Received: 25 November 2021 | Revised: 23 February 2022 | Accepted: 14 March 2022

IET Smart Cities

DOI: 10.1049/smc2.12027

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Traffic speed prediction of high-frequency time series using additively decomposed components as features

Muhammad Ali[1] | **Kamaludin Mohamad Yusof**[1] | **Benjamin Wilson**[2] | **Carina Ziegelmueller**[3]

[1]School of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

[2]HERE Technologies, Melbourne, Victoria, Australia

[3]Michael Bauer International GmbH, Karlsruhe, Germany

Correspondence

Muhammad Ali, School of Electrical Engineering, Universiti Teknologi Malaysia, P19a, Skudai Johor 81310, Malaysia.
Email: amuhammad32@graduate.utm.my

Abstract

Traffic speed prediction is an integral part of an Intelligent Transportation System (ITS) and the Internet of Vehicles (IoV). Advanced knowledge of average traffic speed can help take proactive preventive steps to avoid impending problems. There have been studies for traffic speed prediction in which data has been decomposed into components using various decomposition techniques such as empirical mode decomposition, wavelets, and seasonal decomposition. As far as the authors are aware, no research has used additively decomposed components as input features. In this study, we used additive decomposition on 21,843 samples of traffic speed data. We implemented two statistical techniques designed for double seasonality (i) Double Seasonal Holt-Winter, and (ii) Trigonometric seasonality, Box-Cox transformation, autoregressive integrated moving average errors, trend, and Seasonal components (TBATS), and five machine learning (ML) techniques, (i) Multi-Layer Perceptron, (ii) Convolutional-Neural Network, (iii) Long Short-Term Memory, (iv) Gated Recurrent Unit and (v) Convolutional-Neural Network-LSTM. Machine learning techniques are used in univariate mode with raw time series as features and then with decomposed components as features in multivariate mode. This study demonstrates that using decomposed components (trend, seasonal, and residual), as features, improves prediction results for multivariate ML techniques. This becomes a significant advantage when no other features are available.

KEYWORDS

additive decomposition, GRU, intelligent transport systems, multivariate models, traffic speed predictions, TBATS

1 | INTRODUCTION

Traffic speed prediction is an integral part of an Intelligent Transportation Systems (ITS) and Internet of Vehicles (IoV); an IoV is comprised of a combination of a vehicle to vehicle, vehicle to infra, vehicle to roadside units, and vehicle to pedestrian communication disseminating safety messages using wireless links [1]. A prior estimate of impending traffic conditions greatly helps both of these systems for smooth, safe, and unhindered operation. It also helps in trip planning, route determination, and traffic management.

Modern techniques such as GPS probes, aerial photography, cameras installed on top of tall buildings, or specially equipped vehicles can all be used to measure vehicular speed [2]. These measurements result in a time series, which is a series of temporal observations taken at regular intervals. Time series prediction can be accomplished using a variety of techniques. However, before employing any technique, the data should be thoroughly examined to identify a suitable model compatible with the dataset. The decomposition of time series is one possible approach. At high frequency, the long data set may reveal multiple seasonalities along with trend and residuals.

Two of the major time series forecasting techniques, inheriting their power from statistics, are exponential smoothing [3] and autoregression (AR) [4]. These techniques exploit the statistical features of observed data to fit a model into the current value minimising some cost functions such as root-mean-square-error (RMSE). These techniques and their extensions, such as Holt-Winter (HW) and auto regressive moving average (ARMA), assume stationarity and linearity in the underlying series. If the series is non-linear and volatile (having varying variance), then deep learning architectures are expected to give better results. Nonetheless, there are evidences that favour statistical algorithms like triple exponential smoothing or its variants [5]. In contrast, a vast class of literature demonstrates improved performance with machine learning (ML) techniques such as multi-layer perceptron (MLP) [6], long short-term memory (LSTM) [7], and their hybrids [8].

Researchers demonstrate that neural networks can comfortably handle non-linearity with improved performance [9, 10, 11]. Following alternative footsteps, some researchers have decomposed time series using many ways such as seasonal and trend decomposition using Loess (STL), empirical mode decomposition (EMD), and additive/and multiplicative decompositions. Additive decomposition along with echo state network (ESN) is used in [12], STL decomposition is used in [13], and decomposed components are applied to HW and artificial neural network (ANN) models. Decomposed data is mostly modelled using variants of recurring networks. A combination of two decomposition methods is applied to LSTM networks in ref.[14]. In these approaches key elements are splitting data into deterministic and random parts and exposing each to a suitable model. Modelling the deterministic part seems trivial for many models; however, it is not ascertained by the researchers that whether statistical or ML approach is preferable for random part. Secondly, data with multiple seasonality is not explicitly modelled in these research.

In the underlying study, we looked for a suitable model to make a one-step prediction using long and high-frequency data consisting of 21,843 samples taken every 5-min interval. The data are obtained from Here™ Technologies. The consequence of the high frequency observations is the emergence of daily seasonality and long length of data signifies weekly seasonality. The data are very reliable with only 45 missing values and confidence factor of 40 as claimed by the provider's document. We identified two statistical models which are specifically suitable for double seasonality data, that is, double seasonal Holt-Winter (DSHW) and trigonometric seasonality, Box-Cox transformation, autoregressive integrated moving average (ARIMA) errors, trend and seasonal components (TBATS). Further, five ML models (MLP) [6] (LSTM) [15, 16], gated recurrent unit (GRU) [17, 18], convolutional-neural networks (CNN) [19, 20] and CNN-LSTM [21, 22] were also implemented. Seasonal autoregressive integrated moving average (SARIMA) was used as the benchmark. The ML models were used in univariate and multivariate modes. For multivariate models we first additively decomposed time series into its trend, seasonal and residual components with a view to use them as features (independent variables). The motivation for this approach is derived from the fact that decomposed components are independent (having minimal correlation) and correlated with the predicted variable (mean-speed). The results obtained for three selected highway segments demonstrate that this approach gives improvement for all machine leaning models. The contribution of this research becomes more significant in the case where there are no other features available for a similar kind of data. To the best of our knowledge no research has used additively decomposed components as features in multivariate ML models.

The rest of the paper is organised as follows. Section 2 presents a review of related literature, and Section 3 describes the methodology used, followed by the results given in Section 4. Finally, Section 5 concludes the paper.

2 | RELATED WORK

The literature on time series analysis is rich and extensive, covering many dimensions and domain-specific approaches. In the following, however, we cover a limited set of relevant samples for short-term traffic speed/flow prediction.

Most of the research on time series is inherited from averaging and regression analysis from statistics. The researchers have extensively used smoothing and AR techniques such as HW and variants of ARIMA. However, due to the huge impact of ML, techniques like MLP, LSTM, GRU, CNN, and their hybrids are also used.

Broad coverage of data models and algorithms used for smart transport planning was presented in ref.[23]. The researchers have focussed on the mode of communication (e.g. connected vehicles) and data collection methods, emphasising the need for more accurate data sources. They also provided a hybrid technique that combines data from two sources using STL decomposition and CNN techniques. However, the table comparing a few techniques claiming ARIMA requires a large amount of historical data appears contradictory.

Various statistical and ML models for multiple horizons (5, 15, 30, and 60 min) and two aggregation levels (5 and 15 min) have been evaluated in ref.[24]. Their modelling approaches treat the periodic seasonal components and separately seasonally adjusted data with improved performance for multi-step prediction and better results for higher aggregation (15 min). They have used mean absolute error (MAE), RMSE, and mean absolute percentage error (MAPE) as performance measure criteria. One of the conclusions their research made is that multi-step prediction greatly improves with periodic information.

Double Seasonal Holt-Winter model, on hourly data from Freeway I-5 California, was applied in ref.[25]. A comparison was made with standard HW and ARIMA, showing results in favour of the DSHW technique for multi-step forecasting. The authors also came up with an improvement (in-sample MAPE improvement of 13.31%) in the basic model by including a first-order autoregression (AR(1)) term for one-step forecasting.

Seasonal autoregressive integrated moving average model and seasonal discrete grey model (SDGM) approaches, for data collected at different time intervals, were combined in ref.[26]. Seasonal discrete grey model are used in finance, engineering, control, and economics. The combination of SARIMA and SDGM in ref.[26] was compared with SARIMA, SDGM, support vector regression (SVR), and artificial neural networks (ANN) models, claiming best results for SARIMA-SDGM using MAE, MAPE, and RMSE as performance metrics.

Outcomes of M3 competition were reported in ref.[27]. The results predominantly favour eight selected statistical approaches on simulated data over ML techniques. However, the M4 competition that used real-life data reported the winner as a combination of triple exponential smoothing and LSTM used hierarchically [28]. These results are significant because they indicate a data dependency on algorithm performance and consonant with an earlier publication [29] that rigorously investigated the relation of performance with data characteristics.

An attempt is made in ref.[30] to apply SARIMA for flow prediction with a limited dataset of 3 days taken every 1 min, aggregated into 10 min. The fourth day is used for validation. The authors used a seasonal difference of 144 with multiple AR and moving averages lags. They also investigated morning and evening peak flow behaviours; all results are shown to be within acceptable limits using the MAPE metric.

The ANN is one of the popular non-parametric techniques particularly suited for modelling non-linear high-frequency data compared to statistical methods. In ref.[31], authors identified difficulties in selecting the number of input nodes for high-frequency data for a set of daily, weekly and monthly data selected from the NN5 competition. Their research signifies that although the low-frequency aggregated data gives better results, this is achieved at the cost of losing valuable information.

A performance comparison between SARIMA, radial-basis-function neural network with Gaussian function and their hybrid was made in ref.[32] using a 5-min interval traffic flow data of 14 days with 1 day ahead prediction. Their results significantly favoured the hybrid technique over the individual ones. A study in ref.[33] applied a deep belief network (DBN) to predict speed on a road segment, showing performance improvement over ARIMA and back-propagation neural networks. A DBN is made up of a stack of restricted Boltzmann machines trained to extract features from data at each hidden layer. This study made use of 2-min data collected over 3 months. A grid search was used to select model parameters. Another study [34] used a neural network with radial-basis-function to predict speed on an urban freeway, demonstrating that spatial information can improve prediction accuracy.

In another study involving traffic prediction [35], the authors used MLP, feeding 19 different features (time of day, type of vehicles, etc.) into the network, one hidden layer with six neurons and one output to predict the flow. Data had 15 min intervals, and only 45-min data was used to predict a 15-min horizon. An important aspect of their research is that they have predicted the speed of every type of vehicle separately. Their work claims to have consistent performance for varied time intervals.

A sequential combination of CNN and LSTM for speed prediction and extracting spatio-temporal features was suggested in ref.[36]. Convolutional-Neural Network was first used to extract daily and weekly periodicity before being combined with spatio-temporal features. The results were compared with SVR, MLP, Lasso, Random Forest, and LSTM, which better prediction for prediction horizons ranging from 15 to 90 min.

A distributed, adaptive and customised LSTM was used in ref.[37] for a large-scale transportation network targeting fine-grained and accurate speed prediction. In their scheme, the nodes observing similar traffic behaviour can use a model trained for such traffic patterns, saving the computational load of the network.

