	MAPE, MAE, RMSE	The time-series data was obtained from the Caltrans Performance Measurement System (PeMS), which is collected from the detectors spanning across California.
Yang et al. (2019) (2019)	Railway Passenger Demand Forecasting	Chongqing, China	ELF-LSTM, ARIMA, LSTM, RNN, SVR	1st January 2014 – 30th March 2015	MAE, RMSE	–
Shahriari et al. (2020) (2020)	Traffic Flow Prediction	Sydney, Australia	E-ARIMA, ARIMA, LSTM	November 2017 – November 2018	MAPE, RMSE	–

SVM – Support Vector Machine.
KNN – K-nearest neighbour.
RF – Random Forest.
STARIMA – Space–Time Autoregressive Integrated Moving Average.
SVR – Support Vector Regression.
MCNNM – Multiple Convolutional Neural Network for Multivariate model.
SAE – Stacked Autoencoder.

obtained in.csv format containing the information about ‘date issued’, ‘time issued’, ‘operator ID’, ‘terminal ID’, ‘device type’, travel direction information such as ‘issued boarding station’ and ‘issued alighting station’, ‘payment method’, ‘ticket serial number’, ‘rider type’, ‘ridership’ and ‘total revenue’.

4. Data processing

The raw data required preprocessing and in case of the current research work, the raw data contained a total of 54,27,301 observations from 1st December 2019 to 29th February 2020. The extra information in all the other fields except ‘date issued’, ‘time issued’, ‘issued boarding station’, and ‘ridership’ was removed. Then total passengers boarded at each station in the considered 91-day period was calculated. Then top-5 stations with higher passenger density were identified, and only the data of these five stations – station 05, station 28, station 33, station 34 and station 35 – was further considered to study. The observations of each of the selected top-5 stations were filtered out and stored in separate.csv files. Then, the fields ‘date issued’ and ‘time issued’ were merged into a single field called ‘issued at’ and the field ‘issued boarding station’ was removed so that the only fields present were ‘issued at’ and ‘ridership’. Then, using the resample () function of the datetime module of Python, the passenger data of all of the top-5 stations was resampled into time intervals of 15 min, 30 min, 45 min and 60 min. In this way, a total of 20 time-series – five stations with four time-frames per station – were obtained. A sample of a final time-series ready for analysis is given in Table 3.

The passenger data was plotted into time-series. A time series is an ordered sequence of values of a quantitative random variable at equally spaced time points that measures the status of some activity over time (Anvari et al., 2016). Here, the total passenger count per time interval is plotted against the time. The time series, of 15 min, 30 min, 45 min and 60 min time intervals were plotted and observed for patterns such as trend, seasonality and outliers. Additionally, a day-wise time-series was also plotted and observed for the previously mentioned time-series patterns.

In time-series analysis trend is a long-term increase or decrease in the value of the observation. However, no apparent trend was observed in any of the time-series plotted. Seasonality is the presence of variations that occur at specific time intervals regularly over time. The seasonality can be of any time interval such as yearly, quarterly, weekly, daily or even hourly. Here, the data showed daily seasonality. An outlier is an observation that significantly differs from other observations of the same feature. Outliers are usually unexpected spikes or dips in the value of observations on the time-series graph and have to be removed from the data. Some ways of dealing with outliers include modifying them after identifying their source or replacing them with the mean values, or neglecting them (Kwak and Kim, 2017; Dhakal, 2017). There were hardly any outliers present across all the time-series here, and the ones present were simply due to technical or operational errors, so they were replaced by the mean values of passengers for that time. Moreover, it was observed that the passenger flow after 23:00 and before 06:00 was very negligible and almost nonexistent. So, the observations between that time period were removed and data contained only between 06:00 to 23:00 were considered for the further analysis. This means, as the data exhibits daily seasonality, the periodicity of the season will be 17, 23, 34 and 68 for 60 min, 45 min, 30 min and 15 min time-series, respectively.

The data was then divided into the training set and testing set. 80 % of the data was used for training, and the remaining 20 % was used for the testing (Medar et al., 2017). More accurately, the data from 1st December 2019 to 11th February 2020 (73 days) was used for training and the rest, from 12th February 2020 to 29th February 2020 (18 days), was reserved for testing.

