ISSN: 2182-2069 / E-ISSN: 2182-2077

Revolutionizing Traffic Flow Prediction Using a Hybrid Deep Learning Models with Kookaburra Optimization Algorithm

R.Y. Aburasain[1*]

[1*]Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia. raburasain@jazanu.edu.sa,
https://orcid.org/0009-0003-5786-8011

Received: December 25, 2024; Revised: January 30, 2025; Accepted: February 10, 2025; Published: February 28, 2025

Abstract

Traffic flow prediction is an essential element in intelligent transportation models, aiming to forecast vehicular movement patterns on road networks. This system aids in diminishing traffic issues, particularly traffic congestion, to enhance traffic quality. This system uses extensive data information and transmission technology to provide more efficient traffic control and real-time road infrastructure analysis. This system relies on traffic prediction as a crucial component. Traffic prediction main intention is to forecast future traffic conditions on transportation networks based on historical analysis. Using artificial intelligence technology, especially machine learning, to generate and create traffic flow prediction is ground-breaking. It provides conventional techniques for generating and developing prediction of traffic flow. Deep learning (DL) is a machine learning (ML) subclass strongly associated with each other in artificial intelligence. In this aspect, this manuscript devises an effective traffic flow prediction using the Kookaburra Optimization Algorithm with a hybrid DL (ETFP-KOAHDL) model. The aim of the ETFP-KOAHDL method is to provide a reliable and efficient solution for real-time traffic flow prediction using advanced DL models. Initially, the ETFP-KOAHDL technique follows a min-max scaler to pre-process the input data. Besides, the HDL algorithm implements the prediction process, which incorporates convolution long short-term memory with an autoencoder (CLSTM-AE) system. Finally, the ETFP-KOAHDL model designs the KOA for the optimal hyperparameter selection. An extensive set of experiments was conducted to examine the improvised prediction results of the ETFP-KOAHDL method. The simulation exploration of the ETFP-KOAHDL system exposed an excellent MAPE value of 04.82% over other approaches under diverse runs. Also, the process can predict traffic flow at a consecutive time, which helps in effective travel planning for autonomous vehicles.

Keywords: Traffic Flow Prediction, Min-Max Scaler, Intelligent Transportation Systems, Deep Learning, Kookaburra Optimization Algorithm.

1 Introduction

The prediction of traffic flow is a fundamental aspect of intelligent transportation models, enabling the effective management of urban traffic and reduction of congestion (Zhou et al., 2019). One of the essential parts of ITS is precise traffic flow prediction (TFP), which has gained significant attention (Manipriya et al., 2020). At the same time, traffic flow in the system can be expected by employing prior traffic flow information (Jia & Yan, 2020). Traffic flow prediction is considered difficult owing to the

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 1 (February), pp. 486-501.
DOI: *10.58346/JISIS.2025.I1.032*

*Corresponding author: Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia.

variables' unpredictable nature (Gu et al., 2019). Accurate TFP is significant to numerous real-world applications (Yağız et al., 2022). For instance, TFP aids the city in alleviating congestion and car-hailing requests, estimating rapid car-sharing businesses pre-allocating cars to high-demand areas (Nguyen et al., 2021). To discover this issue, accessible traffic-related datasets offer possible novel perspectives. So, traffic data displays robust dynamic correlation in spatial and time-based dimensions (Azgomi & Jamshidi, 2018). It is one of the most popular study areas for mining nonlinear and complex spatial-temporal designs that aid in making precise traffic forecasts (Kuru & Khan, 2020).

Preceding short-term traffic flow forecast techniques are categorized into three types: hybrid, parametric, and non-parametric (Seuwou et al., 2020). The parametric method contains Kalman filtering and time-series models (Kostić et al., 2021). The Autoregressive Integrated Moving Average (ARIMA) approach has several variants like subset ARIMA, Kohonen ARIMA (KARIMA) and seasonal ARIMA (SARIMA) (Chehri & Mouftah, 2019). However, owing to traffic flow's stochastic and nonlinear nature, these models reflect only sequential differences in traffic flow and offer unsatisfying prediction results (Chen et al., 2021). The non-parametric approach contains support vector regression (SVR), k-nearest neighbour (KNN), and artificial neural networks (ANNs) models (Ma et al., 2015). However, it is exposed that the KNN model for TFP will not overtake time-series techniques. Additionally, classical machine learning (ML) based techniques use handcrafted features to capture the traffic flow features, which need to be revised for accurate prediction performance (Zhao et al., 2019). Furthermore, early works based on neural networks employ shallow networks or single hidden layers (HL) incapable of seizure indeterminate and complex traffic flow nonlinearity (Li et al., 2017). Deep learning (DL) has gained popularity in numerous fields, namely speech recognition and computer vision (Li et al., 2017). DL methods employ a multiple-layer framework to mechanically remove characteristic features from a vast amount of raw data compared to ANN methods (Yu et al., 2017). Recently, DL has motivated a flow of interest in transportation studies. For TFP, many DL techniques have been projected (Malik, 2022). The surge of ITS has accentuated the requirement for precise TFP, exhibiting challenges due to the unpredictability of traffic variables. Whereas DL holds promise, its effectual application in transportation studies remains challenging (Contreras et al., 2011).