Ref.[38] investigated the LSTM neural network, taking advantage of its ability to handle non-linear dynamics, remember long-term dependencies, and automatically determine optimum time-lag. Using 1-month data collected at 2-min intervals, they found that support vector machines (SVM), ARIMA, and the Kalman Filter performed better than the three neural network variants, SVM, ARIMA, and Kalman Filter, with multiple error metrics. A deep learning model for speed prediction in ref.[39] claimed a rational integration of recurrent neural network (RNN) and CNN. Their design comprised of a context learning section, a periodicity learning section, and an LSTM based time series learning section. For learning time series patterns, neighbouring traffic information was embedded in the convolution operation in the context learning section capturing the spatial information that may influence a segment of the road network.

Performance of two variants of MLP (one and five nodes input layers and three hidden layers each) was compared with a CNN, having five parallel sublayers in ref.[40]. This work emphasised that MLP cannot capture local data dependence and is prone to noise, while CNN overcomes both of these limitations, improving the prediction results for 5, 10, 15, and 30-min intervals.

Another approach generally used is to decompose a time series into its components and then model these components in various ways. One such approach is adapted in ref.[12], where additively decomposed components are applied to echo state networks (ESN) and exposed to various datasets. Results are compared with LSTM, GRU, and ESN, showing an improvement. Seasonal and trend decomposition using Loess was used in ref.[13] to separate random and predictive parts. LSTM-NN models HW models seasonal and trend components and residual parts. The results show improvement for passenger flow prediction for 15, 30, 45, and 60-min intervals. In a similar approach in ref.[14], the author used (EMD) and STL and applied decomposed components directly to single-step LSTM models. Their results favour the STL-LSTM combination compared to SVR, LSTM, and EMD-LSTM combination. The work in ref.[41] has used a combination of EMD and ARIMA for 20 s data taken for peak hours on

working days only. Whereas in our work, we have applied statistical and ML models on 10 weeks of data, including 24 h and weekends, using univariate and multivariate modes. They applied their proposed EMD-ARIMA model to 10 samples of data reporting improvement.

The literature review above shows the diversity of various statistical and ML models applications and their hybrids. It also appears that there is an inconsistency in performance measure methods making it difficult to make a direct comparison. However, this survey reveals the following facts:

- High-frequency long data poses a challenge in modelling due to the emergence of multiple seasonality components.
- Decomposing is beneficial as a pre-processing technique so that deterministic and random parts can be modelled separately.
- It is also reported that results are data specific in many cases and may not be generalised.
- To the best of our knowledge, no research has used additively decomposed data as features for multivariate machine learning models for traffic speed prediction.

3 | METHODOLOGY

3.1 | Data description and pre-processing

The data for this research were obtained from Here™ Technologies who collect and maintain GPS probe data obtained from vehicles travelling on the road through more than 100 providers worldwide. The service provider collects GPS data from a variety of sources. According to the provider, the data is sourced from various suppliers including OEM's who have SIM-connected passenger vehicles, logistics service providers, which have vehicles with telematics tracking devices and Smartphone applications providers that use GPS location.

With a frequency of 5 min, the selected dataset refers to a 1345-m-long north-bound highway segment in Berlin, highlighted in red as shown in Figure 1. The data has a consistent confidence factor of 40 indicating highest level of trust as claimed by the provider.

The study focuses on mean-speed (Mean) prediction. There are 21,843 rows of data with 45 missing values, which are back-filled and forward-filled by adjacent rows resulting in 21,888 rows corresponding to 76 days. A univariate time series was obtained by extracting the 'Mean' column from the data frame, as shown in Table 1. The other fields mentioned in the table are length which gives the total length of the highway segment, and the count is the number of vehicles used for the measurement.

The histogram and the summary statistics of the data are shown in Figure 2 and Table 2, respectively. The histogram depicts the bimodal nature of the data relating to distinct day and night behaviour of traffic.

$$\text{Additive decomposition} \quad x_t = m_t + s_t + e_t \tag{1}$$

$$\text{Multiplicative decomposition} \quad x_t = m_t * s_t * e_t \tag{2}$$

In Equations (1) and (2), time series x_t is decomposed into its trend, m_t, seasonal, s_t, and a residual, e_t components. For an ideal case, the residual is normally distributed, indicating the absence of any correlated part. The decomposition helps in selecting a suitable model for a time series. We preferred additive decomposition because seasonal variation is independent of the trend.

For this work, a multiple seasonal decomposition method (MSTL) from the R library was used. Figure 3 shows the actual data on top and decomposed components, namely, trend, seasonal288 ($12 \times 24 = 288$) for daily seasonality, seasonal2016 ($12 \times 24 \times 7 = 2016$) for weekly seasonality followed by a residual component. Daily seasonality showed day-night minima and maxima, and weekly seasonality indicated low values for working days and high values for weekends due to light traffic. The trend depicted a decaying pattern attributed to gradually increasing mean speed due to increasing business activities.

The autocorrelation function of the remainder (residual), given in Figure 4, revealed correlated daily (288) seasonal components. Hence, the residual is not white, suggesting modelling of seasonally adjusted data and seasonal data modelling.

3.2 | Description of the models

The statistical models for comparative analysis were selected based on their suitability for double seasonality data, and ML models were selected due to their applicability. In the following we give a description of selected models in each group.

3.2.1 | DSHW and TBATS

Exponential smoothing techniques are based on the principle of modelling the current observation using an exponentially weighted average of past values as illustrated in Equation (3) in the recursive form:

$$S_t = \alpha y_t + (1-\alpha) S_{t-1} \quad 0 < \alpha \leq 1 \tag{3}$$

where y_t is the current value in time series, S_t and S_{t-1} are the current and previous estimates, respectively, and α is the smoothing parameter. Extending this principle further, ref.[43] presents the DSHW model as a set of four equations as given in Equations (4)–(7).

$$\begin{aligned}\widehat{y}_t(k) = {} & l_t + d_{t-s_1+k} + w_{t-s_2+k} \\ & + \varnothing^k [y_t - (l_{t-1} + d_{t-s_1} + w_{t-s_2})]\end{aligned} \tag{4}$$

$$l_t = \lambda(y_t - d_{t-s_1} - w_{t-s_2}) + (1-\lambda) l_{t-1} \tag{5}$$

$$d_t = \delta(y_t - l_{t-s_1} - w_{t-s_2}) + (1-\delta) d_{t-s_1} \tag{6}$$

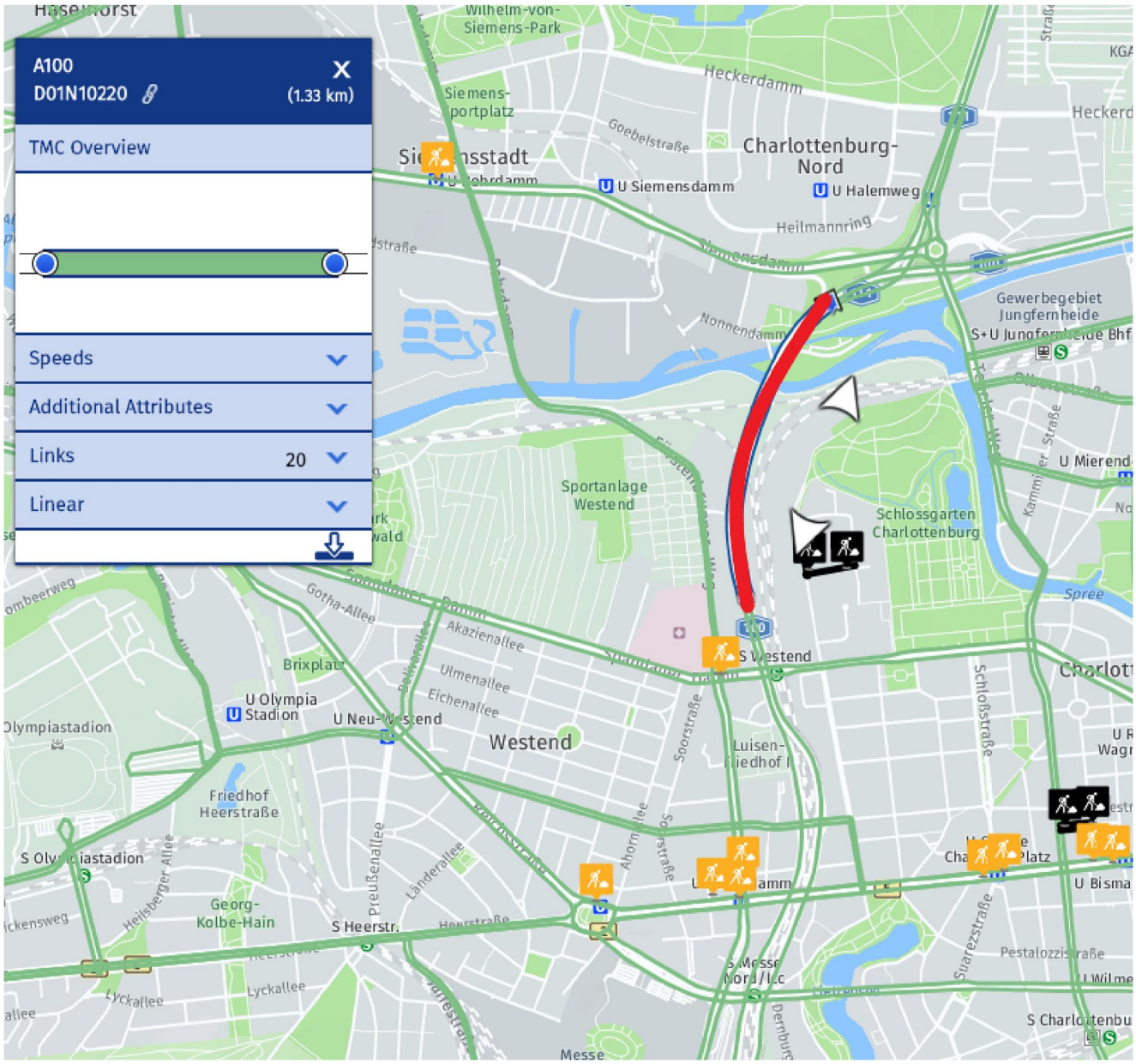

FIGURE 1 The Bundes-Autobahn A100 Highway link is shown in HERE™ Traffic Viewer

Date-time	TMC	Length	Count	Mean	Std. dev	Confidence
2019-07-03 00:00:00	D01N10220	1345	226	83.5	10.7	40
2019-07-03 00:05:00	D01N10220	1345	328	88.7	10.4	40

TABLE 1 First two rows of data frame under observation is a table

Note: TMC: Traffic Message Channel [42].

$$w_t = \omega(y_t - l_{t-s_1} - d_{t-s_1}) + (1-\omega)w_{t-s_2} \tag{7}$$

where $\hat{y}_t(k)$ is the current estimate at kth step, l_t is the level term; d_t, w_t are intraday and weekly seasonality terms, respectively, $\varnothing^k$ is the damping factor at kth step, s_1 and s_2 are the double seasonality cycles. λ, δ, and ω are the three smoothing parameters to be estimated. Also, l_{t-s_1}, d_{t-s_1} and w_{t-s_2} signifies level, daily seasonality, and weekly seasonality, respectively, delayed by seasonality cycle 1, s_1 and seasonality cycle 2, s_2.