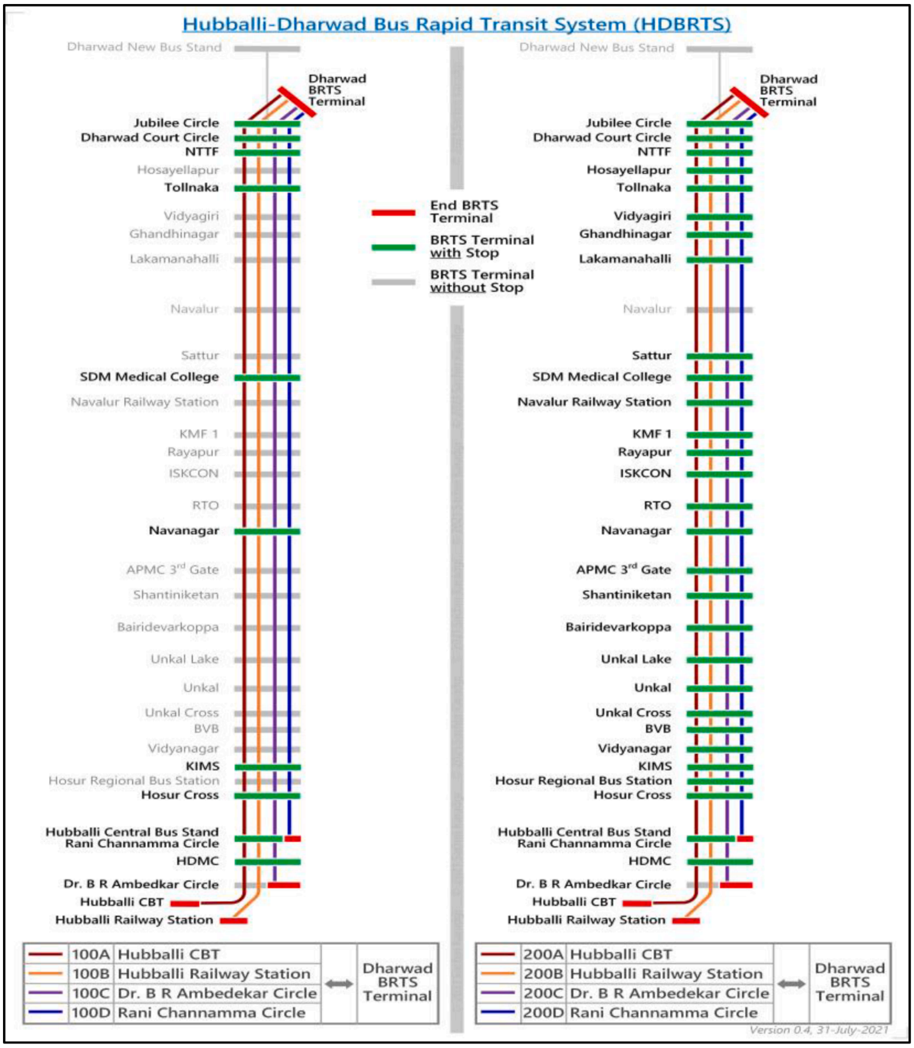

Fig. 1. Route Map of HDBRTS Network.

Table 3
A Sample Of Final Time-Series Ready For Analysis.

Issued at	Total Passengers
01-12-2019 06:00	19
01-12-2019 06:30	44
01-12-2019 07:00	142
01-12-2019 07:30	380
01-12-2019 08:00	467
01-12-2019 08:30	481
01-12-2019 09:00	500
01-12-2019 09:30	387
01-12-2019 10:00	480
01-12-2019 10:30	396
01-12-2019 11:00	490
01-12-2019 11:30	486
01-12-2019 12:00	631
01-12-2019 12:30	450
01-12-2019 13:00	535
01-12-2019 13:30	422
01-12-2019 14:00	466

5. Seasonal ARIMA

Autoregressive Integrated Moving Average (ARIMA) is a traditional forecasting technique that adds the lags of the differenced series (autoregressive (AR) terms) or lags of the forecast errors (moving average (MA) terms) or both to the prediction equation (Anvari et al., 2016). An ARIMA model consists of three parameters – p; d, and q ($ARIMA(p, d, q)$) where p represents autoregressive (AR) lags, and q represents moving average (MA) lags. The parameter d is the integration order representing the number of times the time series must be differenced to make it stationary. Furthermore, it is also possible to handle seasonality in the data using ARIMA models by including additional seasonal terms in the ARIMA models. The resulting model is termed as seasonal ARIMA model and is represented as,

$$SARIMA(p, d, q) \times (P, D, Q)_m$$

where m is the periodicity of the season or number of observations per season. Similar to ARIMA, the lowercase notations denote the non-seasonal part, whereas the uppercase notations denote the seasonal part. The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model but involve backshifts of the seasonal period. The additional seasonal terms are simply multiplied by the non-seasonal terms (Hyndman and Athanasopoulos, 2018).

The time-series has to be stationary for the application of ARIMA or Seasonal ARIMA. The mean and variance of a stationary time series are constant, and covariance is independent of time. So, the properties of the time-series do not depend upon the time at which it is observed. The stationarity of a time series can be determined either by graphical and summary statistics or by statistical tests (Phillips and Perron, 1988; Dickey and Fuller, 1979). The Augmented Dickey-Fuller (ADF.) Test for unit root is one such statistical test, which tests for the presence of a unit root that makes the time-series non-stationary. In this work, the ADF unit root test was used to determine the stationarity of the time-series and hence, effectively, to decide the order of normal differencing, *d*, and the order of seasonal differencing, *D* (Dickey and Pantula, 1987). As it was previously observed that the time series exhibits seasonality, the time series for all stations was made stationary by taking the first difference after the seasonal difference. Hence, the values of *d* and *D* were fixed as one.

Traditionally, the order of p, q, P and Q in $\mathrm{SARIMA}(p,d,q)\times(P,D,Q)_m$ is obtained by observing Auto-Correlation Function (ACF) and Partial Auto-correlation Function (PACF) plots of the time-series, which were made stationary previously. However, with the increasing computing power of modern processors, a large number of models can be fits in a minimal amount of time. Hence, here all possible permutations of p, q, P and Q subjected to the range given in the Table 4 and used to forecast from 12th February 2020 to 29th February 2020. The advantage of this method is that there is a better chance of finding the best model than the traditional method.

Then the values of Akaike Information Criteria (AIC) and Bayesian Information Criterion (BIC) of the most fit models were obtained. Furthermore, the errors for the forecasted model were obtained and then MAE and RMSE were also calculated and tabulated. AIC, BIC, MAE and RMSE are defined and explained in the subsequent paragraphs.

AIC and BIC are the measures that measure the accuracy of fit of the models. AIC is defined as,

$$AIC = T\log\left(\frac{SSE}{T}\right) + 2(k+2) \tag{1}$$

where *T* is the number of observations used for fitting and *k* is the number of predictors in the model. Similarly, BIC is defined as,

$$BIC = T\log\left(\frac{SSE}{T}\right) + (k+2)\log(T) \tag{2}$$

where *T* and *k* have the same meaning as in AIC. The model with minimum AIC and BIC values is the best forecasting model. If there is an actual underlying model present, the BIC will tend to select that model given enough data, but if the number of observations included (*T*) is large enough, the same models will be selected by both AIC and BIC. Hence, the model chosen using the BIC is either the same as that chosen using the AIC or the one with fewer terms as the BIC penalises the number of parameters more heavily than the AIC (Hyndman and Athanasopoulos, 2018). However, it is preferable to use both AIC and BIC in combination, giving equal importance to both (Kuha, 2004; Burnham and Anderson, 2004).

The difference between the actual value and forecasted value is known as forecast error. In this case, the term error does not imply a mistake, but it indicates the unpredictable part of an observation (Hyndman and Athanasopoulos, 2018). It is given by,

$$e_{T+h} = y_{T+h} - \widehat{y}_{T+h|T} \tag{3}$$

where the training data is given by $\{y_1,\cdots,y_T\}$ and the test data is given by $\{y_{T+1}, y_{T+2},\cdots\}$.