In (Wen et al., 2023), a new framework named RPConvformer was developed. A proposed system embedding layer composed of convolutional elements consists of basis and fundamental 1D convolutional (Duhaim et al., 2024). An origin vector is designed to learn relative location data automatically. The encoding and the decoding assume the multi-head attention mechanism. Furthermore, the critical mask method is employed after calculating the attention matrix. Chen et al., (2021) developed data-denoising systems called Empirical Mode Decomposition (EMD), Wavelet (WL), and Ensemble EMD (EEMD) to overwhelm the probable data outliers. Then, LSTM neural architecture is proposed to fulfil the task of forecasting traffic flow. In (Zhou et al., 2022), the Filter Attention-based Spatiotemporal NN (FASTNN) is developed. Primary, 3D-Convolutional Neural Network (CNN) is employed for extraction. Next, the filter spatial attention element is built to measure spatio-temporal aggregation. A matrix factorization-based resample segment is also created that learns intrinsic correlation spontaneously.

In (Li et al., 2021), a technique HDL based on WL decomposition, CNN-LSTM, named W-CNN-LSTM, is developed. The WL decomposition methodology is mainly utilized to decay unique traffic flow data. The decomposed orders are fed into the CNN-LSTM DL method. The arithmetical test is performed on five benchmarks depend upon England's traffic flow database. Xia et al., (2024) developed a hybrid DL using Residual Self-Attention and Bidirectional Gated Recurrent Units (GRUs) integrated with Convolution-GRU (RSAB-ConvGRU) system. The RSA-ConvGRU element contains Conv-GRU

and RSA parts. In addition, Conv-GRU uses convolutional and GRUs. Furthermore, the RSA device is employed to define the influence of traffic feature. Liu et al., (2022) designed a new hybrid technique, FGRU, relating to Fuzzy Inference System (FIS) and GRU neural network. The GRU technique is mainly exploited to grab temporal needs in traffic flow data. FIS is utilized to detect defects in DL by decreasing the effect of ambiguous information. A temporal feature development mechanism is also presented for calculating correct time intervals as classic inputs.

In (Hu et al., 2023), a multi-layer technique based on transformer and DL for traffic flow prediction (MTDLTFP) is designed. Initially, this method appeals to the transformer technique, which employs manifold encoders and decoders. In the prediction stage, MTDLTFP uses the DL method, which inputs unseen features into CNN and multi-layer Feedforward Neural Networks (MFNNs). Finally, a linear method is utilized to combine dual prediction scores. In (Zhang et al., 2022), a new DL framework called Adaptive Reinitialized Differential Evolution (ARDE) and neural basis Expansion Analysis for Time-Series (N-BEATS) prediction is presented. Foremost, the developed model-based DL framework is expressed to demonstrate traffic flow information. Next, an improved evolutionary technique called ARDE was proposed to optimize the N-BEATS structure and hyperparameter. Gao et al., (2024) present a fusion DL technique of three components. The spatio-temporal data (STD) fusion module is also utilized. In (Abdullah et al., 2023), a bidirectional recurrent neural network (BRNN) with GRU techniques. The model also employs real-time sensor information.

Bharathi et al., (2024) proposed a fusion DL model joining multivariate LSTM, variational mode decomposition, and regression methods. In (Zhang et al., 2017), an end-to-end ST-ResNet framework is developed, implementing residual neural networks. Shi et al., (2015) introduce the Convolutional LSTM (ConvLSTM) approach by extending the fully connected LSTM with convolutional models. Wang et al., (2020) proposed a technique for implementing DL to STD mining (STDM) models. Miao et al., (2022) present a multi-task Bayes-enhanced Adversarial Spatial Temporal Network (MBA-STNet) model integrating shared-private, Bayesian heterogeneous STNet, attentive temporal queue, and adversarial loss. Lai et al., (2023) developed the lightweight Correlated time series (LightCTS) model by employing plain stacking of time-based and spatial operators and features light operator modules (L-TCN and GL-Former) together with a last-shot compression system. The advantages of the implemented techniques encompass enhanced accuracy in TFP, while their disadvantages mainly involve greater computational complexity and difficulty in handling convolutional spatial-temporal patterns. However, a gap is present in effectually incorporating various models for addressing the stated threats collectively.