Another model called TBATS extends the capability of the DSHW model by including Box-Cox transformations to handle non-linearity, Fourier representations to periodic model components, and ARMA representation for capturing any correlated residuals in a time series. The basic model is formulated in Equations (8)–(11),

FIGURE 2 The histogram of traffic speed data

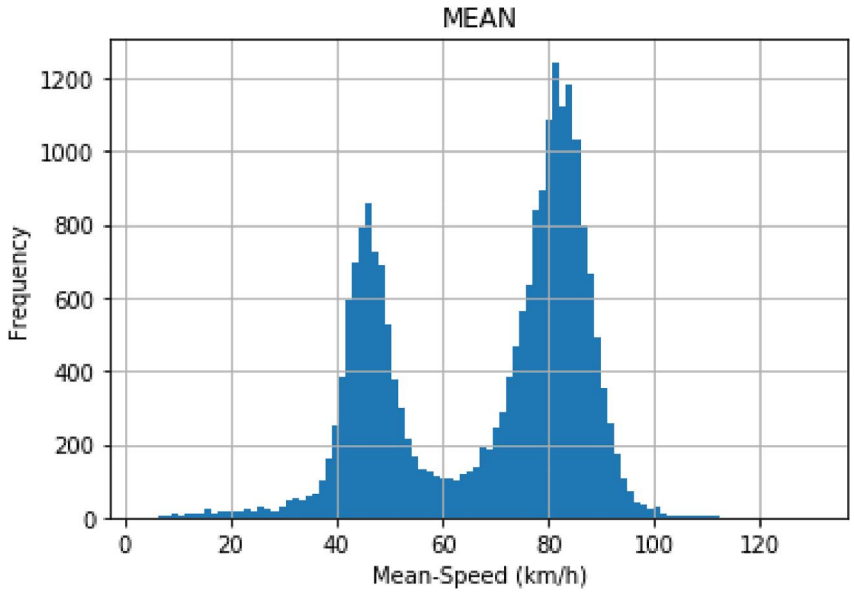

TABLE 2 Summary statistics of time series data

Count	Mean	Std. dev	Minimum	25%	50%	75%	Maximum
21,888	68.2	18.4	3.2	48.6	76.2	83.2	130.2

$$y_t^{(\lambda)} = l_{t-1} + \varnothing b_{t-1} + \sum_{i=1}^{T} s_{t-m_i}^{(i)} + d_t \tag{8}$$

$$l_t = l_{t-1} + \varnothing b_{t-1} + \alpha d_t \tag{9}$$

$$b_t = \varnothing b_{t-1} + \beta d_t \tag{10}$$

$$d_t = \sum_{i=1}^{p} \varphi_i d_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t \tag{11}$$

where: $y_t^{(\lambda)}$—time series at moment t (Box-cox transformed), $s_t^{(i)}$—ith seasonal component, l_t—local level, b_t—ARMA (p, q) process for residuals, d_t is a series of ARMA models with orders (p, q), e_t—Gaussian white noise, T—the number of seasonalities, $\varnothing$ is the damping parameter for trend and α, β are the smoothing parameters. Also, l_{t-1}, b_{t-1} are one-step past values and d_{t-i}, e_{t-i} are ith step past values of respective terms.

3.2.2 | Multi-layer perceptron

An MLP offers a flexible architecture to provide a functional approximation of a dataset. The building block of MLP is a single neuron that takes multiple weighted inputs along with a bias to activate a non-linear processing block to provide an output. The overall network has an input layer with multiple inputs, one or more hidden layers with the number of neurons, and an output layer with one or more outputs. A subset of labelled data trains the network. Input weights of every perceptron are initially randomised, the output is calculated. If there is an unacceptable error between calculated output and desired output, the network weights are updated using the back-propagation algorithm. The architecture mathematically is expressed as in Equation (12).

$$F(\mathbf{x},\ \boldsymbol{w}) = \varphi\left(\sum_k \omega_{ok}\varphi\left(\sum_j \omega_{kj}\varphi\left(\ldots.\varphi\left(\sum_i \omega_{li}x_i\right)\right)\right)\right) \tag{12}$$

The expression shows the nested application of sigmoid function, φ, with the associated input weights, ω. This approximation's granularity depends on the number of hidden layers and the number of neurons in each layer. According to the universal approximation theorem [44], only one layer is needed to approximate any continuous function. However, if computational complexity and learning rate are taken into account, it is argued that a second hidden layer may be justified. Another issue of prime importance is the number of neurons in a hidden layer. This is believed that it cannot be determined mathematically rather, it is based on experimentation, keeping in view the training data size. Over-training a network may cause overfitting compromising the generalisation ability of the network.

3.2.3 | LSTM and GRU

Long Short-Term Memory network is a modified form of RNN. In RNN, as shown in Figure 5, neurons are augmented with feedback connections developing contextual information of the previously seen values and producing the capability to memorise past information.

However, RNNs are maligned with vanishing/exploding gradient problems [45] and cannot memorise long dependence in time series. In [46], the author has suggested mitigating this

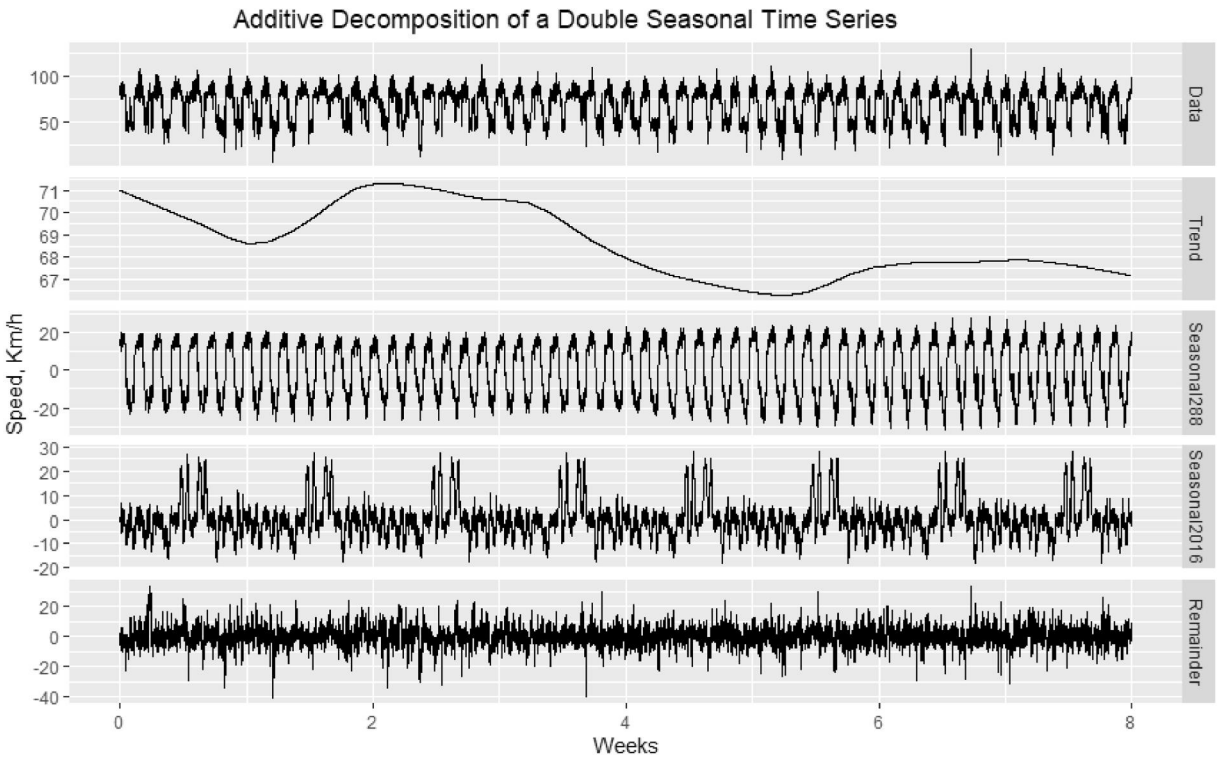

FIGURE 3 Additively decomposed data for 76 days revealing double seasonality (*y*-axis is the mean speed in Km/h)

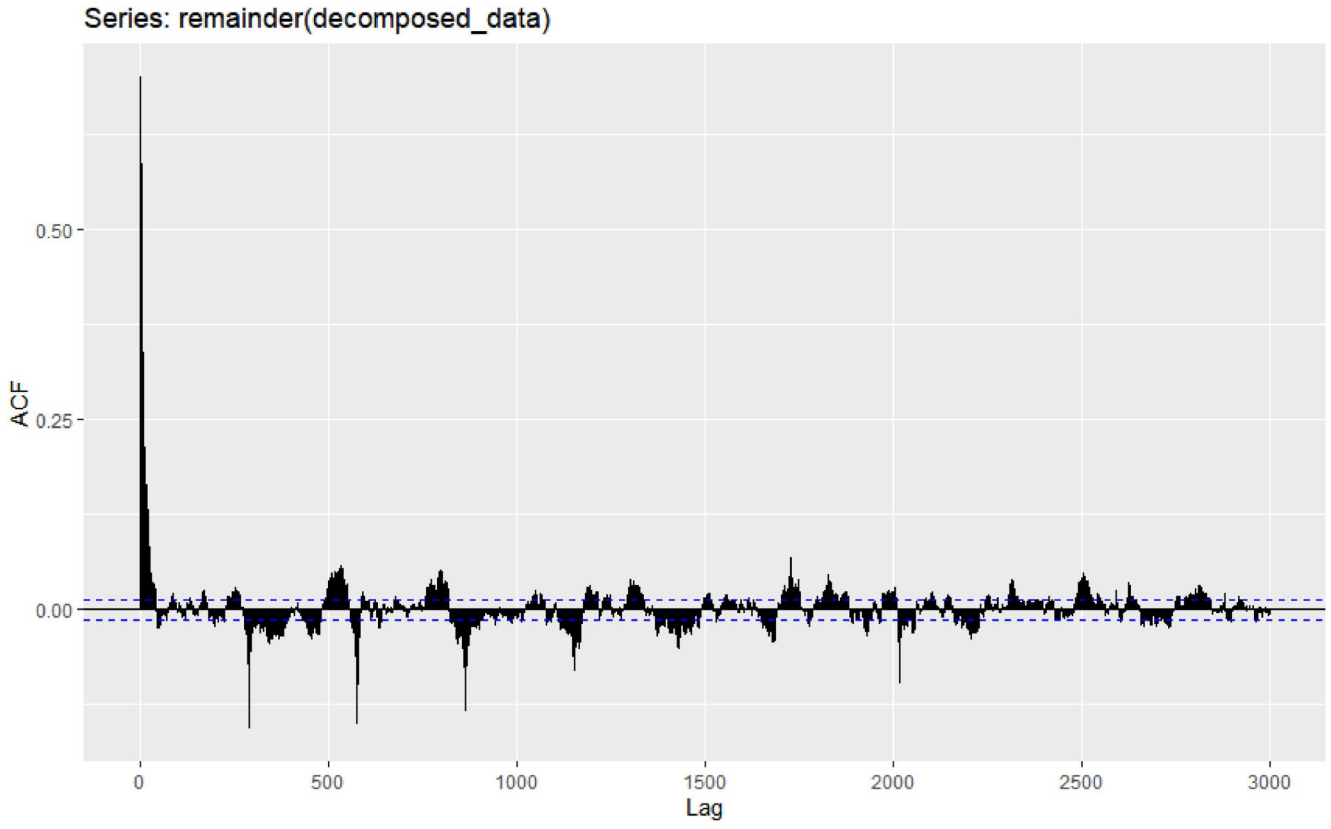

FIGURE 4 Autocorrelation of the remainder of decomposed data

problem by including an adaptive critic in a feedback loop that learns from the environment and controls the network activities, as shown in Figure 5.