MAE and RMSE are two of the most widely used measures to represent errors. The definition of both MAE and RMSE are given below.

Table 4
Range For The Order Of Parameters For Seasonal ARIMA.

Time-frame	*p*	*q*	*P*	*Q*	*m*
15 min	0 to 68	0 to 68	0 to 2	0 to 2	68
30 min	0 to 34	0 to 34	0 to 2	0 to 2	34
45 min	0 to 23	0 to 23	0 to 2	0 to 2	23
60 min	0 to 17	0 to 17	0 to 2	0 to 2	17

$$MAE = mean(|e_t|) = \frac{1}{n}\times\sum_{t=1}^{n}|e_t| \tag{4}$$

$$RMSE = \sqrt{mean(e_t^2)} = \sqrt{\frac{1}{n}\times\sum_{t=1}^{n}e_t^2} \tag{5}$$

MAE is slightly more intuitive and easier to interpret and compute than RMSE. Optimising the forecasts for MAE results in forecasts of the median and optimising the forecasts for RMSE will result in forecasts of the mean. Moreover, RMSE is more sensitive to outliers than MAE (Vandeput, 2021). Suppose a model is selected solely based on the accuracy of fit measures such as AIC and BIC, it is possible that the selected model will be an actual underlying model which provides a reliable characterisation of the sources of uncertainty and understands the underlying data-generating mechanism (Ding et al., 2018). However, this does not guarantee the selection of a model with high forecasting accuracy. Then again, suppose a model is selected based only on the accuracy of forecasting measures such as MAE and RMSE will tend to select a model with excellent predictive performance but may not select a model which captures the actual underlying characteristics. Therefore, the final model was selected, considering all of AIC, BIC, MAE and RMSE giving equal importance to all four measures to select the models which are both robust and have excellent forecasting accuracy. The selected final models and their forecasting performance is discussed in the 'Results and Discussions' section of this paper.

The residual analysis was carried out on the selected final models to ensure that the model captures and utilises most of the features, patterns and information available in the data provided. The residuals in a time series model are what is left over after fitting a model (Hyndman and Athanasopoulos, 2018), or more precisely, residuals are the difference between the observations and the corresponding fitted values:

$$e_t = y_t - \widehat{y}_t \tag{6}$$

where e_t is the residual, y_t is the actual value and $\widehat{y}_t$ is the corresponding fitted value. As a part of the residual analysis, the residual graph, ACF and PACF plots, histogram of residuals were plotted and observed for the following properties:

- The residual plots looked like white noise.
- There was no correlation between the residuals.
- The residuals were normally distributed.

If the residual plot looks like the white noise (a series with zero mean, constant variance and zero correlation), it indicates that all the information present in the data was successfully captured by the model. The ACF and PACF plots and the histogram of residuals were plotted to reinforce the conclusion obtained by observing the residual plots. If the residual analysis results are satisfactory, then the selected final model is adopted, else a different model is chosen, and the same steps were iterated.

6. LSTM

Artificial Neural Networks (ANN) are a set of computing units that simulate how human brains analyse and process information. They have self-learning capabilities, i.e., there is no need to programme everything. Recurrent Neural Networks (RNN) are a type of neural network suitable for processing sequential data such as time series, language, and speech. However, these RNNs have a drawback. Though in principle, RNNs are capable of learning long term dependencies, in practice, they cannot pick up these due to the vanishing gradient problem (Hochreiter, 1998).

LSTM is a special type of RNN, proposed to solve this problem of long-term dependency (Hochreiter and Schmidhuber, 1997; Hochreiter and Schmidhuber, 1997) and the vanishing gradient problem. The initial version of the LSTM block included (possibly multiple) cells and input and output gates, but no forget gate and no peephole connections (Greff et al., 2016). Later forget gate was introduced into LSTM architecture, enabling the LSTM to reset itself at appropriate times (Gers et al., 2000). All RNNs, including LSTMs, have a chain of repeating modules of neural networks. The repeating module in standard RNNs will be a simple structure containing a single layer, such as a single *tanh* layer. However, in LSTMs, the repeating module will have four layers interacting in a very special way instead of a single neural network layer. The structure of an LSTM unit is given in Fig. 2 (Phi, 0000).

The sigmoid function here outputs a value between zero and one for all the input values. The output value describes the amount of information that must pass through the gate. More precisely, the value of zero indicates no information is passed through, and the value of one indicates everything is passed. As previously mentioned, the cell state is controlled by three gate layers, namely forget gate layer, input gate layer and output gate layer. The forget gate layer decides how much of the previous cell state should be kept. The previous hidden state and information from the current input are fed to the forget gate layer and passed through the sigmoid function present there. As described earlier, this sigmoid function outputs a value ranging from zero to one, indicating remembering completely to forgetting completely.

The previous hidden state and the information from the current input are again fed to the input gate. The input gate layer decides which values should be updated for the new cell state. Furthermore, the same previous hidden state and the information from the current input is fed to the nearby *tanh* layer too, which creates a vector of new candidate values that could be added to the cell states. The outputs from these two layers are combined using a pointwise multiplication operation. Then to update the cell state actually, the old cell state is multiplied by the output of the forget gate layer and added with the combined output from the input gate layer and the nearby *tanh* layer. Finally, the output gate layer decides what the next hidden state should be. Again, the previous hidden state and the information from the current input are fed to the output gate, and the new (or updated) cell state obtained previously will be passed through another nearby *tanh* layer. The outputs from these two layers are combined using a pointwise multiplication operation to obtain the new hidden state as output.