This manuscript devises an effective traffic flow prediction using the Kookaburra Optimization Algorithm with a hybrid DL (ETFP-KOAHDL) model. The ETFP-KOAHDL technique primarily follows a min-max scaler to pre-process the input dataset. Besides, the prediction process can be performed using the HDL model, which incorporates convolutional LSTM with an autoencoder (CLSTM-AE) model. Finally, the ETFP-KOAHDL technique designs the KOA for the optimal hyperparameter selection. An extensive set of experiments was conducted to examine the improvised prediction outcomes of the ETFP-KOAHDL method. The significant contribution of the study includes:

- Development of the ETFP-KOAHDL technique, incorporating the KOA model with a hybrid DL approach, for effective traffic flow prediction.
- Implementing a min-max scaler for pre-processing the input dataset is followed by prediction utilizing a hybrid DL technique incorporating convolutional LSTM with an autoencoder.
- Utilizing the KOA to optimize hyperparameters results in improved prediction outcomes, validated via extensive investigation.

2 The Proposed Model

This article introduces an automated and accurate ETFP-KOAHDL system. The aim of the ETFP-KOAHDL model is to provide a reliable and efficient solution for real-time traffic flow prediction using advanced DL models. It follows three-phase processes: min-max normalization, CLSTM-AE-based prediction, and a KOA-based hyperparameter tuning model. Figure 1 depicts the entire process of ETFP-KOAHDL system.

Min-max Scaler

In this phase, the min-max normalization technique was applied to measure the data input into the relevant format (Sinsomboonthong, 2022). During the TFP, the min-max scaler is a pre-processing method used to convert raw traffic information into a standard form that is effectively used by the ML algorithms. Before using the min-max scaler, preparing and cleaning the information is necessary. This might include handling outliers, data quality issues, and other missing values. Then, min-max scaling is performed on the selected feature of the traffic data. The scaling method includes the following steps:

Identify the range for all the features. Generally, the scaled data ranges from 0 to 1. Subtract the minimum value from the data and divide by the range (the difference between the minimum and maximum values) for all the features. This scales the data into the desired range:

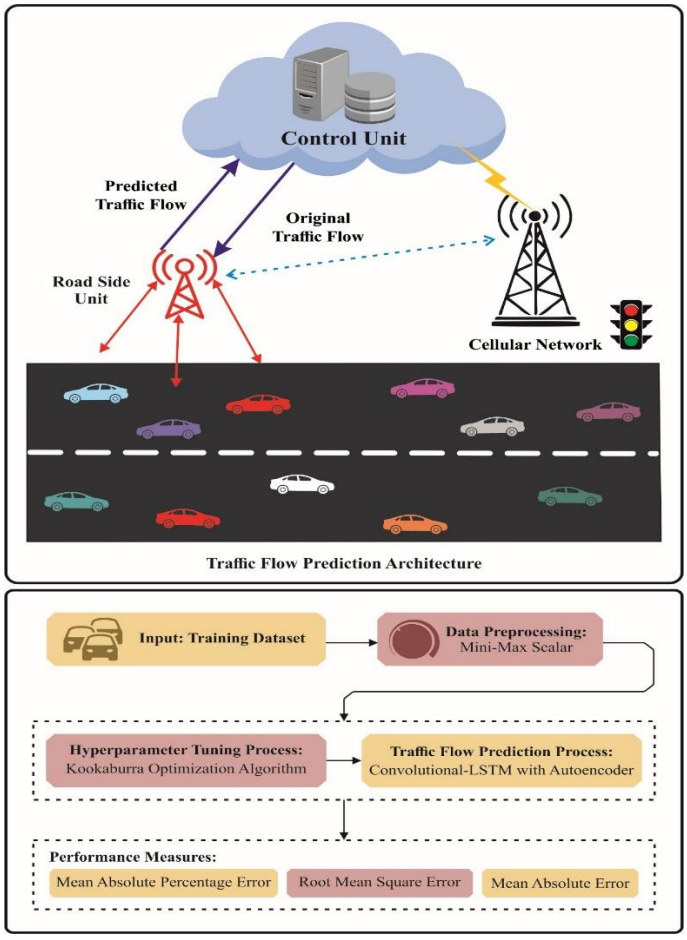

Figure 1: Overall Process of ETFP-KOAHDL Approach

$$x_{scaled} = \frac{(x - \min(x))}{(\max(x) - \min(x))} \quad (1)$$

Here, x represents a variable or a set of values, and $\min(x)$ and $\max(x)$ denote the minimum and maximum values of x, respectively in eq.1.

Min-max scaling is essential in predicting traffic flow since it ensures that various features are on a consistent scale, preventing specific features from dominating the model due to the large magnitude. This enables us to make fair and more accurate predictions, contributing to transportation system decision-making and better traffic management.

CLSTM-AE-based Predictive Method

In this work, the CLSTM-AE system is employed in order to forecast traffic flow. CNN can restore different irregular trends and is especially capable at obtaining sophisticated feature (Jogunola et al., 2022). The pooling layer is used to sample the mapping feature for separating high-order convolutional features, thus by reducing the dimension of feature's map. Next, the convolution layer is exploited to transform the input dataset into a mapping feature. The downsampling and extraction of feature models decrease the execution time, which creates it fit for the intended purpose.