The use of adaptive critic will induce two properties in the network:

1. It will learn the long-term dependence in a time sequence used for training.
2. The network will adapt to short-term statistical variations in the environment (training sequence) without affecting the tuned network.

An LSTM network is a manifestation of this idea of maintaining and controlling contextual information with the help of three control gates:

- An input gate (i_t) controls when and how much information from the input is allowed to be remembered by the contextual memory.
- A forget gate (f_t) controls the case when the context has become insignificant and should be erased.
- An output gate (o_t) determines whether or not the contextual information should be passed to the next cell.

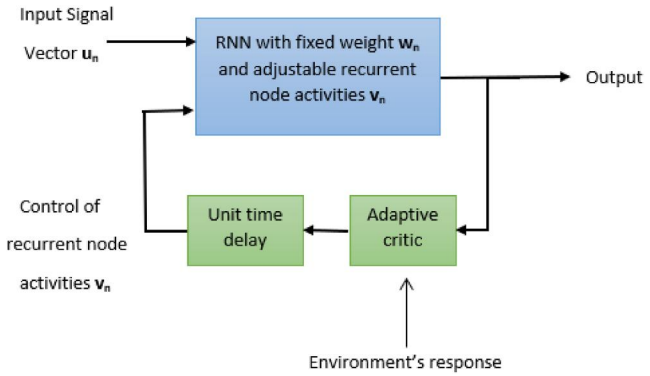

FIGURE 5 Conceptual architecture of recurrent neural network (RNN) with feedback; the basis for LSTM

A typical LSTM cell is shown in Figure 6.

The set of equations from Equations (13)–(18) describe the functionality of an LSTM cell.

$$f_t = \sigma\left(U_f x_t + W_f h_{t-1}\right) \tag{13}$$

$$\tilde{c}_t = tanh(U_c x_t + W_c h_{t-1}) \tag{14}$$

$$i_t = \sigma(U_i x_t + W_i h_{t-1}) \tag{15}$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1}) \tag{16}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{17}$$

$$h_t = o_t * tanh(c_t) \tag{18}$$

where x_t is the input vector, h_{t-1} is the previous cell output, c_{t-1} is the previous cell memory, h_t is the cell output, c_t is the current cell memory, U_f, U_i, and U_o are the U weights for forget gate, input gate and output gate. Similarly, W_f, W_c, W_i, W_o are the W weights for the forget gate, cell state, input, and output gates, respectively. Asterisk (*) denotes a point-wise multiplication.

Gated Recurrent Unit is similar to the LSTM, however, it does not have a separate memory cell. In GRU, the forget gate and the input gate are combined to form an update gate making it simpler and computationally efficient. Thus, there are only two gates present that is, reset gate and update gate. The entire functionality is shown in Figure 7 and given by Equations (19)–(22).

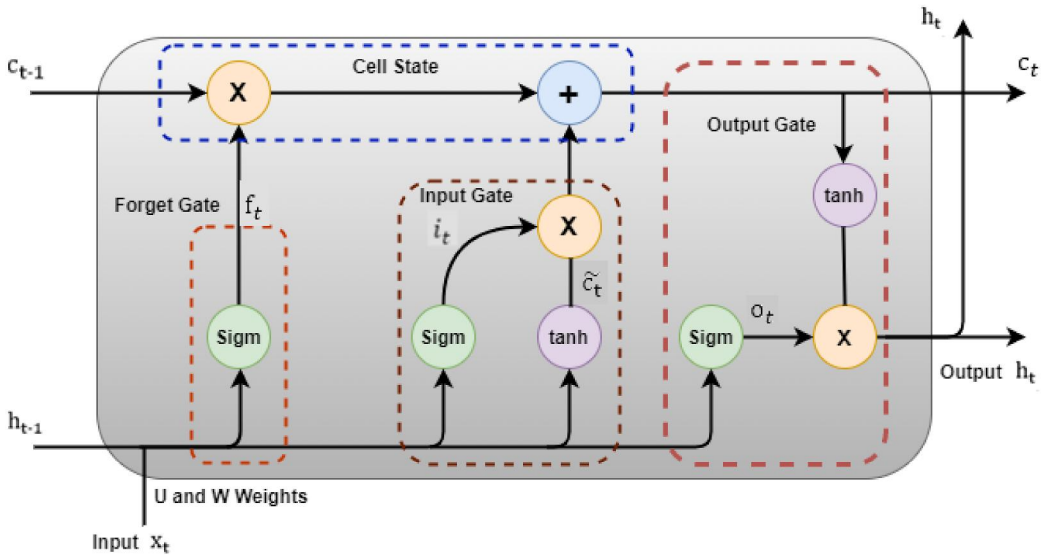

FIGURE 6 A Long Short-Term Memory (LSTM) cell

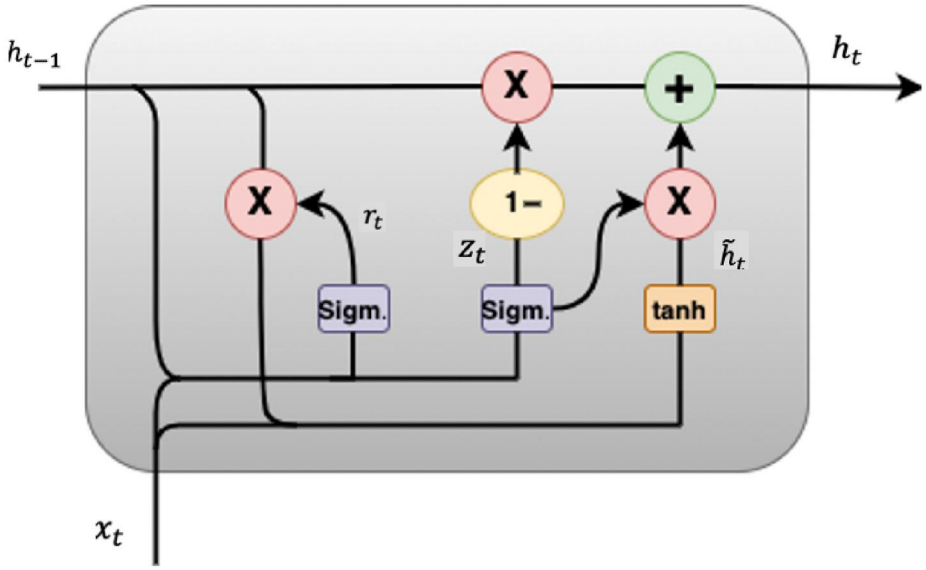

FIGURE 7 Gated Recurrent Unit

$$r_t = \sigma\left(U_r x_t + W_r h_{t-1}\right) \tag{19}$$

$$\tilde{h}_t = tanh\left(U x_t + W(r_t * h_{t-1}\right) \tag{20}$$

$$z_t = \sigma\left(U_z x_t + W_z h_{t-1}\right) \tag{21}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t\Big) \tag{22}$$

In Figure 7, r_t denotes reset gate having the similar function as in LSTM as given in Equation (19). The reset gate enables the system to forget the previous state. The candidate activation $\tilde{h}_t$, is given by Equation (20), whereas h_t is the activation of GRU which takes account of previous activation that is, h_{t-1} and the candidate activation $\tilde{h}_t$. The update gate is shown by z_t in Equation (21). It decides on how much the previous information will be passed onto the future.

3.2.4 | CNN

Convolutional-Neural Networks, mainly popular for image classification, are also adopted for time series prediction. CNNs are composed of a sequence of convolution and pooling layers. The convolution layer implements a filtering operation using a suitable kernel to extract features of input data, and the pooling layer performs sampling to reduce the size of the feature matrix. This procedure may be repeated many times depending on the depth and nature of sorting features hierarchically. The output layer is fully connected to the output node(s) for classification or prediction. The basic structure is shown in Figure 8.

3.2.5 | CNN-LSTM

CNN-LSTM combination, in a cascaded configuration, is claimed to render benefit of both models. The CNN part extracts the features of a time series, and LSTM captures interdependency between the data segments. Here, a CNN is used as a front-end followed by the LSTM. The features extracted by CNN are further processed by LSTM, which extracts the temporal variations. A typical configuration is shown in Figure 9, where the flattened output of CNN is given as the input to the LSTM for the final prediction.

3.3 | Regression analysis

Regression analysis involves finding a relation between an independent variable (also called a feature, predictor or regressor) and the dependent variable (outcome, response) that is to be predicted. It is shown in Equation (23) where x_i is a predictor, y is predicted variable and β_0, β_1 are parameters that need to be estimated for better approximation. The resulting approximation errors, ε_i, are termed as residual. The parameters are estimated usually using least-square method.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{23}$$

The residual error may be reduced if one can find more independent features that have correlation with the outcome. The resulting set of equations are given in compact and detailed form in Equation (24) and Equation (25), respectively.

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \epsilon \tag{24}$$

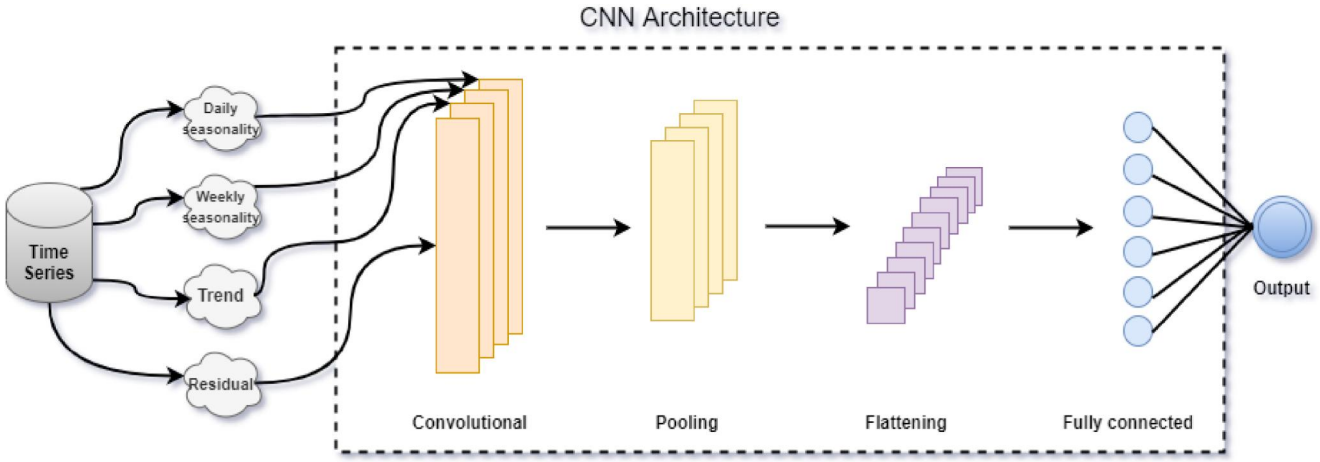

FIGURE 8 Multivariate Convolutional-Neural Network (CNN) architecture

FIGURE 9 CNN-LSTM

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1q} \\ 1 & x_{11} & x_{11} & \dots & x_{2q} \\ 1 & x_{11} & x_{11} & \dots & x_{3q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{11} & x_{11} & \dots & x_{nq} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_q \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{25}$$