As one can observe, the functioning of LSTM is briefly explained here

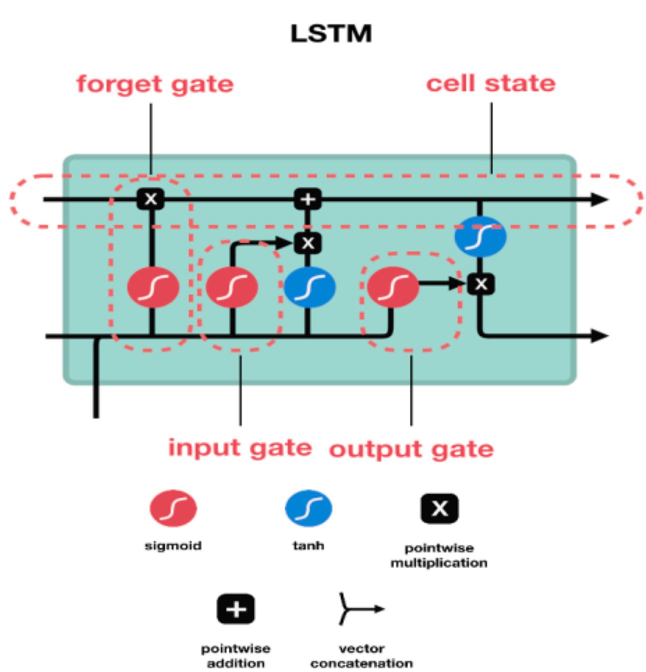

Fig. 2. Structure of an LSTM unit.

without any mathematical representations or in-depth operations, as they exceed the scope of this paper. Chris Olah excellently explained the functioning of LSTMs in detail in an article ‘Understanding LSTM Networks’ posted in his blog ‘colah’s blog’ (Olah, 2015).

In this work, the same passenger data that was used for forecasting using seasonal ARIMA earlier was used to forecast with LSTM. The training and testing sets used were also identical for both cases. The LSTM was implemented here using Anaconda Python distribution with Python 3.8.5, and the IDE (Integrated Development Environment) used was Jupyter Notebook 6.1.4 with TensorFlow (Abadi et al., 2016) and Keras (Chollet, F.: Keras, , 2015) APIs (Application Program Interfaces). The programs were run on the GTX 1050 graphics with 4 Gigabyte of video memory. Initially, the exact same time-series used for forecasting with seasonal ARIMA was used for LSTM. However, it was then slightly modified to suit the needs of LSTM. The standardisation of the dataset was carried out as some activation functions are sensitive to the magnitude of numbers. Different standardisation functions (from scikit-learn API (Pedregosa et al., 2011) such as MinMaxScaler, Quantile Transformer, StandardScaler were used, and StandardScaler was chosen. Then, time-series generator was used to organise the training data into a suitable format. A look-back of the last 15 days and a batch size of 16 was specified. A stacked LSTM model with two LSTM layers each consisting of 100 units and a dense layer with a single node, was used. The activation function used was *tanh*, and a learning rate of 0.0004 was specified with adam optimiser. The model was trained with mean absolute error as a loss function for 75 epochs. All these parameters and hyper-parameters were fixed based on trial-and-error procedure. Additionally, the number of epochs v/s loss function graph was also used to decide the number of epochs.

The trained LSTM model was made to forecast in the testing period the same as that of seasonal ARIMA (from 12th February 2020 to 29th February 2020). The forecasted values were then transformed to the original scale using inverse transforming functions. The obtained values were then compared with the actual values in the testing set. Finally, MAE and RMSE (defined and explained in the previous section) were calculated.

7. Results and discussions

The forecasting was also done using the seasonal naïve method, which is used for the seasonal data. This method was used as a control and as a benchmark in this work. Here, each forecasted value is equal to the last observed value from the same season. For example, in this case, as the data shows daily seasonality, the forecasted passenger demand for tomorrow 06:00 is the same as today’s passenger count at 06:00. Mathematically, the forecast for any time, $T+h$ is given by,

$$\widehat{y}_{\mathrm{T+h|T}} = y_{T+h-m(k+1)} \tag{7}$$

where, $\widehat{y}_{\mathrm{T+h|T}}$ denotes the forecast of y_{T+h} using the historical data $y_1, \cdots, y_T$, m is the periodicity of the season, and k is the integer part of $(h-1)/m$.

The training and testing datasets used for this method were the same as that were used for forecasting with seasonal ARIMA and LSTM. The forecasts for the testing period (12th February 2020 to 29th February 2020) will be equal to the values of 11th February 2020. After obtaining the forecasted values, MAE and RMSE were calculated. The forecasting performance, in the form of MAE and RMSE, of seasonal ARIMA, LSTM and seasonal naïve method for all 20 time-series, were tabulated in tables which are reproduced here as Table 5 (MAE) and Table 6 (RMSE) and selected Seasonal ARIMA models for all the time series are given in Table 7. However, MAE and RMSE are scale-dependent, i.e., the magnitude of MAE and RMSE depend on the magnitude of the actual and forecasted values. For example, the MAE of 10 is excellent if the numbers involved are in the magnitude of 1000, but the same MAE of 10 is by no means excellent if the values involved are of single digits. So, it would be

Table 5
Summary Of Results (MAE).

Sl. No	Station No.	Time-frame	MAE for Seasonal ARIMA	MAE for LSTM	MAE for Seasonal Naïve	Mean of Passenger	MAE % for Seasonal ARIMA	MAE % for LSTM	MAE % for Seasonal Naïve	% Reduction in MAE % with LSTM*
1.	5	60	96.21	63.24	111.35	707.85	13.59	8.93	15.73	34.27
2.	28	60	48.41	34.75	53.56	159.68	30.32	21.76	33.54	28.22
3.	33	60	36.86	32.06	36.93	164.80	22.37	19.45	22.41	13.02
4.	34	60	70.76	50.81	90.38	460.04	15.38	11.04	19.65	28.19
5.	35	60	49.83	32.27	56.93	275.43	18.09	11.72	20.67	35.24
6.	5	45	74.43	53.92	87.52	523.32	14.22	10.30	16.72	27.56
7.	28	45	37.30	25.99	43.26	118.02	31.60	22.02	36.65	30.32
8.	33	45	28.37	23.05	30.98	117.49	24.15	19.62	26.36	18.75
9.	34	45	54.82	40.90	69.99	340.04	16.12	12.03	20.58	25.39
10.	35	45	39.03	30.79	45.01	203.58	19.17	15.12	22.11	21.11
11.	5	30	51.88	38.61	62.43	353.92	14.66	10.91	17.64	25.58
12.	28	30	26.42	19.02	30.63	79.84	33.09	23.82	38.36	28.01
13.	33	30	21.29	17.39	23.26	79.43	26.80	21.89	29.29	18.32
14.	34	30	39.04	30.67	50.95	230.02	16.97	13.33	22.15	21.44
15.	35	30	29.63	25.95	34.07	137.71	21.52	18.84	24.74	12.42
16.	5	15	36.37	22.37	36.41	176.96	20.55	12.64	20.57	38.49
17.	28	15	19.82	9.70	17.06	39.92	49.65	24.30	42.73	51.06
18.	33	15	14.54	10.19	14.57	39.71	36.62	25.66	36.68	29.92
19.	34	15	28.64	16.86	28.64	115.01	24.90	14.66	24.91	41.13
20.	35	15	23.36	16.18	23.41	68.85	33.93	23.50	34.01	30.74

$MAE\% = 100 \times \frac{MAE}{MeanofPassengers}$

*This column indicates the percentage reduction in MAE % of LSTM with respect to MAE % of Seasonal ARIMA.