The CNN output layer is inputted to BLSTM input, which performs as an input to AE layer. The BiLSTM-AE is used for data analysis and series prediction, while CNN extracts the relevant feature from the data. The typical LSTM is a popular alternative of RNN to combat the issue of gradient disappearing with the help of gain mechanisms (forget, input, and output) and memory units. The memory unit keeps the processing stage, an input gate states the input database to be retained, and the forget gate decides which data wants to be removed. However, an output from LSTM was distributed via an output gate. On the other hand, LSTM considers the prior state of information, thus dropping essential data from the subsequent state. Thus, a BiLSTM unites the data in the series forecast. However, AE is remarkably improved for learning representation to understand unsupervised input in the feature vector. It involves a decoding and encoding. Initially, the series of input is encoded before being decoded through internal representation. Consequently, the BiLSTM -AE learned the data's temporal dependency, positively affecting the anticipated output.

As opposed to the encoder BiLSTM, a one LSTM is exploited at the decoding AE for decreasing the complexity of proposed model. Also, the one LSTM learns from temporal dependency before the two FC layers proceed to the last predicted output. The data encoded from the BiLSTM-AE output can be decoded via LSTM-AE.

Assume the input vector $x_i^m = \{x_1, x_2, \cdots, x_n\}$, where x^m denotes the input vectors of $m \in M$, and n refers to the amount of standardized 30-minute units, serve x_{i^m} input vectors into the layer of CNN; the subsequent output is shown below:

$$y_{ij}^m = \sigma\left(b_j^m + \sum_{m=1}^{M} w_{m,j}^1 \, x_{i+m-1,j}^0\right) \quad (2)$$

In Eq. (2), y_{ij}^m is derived from the x_{ij}^m output vector. b_j^m indicates the bias for jth mapping features, the activation function is σ, the kernel's weight is w, and the index value is m. In such cases, the output vector for kth convolutional layer was assumed in Eq. (3).

$$y_{ij}^{m(k)} = \sigma\left(b_j^{m(k)} + \sum_{m=1}^{M} w_{m,j}^{m(k)} x_{i+m-1,j}^{0}\right) \quad (3)$$

The pooling layer is used to down sample an activation from feature map to minimize the computational cost and number of parameter. The max-pooling layer, characterized by (4), develops the biggest value to adjust the overfitting model.

$$P_{ij}^{m(k)} = \max_{r\epsilon R} y_{i\times T+rj}^{k-1} \quad (4)$$

In Eq. (4), the pooling size is y, and the length of an input dataset is T.

The max-pooling layer output is inputted to BiLSTM input layer, which is the gating unit. BiLSTM includes dissimilar gating mechanisms, and every gates were initiated when the memory units update the state, as follows in eq. (5) – (7).

$$i_t = \sigma\left(W_{pi}Pt + W_{hi}h_{t-1} + W_{ci}\cdot c_{t-1} + b_i\right) \quad (5)$$
$$f_t = \sigma\left(W_{pf}Pt + W_{hf}h_{t-1} + W_{cf}\cdot c_{t-1} + b_f\right) \quad (6)$$
$$O_t = \sigma\left(W_{op}Pt + W_{ho}h_{t-1} + W_{C0}\cdot c_{t-1} + b_o\right) \quad (7)$$

Here the activation function is σ, b shows the bias, and c_t indicates the cell state. The max pooling layer output at tth time is P_t. i_t, f_t, and 0_t denotes the input, forget and output gates. The HL h_t is updated at every tth step in backward and forward routes. By using the gating function of BiLSTM, the cell and hidden state (HL) are identified as shown in Eqs. (8) & (9), correspondingly

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \sigma\left(W_{pc}p_t + W_{hc}h_{t-1} + b_c\right) \quad (8)$$
$$h_t = O_t \cdot \sigma(c_t) \quad (9)$$

The BiLSTM output layer is concatenated for both the directions as demonstrated below (Eq. (10)):

$$y = \sigma\left(\overleftrightarrow{W_y h_t} + b_y\right) \quad (10)$$

The output of BiLSTM y is fed into the LSTM decoder input, where the resultant output $\hat{y} = \sigma\left(W_y h_t + b_y\right)$ signifies an input to the 2 FC layers (Eq. (11)).

$$d_i^k = \sum_j w_{ji}^k - 1\left(\sigma\left(\hat{y}_i^{k-1}\right) + b_i^{k-1}\right) \quad (11)$$

The CNN layer extracts spatial feature from the data input, and the BiLSTM -AE recognizes these features to discover time-based relations for the anticipated output.