In order to reduce predicting error, the independent variables are required to be independent (uncorrelated) with each other and expected to have some correlation with the dependent variable. A correlation matrix completely depicts the relation between independent and dependent variables as shown in Equation (26)

$$R = \begin{bmatrix} r_{x_1x_1} & r_{x_1x_2} & r_{x_1x_3} & r_{x_1x_4} & r_{x_1y} \\ r_{x_2x_1} & r_{x_2x_2} & r_{x_2x_3} & r_{x_2x_4} & r_{x_2y} \\ r_{x_3x_1} & r_{x_3x_2} & r_{x_3x_3} & r_{x_3x_4} & r_{x_3y} \\ r_{x_4x_1} & r_{x_4x_2} & r_{x_4x_3} & r_{x_4x_4} & r_{x_4y} \\ r_{yx_1} & r_{yx_2} & r_{yx_3} & r_{yx_4} & r_{yy} \end{bmatrix} \tag{26}$$

where

$$\text{Correlation coefficient} = r_{x_ix_j} = \frac{\sum_{j=1}^{5}\left(x_{ij}-\overline{x_i}\right)\left(x_{kj}-\overline{x_k}\right)}{\sqrt{\sum_{j=1}^{5}\left(x_{ij}-\overline{x_i}\right)^2}\sqrt{\sum_{j=1}^{5}\left(x_{kj}-\overline{x_k}\right)^2}} \quad -1 \le x_{ij} \le 1 \tag{27}$$

As can be seen that the matrix is symmetric about diagonal ($x_{ij} = x_{ji}$) and the diagonal itself gives the normalised variance (equal to 1) of all variables in context. The last column (and row) gives correlation between independent variables x_i and the response variable y.

3.4 | Performance measures

Three performance measures were applied to evaluate the predictions from the models, using test data in this work. Root means square, as given in Equation (28), is one of the popular measures due to its mathematical tractability

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(\text{Predicted}_i - \text{Actual}_i\right)^2}{N}} \tag{28}$$

Mean absolute percentage error, as given in Equation (29), is a unit-free measure of relative error.

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{\left|\text{Predicted}_i - \text{Actual}_i\right|}{\text{Actual}_i} \tag{29}$$

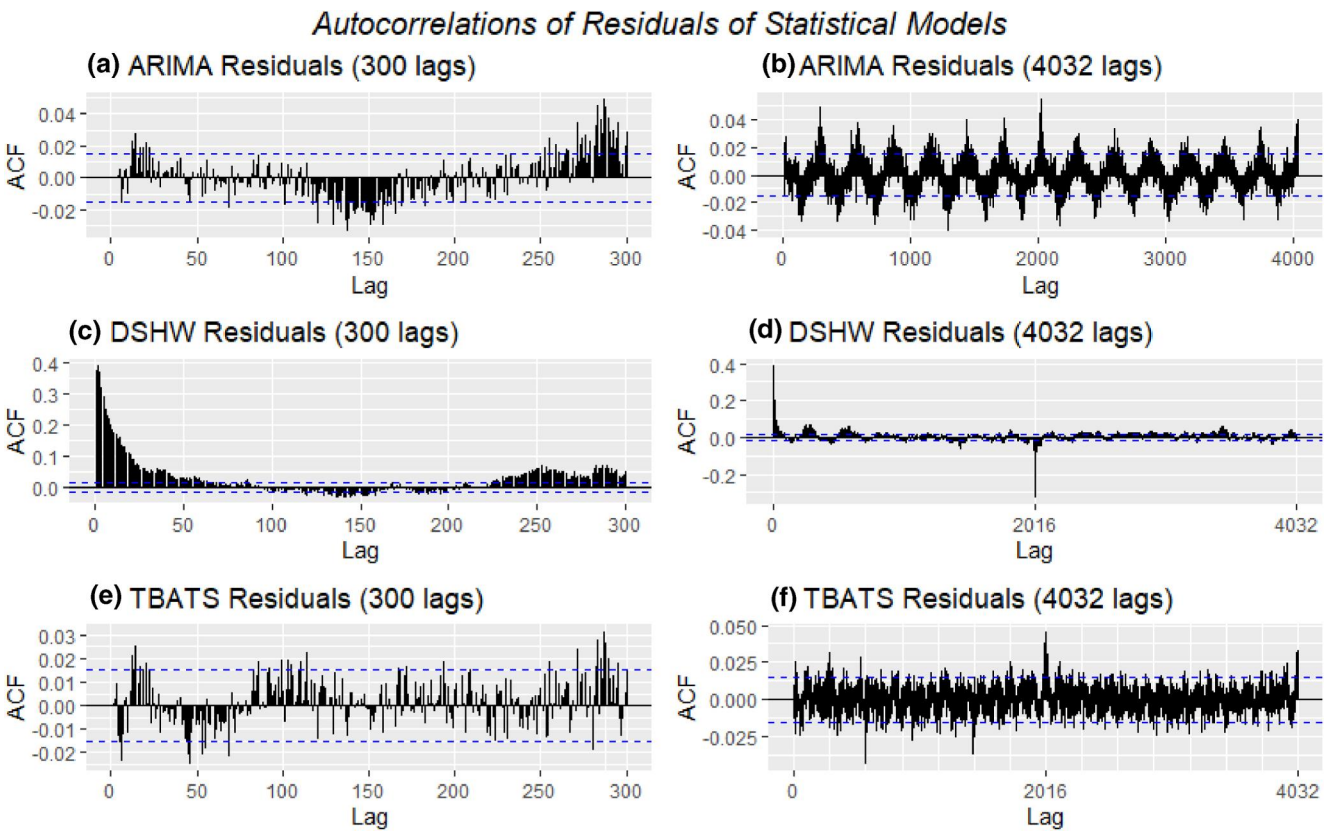

FIGURE 10 Auto correlation of model's residuals for lag 300 and 4032

However, it suffers the drawback of resulting in large values for errors close to zero. Secondly, it is biased towards positive errors and penalises greatly to negative errors. These shortcomings are reasonably addressed by symmetric-mean absolute percentage error (s-MAPE) as defined in Equation (30).

$$\text{sMAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|\text{Predicted}_i - \text{Actual}_i|}{\left(\frac{|\text{Predicted}_i|+|\text{Actual}_i|}{2}\right)} \quad (30)$$

3.5 | Implementation

Before exposing data to modelling, we filled the missing values by backward and forward filling methods and checked for outliers. Further, the data was split into training and test parts with 16,128 samples (56 days) and 5760 samples (20 days), respectively. For the classical ARIMA model, used as a benchmark, we first tested data for stationarity using 'adfuller' test and then searched for model order using 'auto.arima' function. The obtained model was then fitted on training data using SARIMA. It is to be noted that SARIMA takes care of single seasonality only. Double Seasonal Holt-Winter and TBATS, on the other hand, are designed for double seasonalities. For DSHW and TBATS, we used R libraries. The time series was formatted as an extended time-series (XTS) object and then converted to a multiple seasonality time-series (MSTS) object. Exponential triple smoothing was implemented using the DSHW model with daily (288 samples) and weekly ($288 \times 7 = 2016$ samples) seasonality arguments. The same data was used for TBATS.

Figure 10 shows the relative performance of three statistical models using two scales to highlight daily and weekly seasonality patterns with 300 and 4032 lags, respectively. Figure 10c,d show that the residual autocorrelation plot of the DSHW model captured daily seasonality (no significant spike at lag = 288). In contrast, it has missed some proportion of weekly seasonality as evident at lag = 2016. TBATS model, as shown in Figure 10e,f, has modelled both seasonality components better than the other two counterparts. Figure 10a,b show that the ARIMA model has poorly modelled both daily and weekly seasonality components. This behaviour is also reflected in Table 3 where TBATS enjoys the least error amongst three statistical models.

Machine learning models were implemented in univariate and multivariate modes. Motivation for multivariate implementation was derived from the fact that additively decomposed components (trend, seasonality, and residuals) are independent to each other because each convey a different characteristic of the data, as also revealed by correlation matrix shown in Figure 11. It may also be noted that each component shows a significant correlation with the predicted variable that is 'Mean_Speed'.

For each implementation, data was first formatted in supervised learning mode for the ML models with 11 columns for a time-lag of 11 steps and one column as a label. A train-

test split was performed in a 70/30 ratio. A grid search was performed to identify suitable configurations in each case. In order to avoid over/under-fitting, train-validation-loss curves for univariate and multivariate models were obtained as shown in Figure 12 and Figure 13 respectively. The number of epochs for each model was selected based on these curves. Each model was implemented using one hidden layer only to achieve a fair comparison. In order to generalise the results, all ML models were implemented to three different road segments that carry similar type of traffic. The performance of the ML models, for highway segment 1 (Figure 1), is given in Table 4, and the summary of model parameters is shown in Table 5. The generalisation results are shown in Table 6 in Section 4.

TABLE 3 Performance of implemented statistical techniques

Measure/Model	ARIMA	DSHW	TBATS
RMSE	5.52	6.8	5.44
MAPE	7.01	8.87	6.96
sMAPE	6.70	7.94	6.66