Table 6
Summary of results (RMSE).

Sl. No.	Station No.	Time-frame	RMSE for Seasonal ARIMA	RMSE for LSTM	RMSE for Seasonal Naïve	Mean Passengers	RMSE % for Seasonal ARIMA	RMSE % for LSTM	RMSE % for Seasonal Naïve	% Reduction in RMSE% with LSTM*
1.	5	60	145.78	80.14	164.44	707.85	20.59	11.32	23.23	45.03
2.	28	60	79.6	54.74	81.44	159.68	49.85	34.28	51.01	31.23
3.	33	60	48.09	41.07	49.07	164.80	29.18	24.92	29.77	14.60
4.	34	60	106.16	65.88	141.04	460.04	23.08	14.32	30.66	37.94
5.	35	60	73.86	44.48	82.90	275.43	26.82	16.15	30.10	39.78
6.	5	45	111.25	75.7	129.06	523.32	21.26	14.47	24.66	31.96
7.	28	45	61.02	44.27	65.34	118.02	51.70	37.51	55.36	27.45
8.	33	45	37.32	29.58	40.18	117.49	31.76	25.18	34.20	20.74
9.	34	45	82.79	57.88	109.30	340.04	24.35	17.02	32.14	30.09
10.	35	45	57.08	40.92	65.21	203.58	28.04	20.10	32.03	28.31
11.	5	30	73.15	49.57	89.90	353.92	20.67	14.01	25.40	32.24
12.	28	30	42.99	30.07	46.39	79.84	53.85	37.66	58.11	30.05
13.	33	30	27.97	22.39	30.08	79.43	35.21	28.19	37.87	19.95
14.	34	30	58.24	40.23	77.16	230.02	25.32	17.49	33.55	30.92
15.	35	30	41.5	34.24	48.41	137.71	30.14	24.86	35.16	17.49
16.	5	15	51.55	29.57	51.42	176.96	29.13	16.71	29.06	42.64
17.	28	15	25.74	15.61	25.76	39.92	64.48	39.10	64.54	39.36
18.	33	15	18.77	13.49	18.77	39.71	47.27	33.97	47.27	28.13
19.	34	15	41.95	23.29	41.94	115.01	36.48	20.25	36.47	44.48
20.	35	15	31.91	21.85	31.87	68.85	46.35	31.74	46.29	31.53

$RMSE\% = 100 \times \frac{RMSE}{MeanofPassengers}$

*This column indicates the percentage reduction in RMSE % of LSTM with respect to RMSE % of Seasonal ARIMA.

invalid to make a comparison with different stations or different time intervals. Hence, the MAE and RMSE values were divided by the mean values of respective testing datasets and converted into percentages so that all values are on the same scale. The new values obtained are now represented as MAE % and RMSE % in Table 5 and Table 6, respectively. The mean values for the respective testing data sets are also given in Table 5 and Table 6.

The comparison of forecasting accuracy of seasonal ARIMA, LSTM and seasonal naïve method can be made by comparing the MAE % columns. Moreover, it can be clearly observed in the graph given in Fig. 3 that LSTM is more efficient than the seasonal ARIMA and seasonal naïve method. (In the graph, the x-axis is the 'Sl No.' column of Table 5.) The same comparison can also be made using RMSE % values instead of MAE % for further validation. The graph given in Fig. 4 making such comparison shows that even based on RMSE %, LSTM outperforms seasonal ARIMA and seasonal naïve methods. (In the graph, the x-axis is the 'Sl No.' column of Table 6.).

Therefore, it is clear that, by all measures and for all time frames,

Table 7
Selected seasonal ARIMA models for all the time-series.

Station No.	Time-Frame	Selected Model	AIC	BIC
5	60	SARIMA (5,1,1) × (1,1,1,17)	14861.32	14907.30
28	60	SARIMA (1,1,4) × (2,1,1,17)	13502.09	13548.08
33	60	SARIMA (2,1,4) × (0,1,1,17)	12638.17	12663.72
34	60	SARIMA (1,1,3) × (2,1,1,17)	14470.00	14510.87
35	60	SARIMA (3,1,2) × (1,1,2,17)	13509.34	13555.32
5	45	SARIMA (2,1,2) × (1,1,1,23)	19169.20	19207.09
28	45	SARIMA (4,1,4) × (2,1,1,23)	17313.39	17372.92
33	45	SARIMA (7,1,3) × (0,1,1,23)	16309.37	16379.72
34	45	SARIMA (2,1,2) × (2,1,1,23)	18582.00	18619.88
35	45	SARIMA (11,1,1) × (1,1,1,23)	17488.17	17569.35
5	30	SARIMA (4,1,2) × (1,1,1,34)	26607.16	26659.38
28	30	SARIMA (2,1,2) × (2,1,1,34)	23709.00	23749.62
33	30	SARIMA (11,1,4) × (0,1,1,34)	22487.84	22592.28
34	30	SARIMA (3,1,2) × (2,1,1,34)	25654.34	25700.76
35	30	SARIMA (1,1,3) × (1,1,1,34)	24668.94	24709.56
5	15	SARIMA (1,1,1) × (1,1,1,68)	35251.08	35334.21
28	15	SARIMA (1,1,1) × (2,1,1,68)	31934.17	32002.38
33	15	SARIMA (1,1,1) × (0,1,1,68)	30945.00	30985.00
34	15	SARIMA (4,1,1) × (2,1,1,68)	35638.58	35714.14
35	15	SARIMA (7,1,2) × (1,1,1,68)	34224.34	34341.15

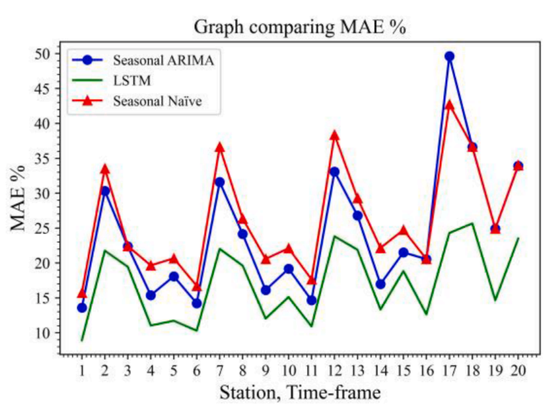

Fig. 3. Graph comparing MAE %.