KOA-based Hyperparameter Tuning Model

Finally, the KOA optimally chooses the CLSTM-AE model's hyperparameter values, thereby enhancing the predictive performance. KOA is a modern population-based optimizer system that provides the optimum solution for the problem in an iteration-based process using random searching in the solution space (Dehghani et al., 2023). The KOA was selected because of its capacity to search effectively via larger solution spaces. It is motivated by the cooperative foraging nature of kookaburras, making it appropriate for optimizing convolutional issues such as TFP. The KOA population comprises kookaburras located in solution space. It defines the value for the decision variable depend upon the location in the solution space; hence, every individual Kookaburra is the vector that can model a solution candidate to the problems. Kookaburras, collectively forming the KOA population matrix, can be mathematically illustrated by utilizing a matrix as depicted by Eq. (12). The kookaburra location is initialized randomly using Eq. (13).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix} \quad (12)$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb) \quad (13)$$

Here, the KOA population matrix is X, and the i^{th} Kookaburra (solution candidate) is X_i, $x_{i,d}$ is its d^{th} dimension in problem space, and the lower and upper boundaries of $the\ d^{th}$ parameter are lb_d and ub_d, correspondingly. A random value within [0,1] is r, the amount of individuals is N, and the amount of search spaces is m.

Consider the kookaburra location in the solution space corresponding to the candidate solution for the problem, which is evaluated as follows. In Eq. (14), the objective function is assessed according to i^{th}kookaburra, which is F_i, and the objective function is F.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (14)$$

The worst values for an evaluated objective function correspond to the worst member, and the best value corresponds fittest member. The objective function evaluated is an appropriate criterion to measure the candidate solution and population member.

In the following two phases, the KOA updates the kookaburra position: exploration and exploitation to improve the candidate solution derived from the Kookaburra's natural behaviour. Then, the population update process in the search range is performed.

Algorithm1: Pseudocode of KOA

Begin KOA
Input problem data: constraints, objective function, and variables.
Establish iterations (T) and KOA population size (N)
Produce the early population matrix randomly utilizing Eq. (2). $xi,d \leftarrow lbd + r \cdot (ubd - lbd)$
Assess the objective function.
    For t=1 to T
    For i=1 to N
Stage 1: hunting approach (exploration)
Define the candidate prey set utilizing Eq. (4). $CPi \leftarrow \{Xki : Fki < Fi\ and\ ki \neq i\}$
Select the prey for ith KOA member randomly.
Compute the novel location of ith KOA member utilizing Eq. (5). $xP1i,d \leftarrow xi,d + r \cdot (SCPi,d - I \cdot xi,d)$
Upgrade ith KOA member employing Eq. (6). $Xi = \{XP1i,\ \ Xi,\ \ FP1i < Fi\ \ else$
Stage 2: Confirming that the prey is murdered (exploitation)
Compute the novel location of ith KOA member by utilizing Eq. (7). $xP2i,d \leftarrow xi,d + (1-2r) \cdot (ubd - lbd)t$
Upgrade ith KOA member utilizing Eq. (8). $Xi = \{XP2i,\ \ Xi,\ \ FP2i < Fi\ \ else$
    End
Prevent the finest candidate solution up to now.
    end
Output the greatest quasi-optimal solution attained with the KOA.
End KOA.

Phase 1: Hunting Tactic (Exploration)

The Kookaburra is a predatory bird that consumes on small insects, birds, frogs, etc. Its strategy is attacking and selecting prey in its location. This procedure is the global search (exploration), representing the detailed scanning of solution space to prevent trapping in local optima and find the optimum area.

To mimic the hunting method of the Kookaburra, the location of other kookaburras, which contains best objective function values that regarded as a prey position for all the kookaburras. Thus, the available prey set for all the kookaburras is defined by Eq. (15).

$$CP_i = \{X_k : F_k < F_i \text{ and } k \neq i\}, where\ i = 1,2,\dots,N \text{ and } k \in \{1,2,\dots,N\} \quad (15)$$

In Eq. (15), the candidate prey for i^{th} kookaburras corresponds to CP_i; the individuals with the best objective function values than i^{th} kookaburras are X_k, and the objective function value is F_k.

It is noted that every individual arbitrarily chooses the prey and attacks it. The novel position for the Kookaburra is evaluated by Eq. (16) based on the Kookaburra's movement towards the prey during hunting. In such cases, when a value of objective function is enriched in the updated location, then it swaps the prior position of Kookaburra based on Eq. (17).

$$x_{i,d}^{P1} = x_{i,d} + r \cdot \left(SCP_{i,d} - I \cdot x_{i,d}\right), i = 1,2,\dots,N, and\ d = 1,2,\dots,m \quad (16)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & else \end{cases} \quad (17)$$

Based on initial stage of KOA, the new location of i^{th} kookaburra corresponds to X_i^{P1}, $x_{i,d}^{P1}$ is its d^{th} dimension, F_i^{P1} denotes the objective function value, random integer with normal distribution within [0,1] is r, the d^{th} dimension of prey selected for i^{th} kookaburras is $SCP_{i,d}$, a random integer range within $(1,2)$ is I, the amount of individuals is N, and the quantity of decision variables is m.