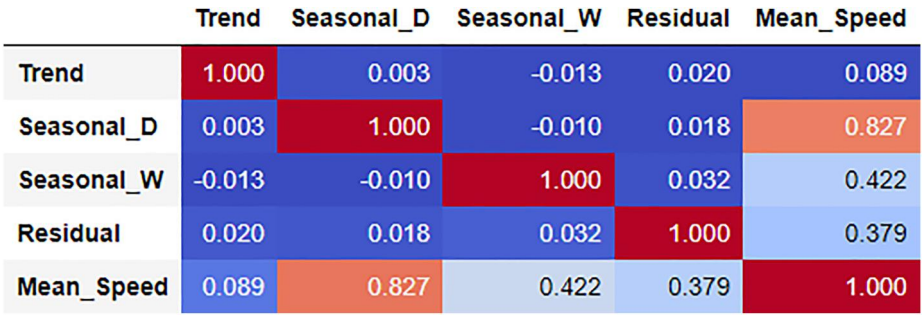

FIGURE 11 Correlation matrix

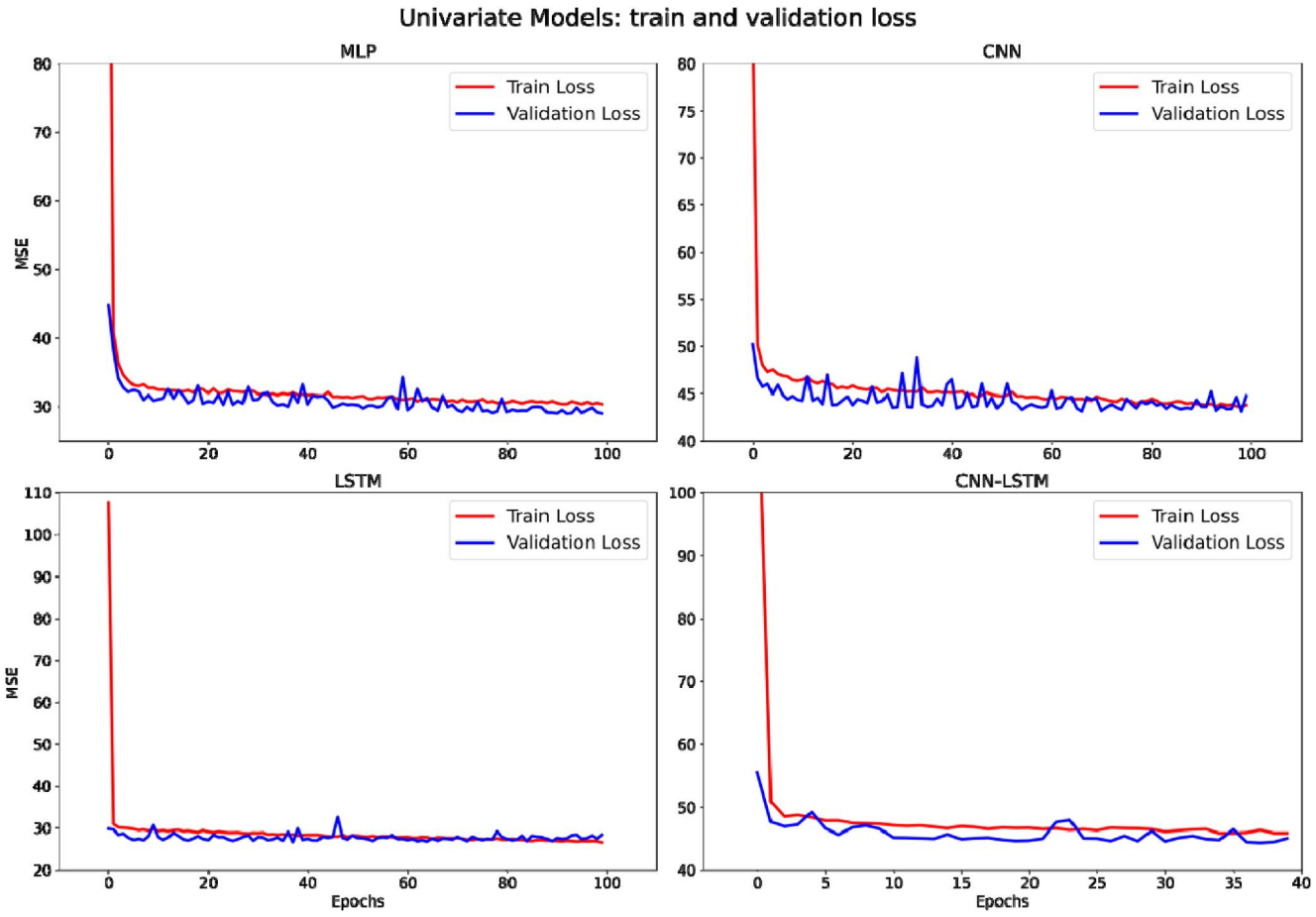

FIGURE 12 Train versus validation loss curves for the implemented univariate machine learning (ML) models

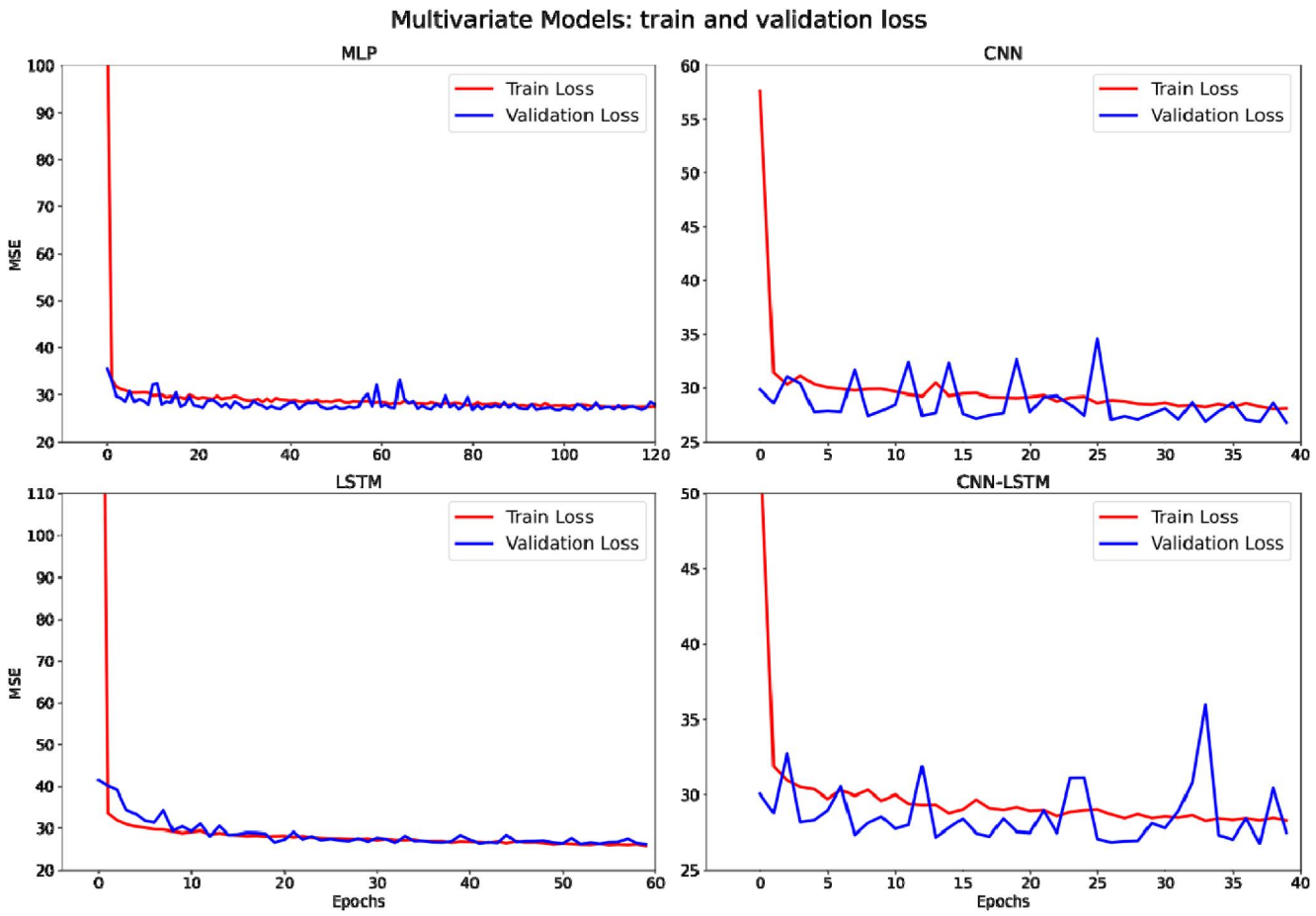

FIGURE 13 Train versus validation loss curves for the implemented multivariate machine learning (ML) models

TABLE 4 Performance of implemented algorithms for highway segment 1 (Figure 1)

	Measure/Model	MLP	CNN	LSTM	GRU	CNN-LSTM
Univariate	RMSE	5.41	5.53	5.25	5.23	5.47
	MAPE	6.74	6.87	6.42	6.72	7.06
	sMAPE	6.55	6.72	6.42	6.41	6.71
Multivariate	RMSE	5.15	5.23	5.22	5.07	5.16
	MAPE	6.49	6.73	6.86	6.42	6.48
	sMAPE	6.35	6.49	6.48	6.19	6.35
	RMSE improvement (%)	4.63	5.33	0.16	2.86	5.60

4 | RESULTS AND DISCUSSION

This research attempted to model a high frequency and relatively long traffic speed data with dual seasonality. A pictorial view of the complete implementation is shown in Figure 14. After exploration, three statistical (ARIMA, DSHW, and TBATS) and five ML techniques (MLP, CNN, LSTM, GRU, and CNN-LSTM) were investigated for the available dataset.

The results for statistical and ML techniques are presented in Table 3 and Table 4, respectively. Amongst the statistical models, TBATS, designed to handle multiple seasonality, has performed better than SARIMA and DSHW resulting in a minimum RMSE error. This superiority is also reflected in autocorrelation plots of residuals shown in Figure 10, which indicates that TBATS has captured double seasonality better than two other models. As may be expected, however, the autocorrelation is not completely white and some traces of seasonal components at samples 288 and 2016 are shown. This behaviour may be attributed to the heteroskedasticity of data which is also obvious in the time-plot of residual in Figure 3.

Performance of univariate and multivariate ML models is given in Table 4. The prediction plots for two consecutive days are shown in Figure 15 and Figure 16 along with corresponding zoomed parts. It may be noted that there is a difference in day

TABLE 5 The summary of model parameters

	Univariate			Multivariate		
Model	**Inputs**	**Hidden layer units**	**Epochs**	**Inputs**	**Hidden layer units**	**Epochs**
MLP	11 + 1	100	100	11 × 4 + 4	100	120
CNN	11 + 1	50 filter = 64, kernel-size = 3	100	11 × 4 + 4	50 filter = 64, kernel-size = 3	40
LSTM/GRU	11 + 1	100	100	11 × 4 + 4	50	60
CNN-LSTM	11 + 1	Filter = 64, kernel-size = 3, LSTM-cells = 50	40	11 × 4 + 4	Filter = 64, kernel-size = 3, LSTM-cells = 50	40

Note: 1. Univariate inputs represent 11 lagged inputs and one label for supervise learning. 2. Multivariate inputs represent 11 lagged values for four features and four labels.