LSTM is better than seasonal ARIMA. This can be attributed to the fact that LSTM can successfully learn long term dependencies and also non-linear relationships. The percentage reduction in both MAE % and RMSE % are given in the last columns of Table 5 and Table 6, respectively. They are also represented as graphs for easy visualisation and is given here in Fig. 5. In summary, the MAE % was reduced by 27.46 %, and the RMSE % was reduced by 31.08 %. (In the graph, the x-axis is the 'Sl No.' column of Table 5 and Table 6).

One more interesting result can be obtained by Table 5 and Table 6.

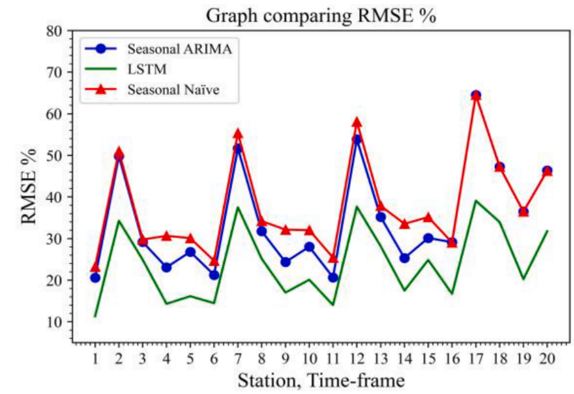

Fig. 4. Graph comparing RMSE %.

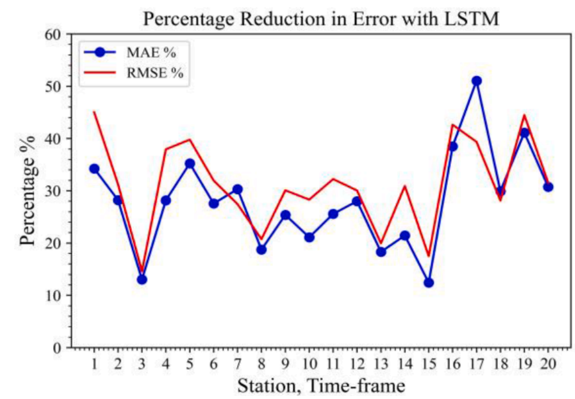

Fig. 5. Percentage reduction in error with LSTM.

The tables can be used to compare how the resampling time interval affects the forecasting accuracy. Firstly, considering MAE %, for LSTM, for all stations, it can be observed 60 min time interval gives better results than 15 min, 30 min and 45 min time intervals. For the visualisation purpose, the graph of this comparison is given in Fig. 6. Similar results can be obtained with RMSE % for LSTM and also with MAE % and RMSE % for seasonal ARIMA. By this, it can be confirmed that the time-series resampled with 60-minute intervals performed better than other time intervals considered.

8. Conclusions

- One of the best and innovative ways to solve the present problems of the transportation industry, such as congestion, contribution to

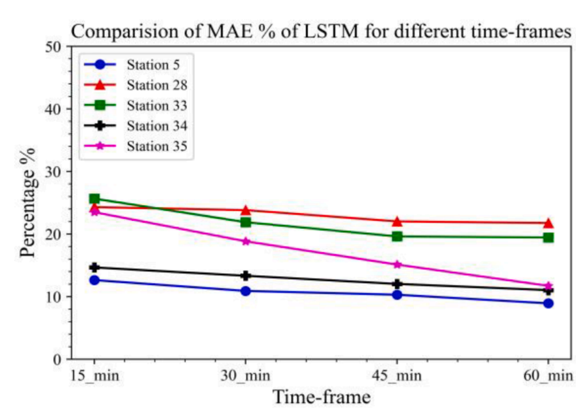

Fig. 6. Comparison of MAE % of LSTM for different time-frames.

environmental pollution, and accidents leading to an inefficient transportation system, is to implement the concept of smart mobility through intelligent transportation system (ITS).