Phase 2: Confirming That the Prey is Assassinated (Exploitation)

After attacking, the Kookaburra pass on the prey with itself and ensures that the target is repeatedly killed by knocking it against the tree, holding it tightly between the crushes and claws, and eating it. These behaviours of the Kookaburra, which happen near the ground, result in slight variations in its location. The local search (exploitation) accomplishes the best solution near the potential areas and the solution attained. Algorithm 1 illustrates the steps involved in KOA.

In the KOA, a random location is evaluated by Eq. (18) to (19) simulate these behaviours of Kookaburra based on the movement near the hunting place. The displacement takes place haphazardly in the neighbourhood towards the centre of Kookaburra, with a radius equivalent to $\frac{(ub_d - lb_d)}{t}$. First, the radius of the neighbourhood is maximum; then, it becomes more minor such that the local search to converge towards the best solution can be more accurately performed during successive iterations. The calculated location for every Kookaburra changes its prior location if it enhances the values of an objective function based on following expression:

$$x_{i,d}^{P2} = x_{i,d} + (1 - 2r) \cdot \frac{(ub_d - lb_d)}{t}, i = 1,2,\dots,N, d = 1,2,\dots,m, and\ t = 1,2,\dots,T \quad (18)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & else \end{cases} \quad (19)$$

Depend upon the 2nd phase of KOA, the new location of i^{th} kookaburra corresponds to X_i^{P2}, $x_{i,d}^{P2}$ is its d^{th} decision variable, F_i^{P2} is a value of an objective function, the iteration and greatest count of algorithms are t and T.

3 Experimental Validation

In this part, the predictive study of the ETFP-KOAHDL system is tested under varying numbers of runs. Figure 2 represents the actual vs predicted traffic flow of ETFP-KOAHDL system under various epochs. The figure highlighted that the ETFP-KOAHDL method properly forecasted the traffic flow under all runs.

The proposed model is pretend utilizing the Python 3.6.5 tool on a PC with an i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