TABLE 6 Generalisation performance for three selected highway segments

		MLP	CNN	LSTM	GRU	CNN-LSTM
Highway segment 1	Univariate	5.41	5.53	5.25	5.23	5.47
	Multivariate	5.15	5.23	5.22	5.07	5.16
	%Improvement	4.63	5.33	0.16	2.86	5.60
Highway segment 2	Univariate	5.93	5.97	5.74	5.70	6.03
	Multivariate	5.71	5.68	5.61	5.57	5.61
	%Improvement	3.54	4.74	2.24	2.28	6.98
Highway segment 3	Univariate	7.26	7.28	7.23	7.24	7.26
	Multivariate	7.11	7.08	7.18	7.04	7.14
	%Improvement	1.97	2.78	0.67	2.71	1.56

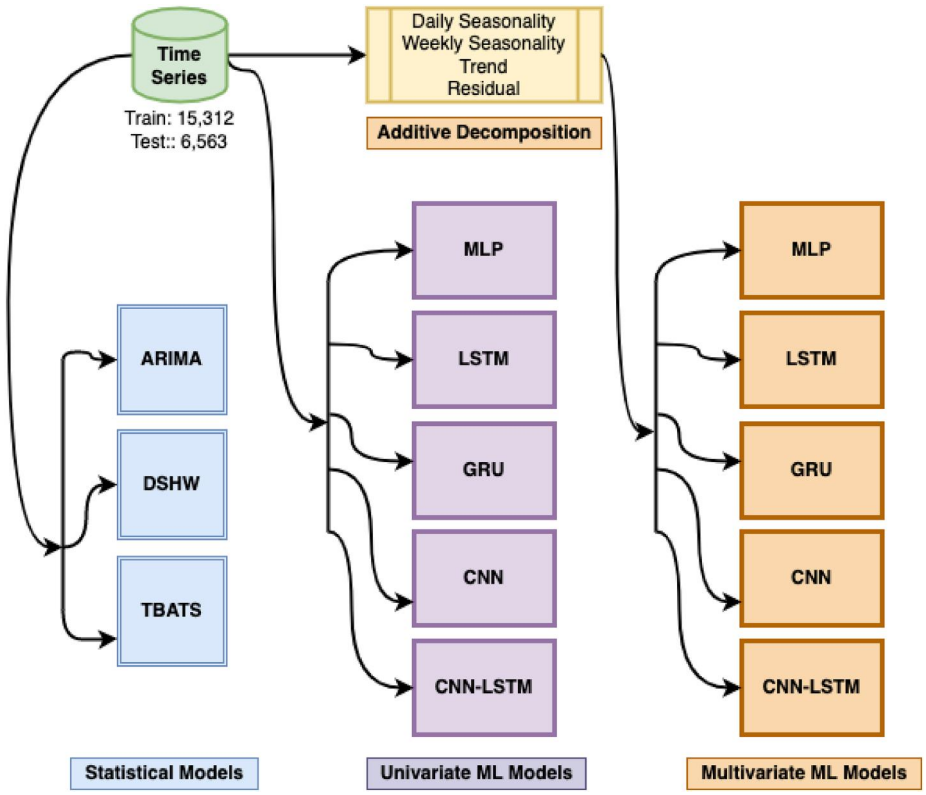

FIGURE 14 Complete implementation

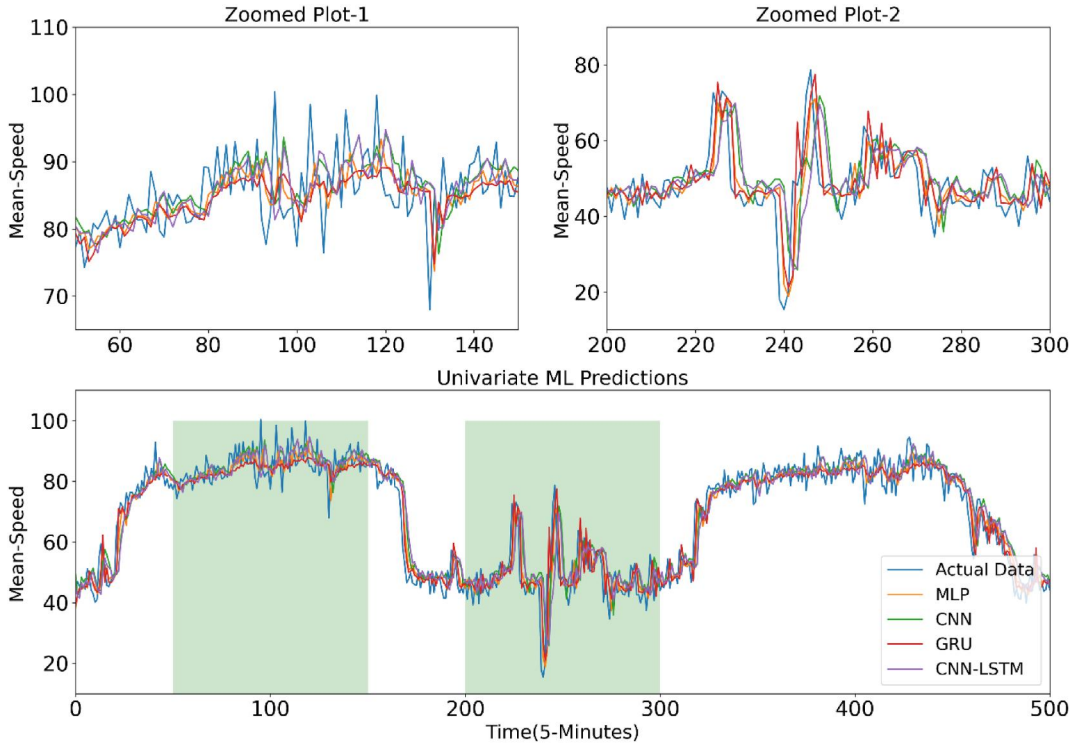

FIGURE 15 Prediction of univariate models taken over the time of 2 days (samples taken with an increment of 5 min)

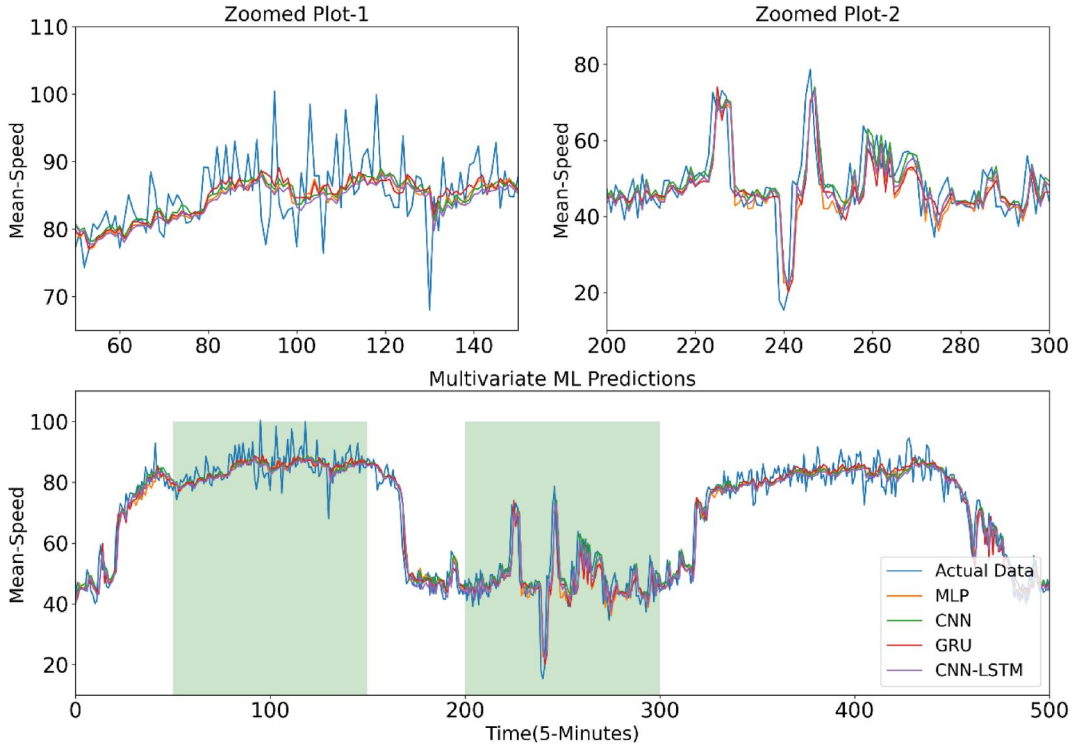

FIGURE 16 Prediction of multivariate models taken over 2 days (samples taken with an increment of 5 min)

and night behaviour of traffic. In daytime, traffic is slower and have gradual variations whereas night traffic is faster and rapid. Random variations are present in both the cases. As expected, the models have responded better to gradual variations whereas random variations are aggregated trend. Amongst all ML models, GRU has performed the best owing to its capability of

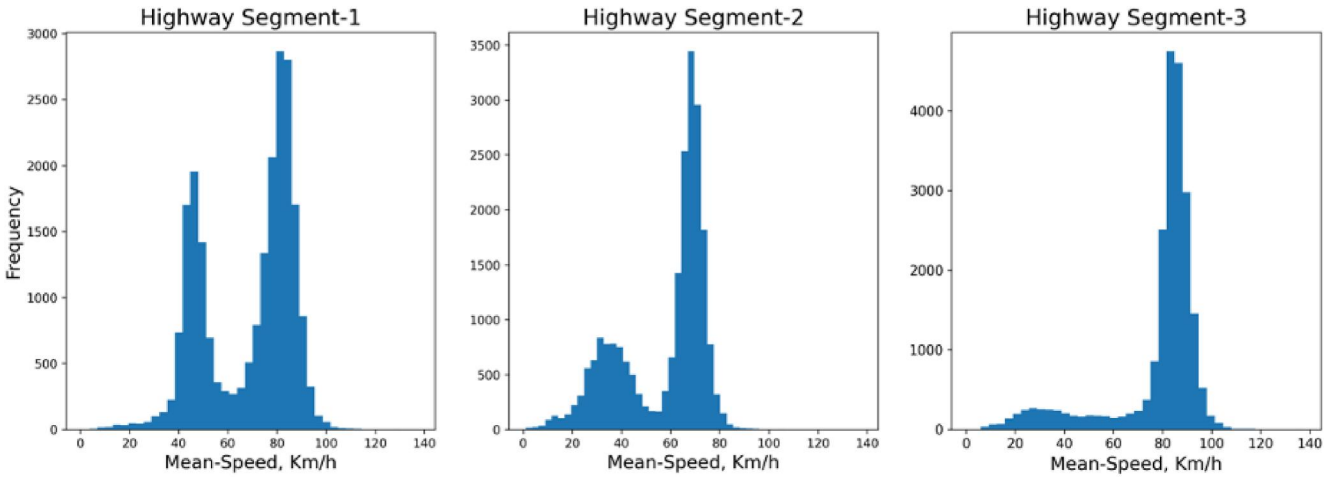

FIGURE 17 Histogram for three highway segments

capturing long term temporal dependence and better stability than LSTM by virtue of reset gate that enables the system to control the information passed further.

The traffic-speed time series used in this exploration has multiple dynamics. Firstly, variations in speed are spread over a range of 3–130 Km/h. Secondly, there are day-night cyclic variations as discussed above. Moreover, there is a pronounced difference between working days traffic and week-end traffic as revealed in Figure 3. The suggested approach breaks these multiple dynamics into various components by additive decomposition. As shown in correlation matrix in Figure 11, these decomposed components are independent and uncorrelated. At the same time, they enjoy good correlation with the predicted variable (Mean-Speed) because these components have been extracted from the same. So, the decomposed components become a natural choice to be used as independent variables in multivariate ML models. When used in multivariate ML configuration, the decomposed components as independent variables facilitate improved optimisation reducing the mean-square error in each multivariate case as compared to the univariate counterpart. Nonetheless, the results show an asymptotic behaviour around a mean RMSE error of 5.16 and standard deviation of 0.11, emphasising the limitation of models to respond to random and heteroskedastic parts of data.

Finally, in order to generalise the results, the work has been extended to additional two highway segments having similar nature of traffic as shown in Figure 17. The generalisation results are shown in Table 6. It may be noted that all models have generalised well with a consistent improvement in performance for multivariate case.