- Moreover, passenger demand forecasting is an unavoidable procedure in establishing and maintaining the ITS. Passenger demand forecasting can be done using both traditional methods and more advanced modern methods using deep learning. With the growing computation power of modern computer processors and the availability of a large amount of accurate, systematic and reliable data, in addition to the constant efforts in producing forecasts with greater accuracy, the modern, more advanced methods of forecasting using deep learning are gaining more popularity.
- However, research works on the matter of passenger demand forecasting using LSTM and other deep learning techniques are quite limited, considering the Indian context. Meanwhile passengers demand forecasting using APC data with conventional techniques or deep learning-based techniques is also limited in Indian context. It can be argued that the reason for this could be the non-availability of such suitable data and LSTM being a relatively new technique.
- Hence, in this research work, an attempt was made to explore the unexplored area of LSTM in passenger demand forecasting, in the Indian context. Also, there is a confusion about the resampling time-interval of the time series to get the best forecasted results and studies on addressing such area are less in the past literature. This work, explored the time series results, resampled with different time intervals, to find the most suitable time interval which gives the best forecasting results. Along with contributing towards solving the problems of the Indian transportation industry, this work will also serve as a future reference for studies carried out in this field.
- As a part of it, the top-5 stations with the highest passenger flow were selected from the 35 stations of the HDBRTS. Then 20 time-series were formed with data resampled into four-time intervals of 15 min, 30 min, 45 min and 60 min for the top-5 stations after removing the additional information which was not needed. Subsequently, the times-series graphs were plotted and were observed for the various time series patterns. Then, the passenger demand was forecasted using LSTM with the APC data of three months obtained from the HDBRTS. Furthermore, seasonal ARIMA was also used to forecast passenger demand, and a comparison of the forecasting accuracy was made, and the seasonal naïve method was used as a benchmarking method.
- The results showed that using LSTM, the MAE % can be reduced by 27.46 % with respect to the MAE % of seasonal ARIMA, and RMSE % can be reduced by 31.08 % with respect to the RMSE % of seasonal ARIMA. It was found that the time-series resampled with 60 min intervals gave better results than the 15 min, 30 min and 45 min intervals.
- SARIMA is a traditional linear model which fails to describe the stochastic and non-linear nature of passenger demand. In such a state, deep learning-based models play their role effectively on passenger demand forecasting. These models will have good outcomes on spatial and temporal evolution of passenger demand flow. LSTM, is one of such deep learning-based models, which captures the characteristics of time series, combines underlying features and works on all modelling issues mentioned above efficiently. However, LSTM gives reliable results with larger datasets and also requires more time. Nevertheless, if the necessary resources are available, one can always go with deep learning-based model LSTM more effectively over traditional methods of forecasting.
- Finally, with this research work, it can be concluded that LSTM can be satisfactorily used to forecast passenger demand with data obtained by APC in the Indian scenario. The basic version of LSTM was used for forecasting in this study but the research can be continued to understand and evaluate the forecasting accuracy of different variants of LSTMs and Gated Recurrent Units (GRU's) as a scope of future work.

Authors contribution

First author Shivaraj Halyal was responsible for the data gathering, data cleaning, analysing and mining the data, finally interpreting the obtained results and presenting it in the form of research article. Second author Raviraj H. Mulangi and third author Harsh M.M were responsible for data collection, interpreting the obtained results and drafting the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Authors acknowledge and thank the Hubli-Dharwad Bus Rapid Transit System Company Limited (HDBRTS) and the North Western Karnataka Road Transport Corporation (NWKRTC), Hubli for their kind help in providing the data and all the support in the current research work.

References

Sources of Greenhouse Gas Emissions. US EPA. https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions. Accessed 25 July 2021.

Sharma, R.D., Jain, S., Singh, K., 2011. Growth rate of motor vehicles in india-impact of demographic and economic development. J. Econ. Soc. Stud. (JECOSS) 1 (1), 137–153.

Overview of Road Accidents in India. prsindia.org. https://prsindia.org/policy/vital-stats/overview-road-accidents-india. Accessed 25 July 2021.

Benevolo, C., Dameri, R.P., D'auria, B., 2011. Smart mobility in smart city. In *Empowering organisations*. Springer, Cham, pp. 13–28.

Šemanjski, I., Mandžuka, S., Gautama, S., 2018. Smart mobility. In: 2018 International Symposium ELMAR. IEEE, pp. 63–66 https://doi.org/10.23919/ELMAR.2018.8534693.

Satheesh Kumar, M., & Mali, R. Y. Applications of Intelligent Transportation Systems using RFID Systems.

Nguyen, N.T., Miwa, T., Morikawa, T., 2020. Demand forecast of public transportation considering positive and negative mass effects. Transport. Res. Part D: Transp. Environ. 85, 102466 https://doi.org/10.1016/j.trd.2020.102466.

Banerjee, N., Morton, A., Akartunalı, K., 2020. Passenger demand forecasting in scheduled transportation. Eur. J. Oper. Res. 286 (3), 797–810. https://doi.org/10.1016/J.EJOR.2019.10.032.

Cyril, A., Mulangi, R.H., George, V., 2018. Modelling and forecasting bus passenger demand using time series method. In: 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE, pp. 460–466, 10.1109/ICRITO.2018.8748443.

Cyril, A., Mulangi, R.H., George, V., 2019. Bus passenger demand modelling using time-series techniquesbig data analytics. Open Transport. J. 13 (1), 41–47.

Cyprich, O., Konečný, V., Kilianova, K., 2013. Short-term passenger demand forecasting using univariate time series theory. PROMET-Traffic Transport. 25 (6), 533–541. https://doi.org/10.7307/PTT.V25I6.338.

Anvari, S., Tuna, S., Canci, M., Turkay, M., 2016. Automated Box–Jenkins's forecasting tool with an application for passenger demand in urban rail systems. J. Adv. Transport. 50 (1), 25–49. https://doi.org/10.1002/ATR.1332.

Milenković, M., Švadlenka, L., Melichar, V., Bojović, N., Avramović, Z., 2018. SARIMA modelling approach for railway passenger flow forecasting. Transport 33 (5), 1113–1120. https://doi.org/10.3846/16484142.2016.1139623.

Gummadi, R., Edara, S.R., 2018. Flow prediction of transit buses along a route based on time series. In: Analysis of Passenger Flow Prediction of Transit Buses Along a Route Based on Time Series. Springer, Singapore, pp. 31–37, 10.1007/978-981-10-7563-6_4.

Ye, Y., Chen, L., Xue, F., 2019. Passenger flow prediction in bus transportation system using ARIMA models with big data. In: 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, pp. 436–443. https://doi.org/10.1109/CYBERC.2019.00081.

Kumar, S.V., Vanajakshi, L., 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. Eur. Transp. Res. Rev. 7 (3), 1–9. https://doi.org/10.1007/S12544-015-0170-8.

Rabbani, M.B.A., Musarat, M.A., Alaloul, W.S., Rabbani, M.S., Maqsoom, A., Ayub, S., Bukhari, H., Altaf, M., 2021. A comparison between seasonal autoregressive integrated moving average (SARIMA) and exponential smoothing (ES) based on time series model for forecasting road accidents. Arab. J. Sci. Eng. 46 (11), 11113–11138.

Xue, R., Sun, D.(., Chen, S., 2015. Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. Discr. Dyn. Nat. Soc. 2015, 1–11.

Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: principles and practice. OTexts, Melbourne, Australia https://otexts.com/fpp2.
Li, J., Huang, P., Yang, Y., & Peng, Q.: Passenger Flow Prediction of High-Speed Railway Based on LSTM Deep Neural Network. In *Rail Norrkoping 2019. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA), Norrköping, Sweden, June 17th–20th, 2019* (No. 069, pp. 723-739). Linköping University Electronic Press (2019).
Ouyang, Q., Lv, Y., Ma, J., Li, J., 2020. An LSTM-based method considering history and real-time data for passenger flow prediction. Appl. Sci. 10 (11), 3788. https://doi.org/10.3390/APP10113788.
Gallo, M., De Luca, G., D'Acierno, L., Botte, M., 2019. Artificial neural networks for forecasting passenger flows on metro lines. Sensors 19 (15), 3424. https://doi.org/10.3390/S19153424.
Xiong, Z., Zheng, J., Song, D., Zhong, S., Huang, Q., 2019. Passenger flow prediction of urban rail transit based on deep learning methods. Smart Cities 2 (3), 371–387. https://doi.org/10.3390/SMARTCITIES2030023.
Han, Y., Wang, C., Ren, Y., Wang, S., Zheng, H., Chen, G., 2019. Short-term prediction of bus passenger flow based on a hybrid optimised LSTM network. ISPRS Int. J. Geo-Inf. 8 (9), 366. https://doi.org/10.3390/IJGI8090366.
Farahani, M., Farahani, M., Manthouri, M., & Kaynak, O.: Short-term traffic flow prediction using variational LSTM networks. (2020). *arXiv preprint arXiv: 2002.07922*.
Yang, D., Chen, K., Yang, M., Zhao, X., 2019. Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features. IET Intel. Transport Syst. 13 (10), 1475–1482. https://doi.org/10.1049/IET-ITS.2018.5511.
Shahriari, S., Ghasri, M., Sisson, S.A., Rashidi, T., 2020. Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction. Transport. A: Transport Sci. 16 (3), 1552–1573. https://doi.org/10.1080/23249935.2020.1764662.
Tourangeau, R., Zimowski, M., Ghadialy, R., 1997. An introduction to panel surveys in transportation studies. National Opinion Research Center.
Hummer, J. E.: Public transportation studies. (1994).
Zheng, X., Chen, W., Wang, P.u., Shen, D., Chen, S., Wang, X., Zhang, Q., Yang, L., 2016. Big data for social transportation. IEEE Trans. Intell. Transp. Syst. 17 (3), 620–630.
Du, J., Barth, M.J., 2008. Next-generation automated vehicle location systems: Positioning at the lane level. IEEE Trans. Intell. Transp. Syst. 9 (1), 48–57. https://doi.org/10.1109/TITS.2007.908141.
Alfred Chu, K.K., Chapleau, R., Trepanier, M., 2009. Driver-assisted bus interview: Passive transit travel survey with smart card automatic fare collection system and applications. Transp. Res. Rec. 2105 (1), 1–10. https://doi.org/10.3141/2105-01.
Jia, F., Li, H., Jiang, X., Xu, X., 2019. Deep learning-based hybrid model for short-term subway passenger flow prediction using automatic fare collection data. IET Intel. Transport Syst. 13 (11), 1708–1716. https://doi.org/10.1049/IET-ITS.2019.0112.
Kwak, S.K., Kim, J.H., 2017. Statistical data preparation: management of missing values and outliers. Korean journal of anesthesiology 70 (4), 407. https://doi.org/10.4097/KJAE.2017.70.4.407.
Dhakal, C.P., 2017. Dealing with outliers and influential points while fitting regression. Journal of Institute of Science and Technology 22 (1), 61–65. https://doi.org/10.3126/JIST.V22I1.17741.
Medar, R., Rajpurohit, V.S., Rashmi, B., 2017. Impact of training and testing Data splits on accuracy of time series forecasting in Machine Learning. In: 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA). IEEE, pp. 1–6. https://doi.org/10.1109/ICCUBEA.2017.8463779.
Phillips, P.C., Perron, P., 1988. Testing for a unit root in time series regression. Biometrika 75 (2), 335–346. https://doi.org/10.1093/BIOMET/75.2.335.
Dickey, D.A., Fuller, W.A., 1979. Distribution of the estimators for autoregressive time series with a unit root. J. Am. Stat. Assoc. 74 (366a), 427–431. https://doi.org/10.1080/01621459.1979.10482531.
Dickey, D.A., Pantula, S.G., 1987. Determining the order of differencing in autoregressive processes. J. Business Econ. Stat. 5 (4), 455–461. https://doi.org/10.1198/073500102753410363.
Kuha, J., 2004. AIC and BIC: Comparisons of assumptions and performance. Sociol. Methods Res. 33 (2), 188–229. https://doi.org/10.1177/0049124103262065.
Burnham, K.P., Anderson, D.R., 2004. Multimodal inference: understanding AIC and BIC in model selection. Sociol. Methods Res. 33 (2), 261–304. https://doi.org/10.1177/0049124103262065.
Vandeput, N., 2021. Data science for supply chain forecasting. De Gruyter.
Ding, J., Tarokh, V., Yang, Y., 2018. Model selection techniques: An overview. IEEE Signal Process Mag. 35 (6), 16–34. https://doi.org/10.1109/MSP.2018.2867638.
Hochreiter, S., 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. Internat J. Uncertain. Fuzziness Knowledge-Based Syst. 6 (02), 107–116. https://doi.org/10.1142/S0218488598000094.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735.
Hochreiter, S., Schmidhuber, J., 1997. LSTM can solve hard long time lag problems. Adv. Neural Inform. Process. Syst. 473–479.
Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2016. LSTM: A search space odyssey. IEEE Trans. Neural Networks Learn. Syst. 28 (10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924.
Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. Neural Comput. 12 (10), 2451–2471. https://doi.org/10.1162/089976600300015015.
Phi, M.: Illustrated Guide to LSTM's and GRU's: A step by step explanation. Towards Data Science. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. Accessed 25 July 2021.
Olah, C.: Understanding LSTM Networks. colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs (2015). Accessed 25 July 2021.
Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X.: Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283) (2016). https://doi.org/10.5555/3026877.3026899.
Chollet, F.: Keras: Deep learning library for theano and tensorflow. *URL: https://keras.io/k*, *7*(8), T1 (2015).
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Machine Learn. Res. 12, 2825–2830.