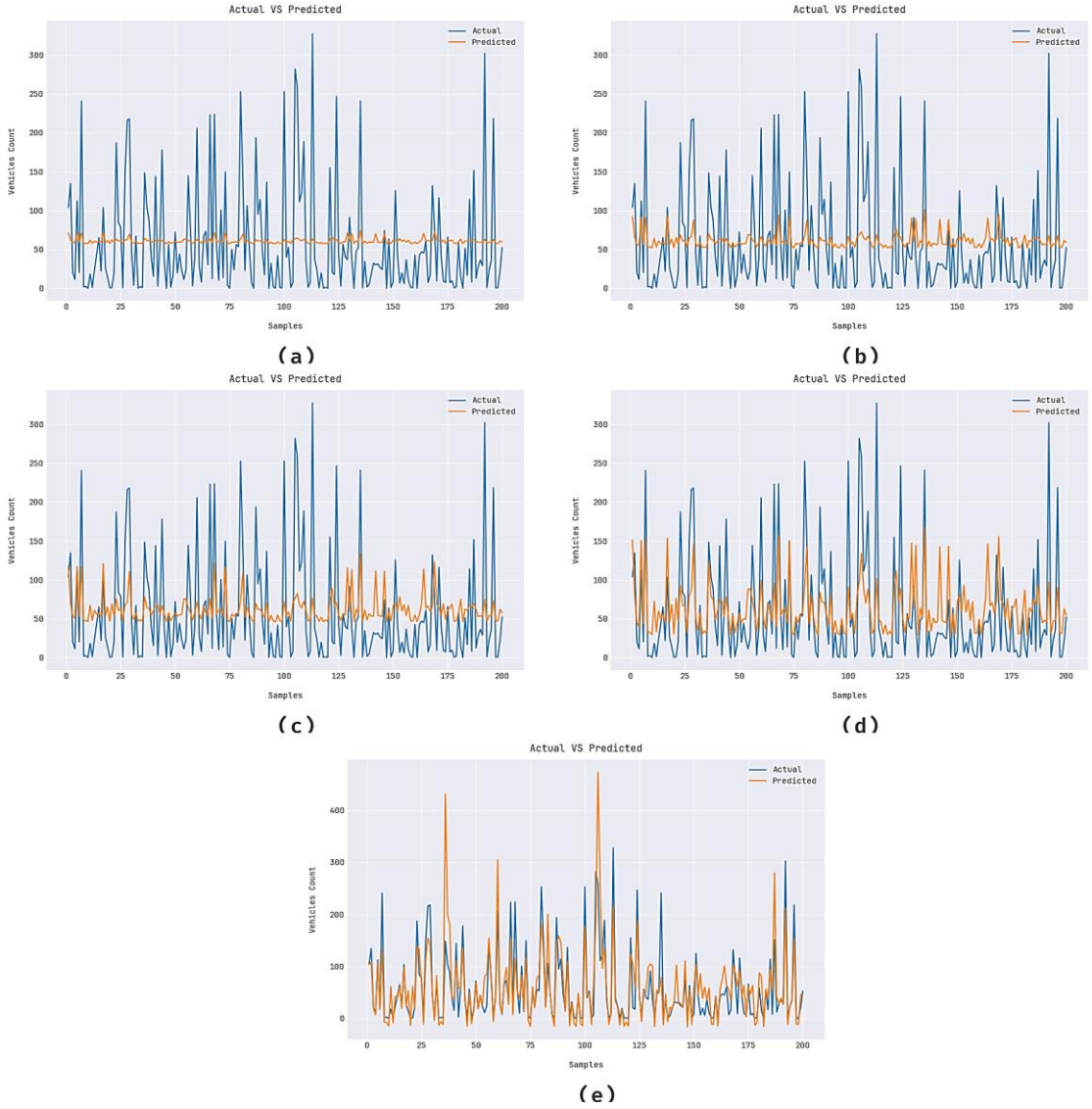

Figure 2: Prediction of Different Types of Runs

Figure 3 also exhibits an overview of ETFP-KOAHDL algorithm loss values across the TRA process. The decreasing trend in TRA loss over epochs represents that the model continuously improves its weights to reduce predictive errors on TRA and TES datasets. This loss curve considers how much the model fits the TRA dataset. Notably, the TRA and TES loss reliably diminish, exhibiting the model's learning of patterns in the dataset. Besides, it pointed out the model's adaptation in lessening differences involving predicted and the original TRA labels.