5 | CONCLUSION

This research has evaluated the performance of selected statistical and ML models for dynamic and high frequency data in a novel way. First, three statistical and five ML models are implemented in univariate configuration. Amongst the statistical models suitable for double seasonality data, TBATS has shown better results showing its superiority over DSHW and ARIMA models in capturing seasonal effects. Amongst univariate ML models, GRU has outperformed others. The main contribution of this work is the implementation of ML models in multivariate configuration using additively decomposed components as features (independent variables). The performance results show a consistent improvement in all the implementations demonstrating benefits of this approach. The improvement has resulted due to the independent nature of decomposed components that are inherently correlated with the predicted variable that is, traffic speed. This contribution becomes even more significant when no other features having greater correlation are available for multivariate models. The results, however, have a performance bottleneck due to heteroskedastic nature of data that may have not been captured by the implemented models. Consequently, we propose to investigate a mixture density network that claims to handle multimodal and heteroskedastic data [47, 48] as part of future work. Alternatively, an autoregressive conditional heteroskedasticity model may also be considered to model similar kind of data [49].

ACKNOWLEDGEMENT

The authors would like to thank HERE Technologies for providing the data for this research.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from HERE Technologies. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from Muhammad Ali and Benjamin Wilson with the permission of HERE Technologies.

ORCID

Muhammad Ali https://orcid.org/0000-0002-4230-9917

REFERENCES

1. Devi, Y.U., Rukmini, M.S.S.: Challenges and issues—a review. In: International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 1864–1867 (2016)
2. Yatskiv, I., Grakovski, A., Yurshevich, E.: An overview of different methods available to observe traffic flows using new technologies. Res. Dev. (May 2015) (2010)
3. Gardner, E.S.: Exponential smoothing part 1. J. Forecast. 4, 1–28 (1985)
4. Hannan, E.J., Quinn, B.G.: The determination of the order of an autoregression. J. R. Stat. Soc. Ser. B. 41(2), 190–195 (1979). https://doi.org/10.1111/j.2517-6161.1979.tb01072.x
5. Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Manag. Sci. 6(3), 324–342 (1960). https://doi.org/10.1287/mnsc.6.3.324
6. Yoon, S., Kum, D.: The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles. IEEE Intell. Veh. Symp. Proc. 2016-Augus(Iv) 1307–1312 (2016). https://doi.org/10.1109/IVS.2016.7535559
7. Hochreiter, S., Schmidhuber, J.: LSTM can solve hard long time lag problems. Adv. Neural Inf. Process. Syst. 473–479 (1997)
8. Zhao, S., Lin, S., Xu, J.: Time series traffic prediction via hybrid neural networks. In: 2019 IEEE International Conference on Intelligent Transportation Systems (ITSC 2019), pp. 1671–1676 (2019). https://doi.org/10.1109/ITSC.2019.8917383
9. Samsudin, R., Shabri, A., Saad, P.: A comparison of time series forecasting using support vector machine and artificial neural network model. J. Appl. Sci. 10(11), 950–958 (2010). https://doi.org/10.3923/jas.2010.950.958
10. Yolcu, U., Egrioglu, E., Aladag, C.H.: A new linear & nonlinear artificial neural network model for time series forecasting. Decis. Support Syst. 54(3), 1340–1347 (2013). https://doi.org/10.1016/j.dss.2012.12.006
11. Hamzaçebi, C.: Improving artificial neural networks' performance in seasonal time series forecasting. Inf. Sci. (Ny). 178(23), 4550–4559 (2008). https://doi.org/10.1016/j.ins.2008.07.024
12. Kim, T., King, B.R.: Time series prediction using deep echo state networks. Neural Comput. Appl. 32(23), 17769–17787 (2020). https://doi.org/10.1007/s00521-020-04948-x
13. Zhao, Y., et al.: Short-term passenger flow prediction with decomposition in urban railway systems. IEEE Access. 8, 107876–107886 (2020). https://doi.org/10.1109/ACCESS.2020.3000242
14. Chen, D., Zhang, J., Jiang, S.: Forecasting the short-term metro ridership with seasonal and trend decomposition using loess and LSTM neural networks. IEEE Access. 8, 91181–91187 (2020). https://doi.org/10.1109/ACCESS.2020.2995044
15. Tian, Y., et al.: LSTM-based traffic flow prediction with missing data. Neurocomputing. 318, 297–305 (2018). https://doi.org/10.1016/j.neucom.2018.08.067
16. Danqing, K., Lv, Y., Chen, Y.: Short-term flow prediction with LSTM recurrent neural network. In: IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), vol. 1(1), pp. 1–6 (2017)
17. Khodabandelou, G., et al.: Attention-based gated recurrent unit for links traffic speed forecasting. In: 2019 IEEE International Conference on Intelligent Transportation Systems (ITSC 2019), pp. 2577–2583 (2019). https://doi.org/10.1109/ITSC.2019.8917027
18. Li, X., et al.: Application of gated recurrent unit (GRU) neural network for smart batch production prediction. Energies. 13(22), 6121 (2020). https://doi.org/10.3390/en13226121
19. Wenqi, L., Dongyu, L., Menghua, Y.: A model of traffic accident prediction based on convolutional neural network. In: 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE 2017), pp. 198–202 (2017). https://doi.org/10.1109/ICITE.2017.8056908
20. Ma, X., et al.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors (Switzerland). 17(4), 818 (2017). https://doi.org/10.3390/s17040818
21. Bogaerts, T., et al.: A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transport. Res. C Emerg. Technol. 112(December 2019), 62–77 (2020). https://doi.org/10.1016/j.trc.2020.01.010
22. Le, V.A., Le Nguyen, P., Ji, Y.: Deep convolutional LSTM network-based traffic matrix prediction with partial information. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2019), pp. 261–269 (2019)
23. Karami, Z., Kashef, R.: Smart transportation planning: data, models, and algorithms. Transport Eng. 2(June), 100013 (2020). https://doi.org/10.1016/j.treng.2020.100013
24. Yang, X., et al.: Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models. J. Adv. Transport. 2020, 1–16 (2020). https://doi.org/10.1155/2020/9628957
25. Yu, Y., Sun, L., Li, G.: Short-term traffic forecasting using the double seasonal Holt-Winters method. In: CICTP 2016—Green Multimodal Transport Logistics—Proceedings of the 16th COTA International Conference of Transportation Professionals, pp. 162–173 (2016). https://doi.org/10.1061/9780784479896.016
26. Song, Z., et al.: Short-term traffic speed prediction under different data collection time intervals using a SARIMA-SDGM hybrid prediction model. PLoS One. 14(6), 1–19 (2018). https://doi.org/10.1371/journal.pone.0218626
27. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and Machine Learning forecasting methods: concerns and ways forward. PLoS One. 13(3), e0194889 (2018). https://doi.org/10.1371/journal.pone.0194889
28. Smyl, S.: A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int. J. Forecast. 36(1), 75–85 (2020). https://doi.org/10.1016/j.ijforecast.2019.03.017
29. Karlaftis, M.G., Vlahogianni, E.I.: Statistical methods versus neural networks in transportation research: differences, similarities and some insights. Transport. Res. C Emerg. Technol. 19(3), 387–399 (2011). https://doi.org/10.1016/j.trc.2010.10.004
30. Kumar, S.V., Vanajakshi, L.: Short-term traffic flow prediction using seasonal ARIMA model with limited input data. Eur. Transp. Res. Rev. 7(3), 1–9 (2015). https://doi.org/10.1007/s12544-015-0170-8
31. Kourentzes, N., Crone, S.F.: Analysis of Modelling Challenges from Increasing Data Frequency, vol. January (2008)
32. Li, K., Zhai, C., Xu, J.: Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN. In: 2017 Chinese Automation Congress (CAC), pp. 2804–2807 (2017)
33. Jia, Y., Wu, J., Du, Y.: Traffic speed prediction using deep learning method. In: IEEE Conference on Intelligent Transportation Systems Proceedings, ITSC, pp. 1217–1222 (2016). https://doi.org/10.1109/ITSC.2016.7795712
34. Gu, Y., et al.: The short-term prediction of vehicle speed of urban freeway based on RBF neural network. DEStech Trans. Comput. Sci. Eng. aita, 32–37 (2017). https://doi.org/10.12783/dtcse/aita2017/15986
35. Kumar, K., Parida, M., Katiyar, V.K.: Short term traffic flow prediction for a non urban highway using artificial ScienceDirect short term traffic flow prediction for a non urban highway using artificial neural network. Procedia-Social and Behavioral Sciences. December, 755–764 (2013). https://doi.org/10.1016/j.sbspro.2013.11.170
36. Cao, M., Li, V.O.K., Chan, V.W.S.: A CNN-LSTM model for traffic speed prediction. IEEE Veh. Technol. Conf. 2020-May, 1–5 (2020). https://doi.org/10.1109/VTC2020-Spring48590.2020.9129440
37. Lee, M.C., Lin, J.C., Gran, E.G.: Distributed fine-grained traffic speed prediction for large-scale transportation networks based on automatic LSTM customization and sharing. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 12247 LNCS, 234–247 (2020). https://doi.org/10.1007/978-3-030-57675-2_15
38. Ma, X., et al.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transport. Res. C Emerg. Technol. 54, 187–197 (2015). https://doi.org/10.1016/j.trc.2015.03.014

39. Lv, Z., et al.: LC-RNN: A Deep Learning Model for Traffic Speed Prediction, pp. 3470–3476 (2018)
40. Song, C., et al.: Traffic speed prediction under weekday using convolutional neural networks concepts. IEEE Intell. Veh. Symp. Proc. (Iv), 1293–1298 (2017). https://doi.org/10.1109/IVS.2017.7995890
41. Wang, H., et al.: Empirical mode decomposition-autoregressive integrated moving average: hybrid short-term traffic speed prediction model. Transport. Res. Rec. 2460(1), 66–76 (2014). https://doi.org/10.3141/2460-08
42. Guide—location library—HERE developer. https://developer.here.com/documentation/location-library/dev_guide/docs/location-referencing-tmc.html. Accessed 20 February 2021
43. Taylor, J.W.: Short-term electricity demand forecasting using double seasonal exponential smoothing. J. Oper. Res. Soc. 54(8), 799–805 (2003). https://doi.org/10.1057/palgrave.jors.2601589
44. Csaji.: <Approximation with artificial neural networks (MSc).pdf>. p. 11 (2001)
45. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Read. Cognit. Sci. A Perspect. Psychol. Artif. Intell. (V), 399–421 (2013). https://doi.org/10.1016/B978-1-4832-1446-7.50035-2
46. Haykin, S., et al.: Neural Networks and Learning Machines Third Edition (2009)
47. Chen, R., et al.: Predicting future locations of moving objects by recurrent mixture density network. ISPRS Int. J. Geo-Inf. 9(2), 116 (2020). https://doi.org/10.3390/ijgi9020116
48. Holm, M.E., Kirkeby, L.V.: Utilizing Mixture Density Networks for Travel Time Probability Distribution Predictions (2020)
49. Kristjanpoller, W., Minutolo, M.C.: A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. Expert Syst. Appl. 109, 1–11 (2018). https://doi.org/10.1016/j.eswa.2018.05.011

How to cite this article: Ali, M., et al.: Traffic speed prediction of high-frequency time series using additively decomposed components as features. IET Smart Cities. 4(2), 92–109 (2022). https://doi.org/10.1049/smc2.12027