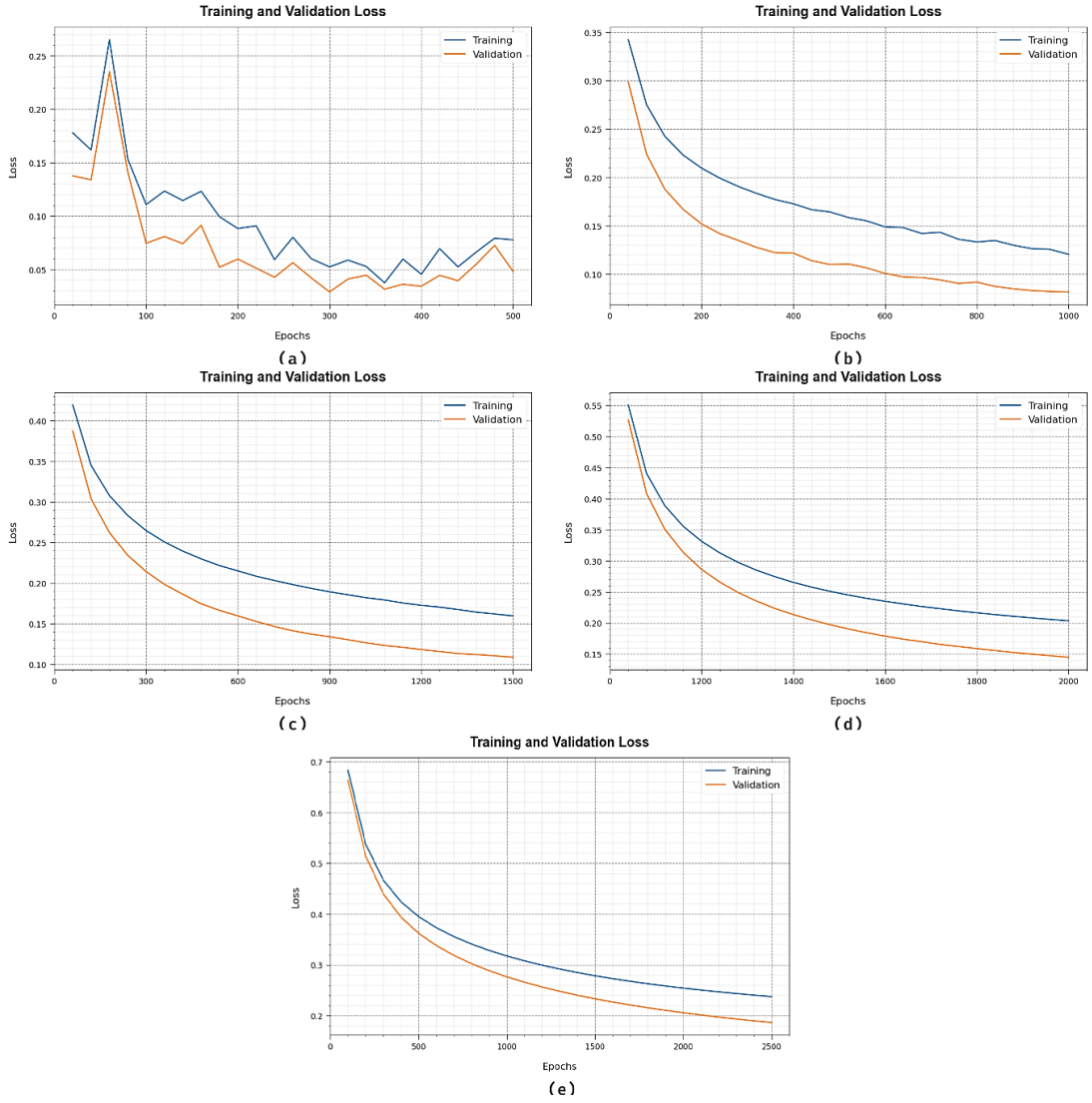

Figure 3: Loss Graph on Different Numbers of Epochs

The TFP outcomes of the ETFP-KOAHDL system with other DL techniques are portrayed in Table 1. The outcomes show that the ETFP-KOAHDL method obtains effective predictive results over other techniques (Ren et al., 2021). In Figure 4, a comparison of MAPE outcomes of the ETFP-KOAHDL methodology with recent approaches is given. The outcome indicates that the CNN model reaches worse outcomes with a maximum MAPE of 17.18%. Simultaneously, the LSTM and GRU models have reported relatively nearer MAPE values of 9.90% and 12.21%, respectively. Meanwhile, the CDLP model has obtained considerable performance with a MAPE of 6.63%. However, the ETFP-KOAHDL technique performed better with a minimal MAPE of 4.82%.

Table 1: Comparison Outcome of ETFP-KOAHDL Technique with Other Approaches

Model	MAPE (%)	MAE	RMSE
LSTM	09.90	06.81	09.61
GRU	12.21	10.01	13.44
CNN	17.18	22.25	33.76
CDLP	06.63	04.56	07.87
ETFP-KOAHDL	04.82	02.61	05.86

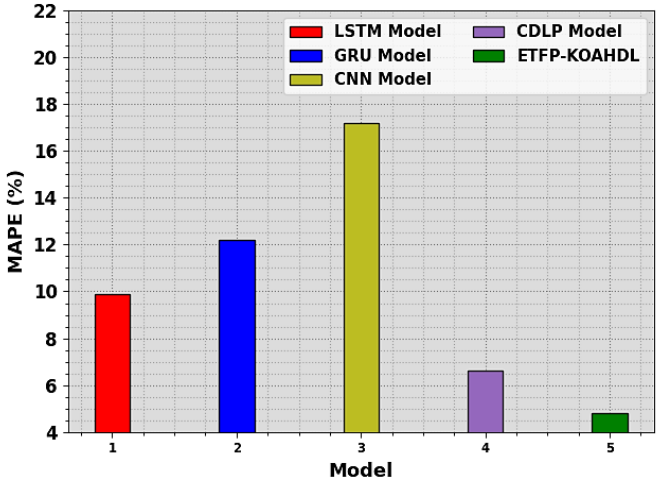

Figure 4: MAPE Analysis of ETFP-KOAHDL System with Other Models

Figure 5 compares the MAE analysis of the ETFP-KOAHDL technique with recent models. The obtained outcome exhibits that the CNN method attains poorer outcomes with a higher MAE of 22.25. Simultaneously, individually, the LSTM and GRU systems get slightly closer MAE values of 06.81 and 10.01. Then, the CDLP algorithm achieved remarkable performance with an MAE of 04.56. However, the ETFP-KOAHDL technique exhibited exceptional performance with a decreased MAE of 02.61.

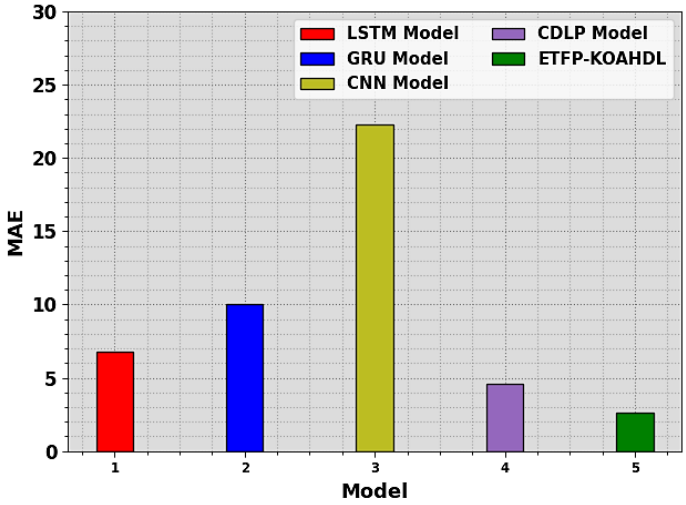

Figure 5: MAE Analysis of ETFP-KOAHDL System with Other Algorithms

Figure 6 shows a comparative RMSE analysis of the ETFP-KOAHDL methodology with recent models. The attained outcome specifies that the CNN model gets inferior outcomes with a greater RMSE of 33.76. Concurrently, the LSTM and GRU systems were acquired to report moderately closer RMSE values of 09.61 and 13.44, correspondingly. Besides, the CDLP algorithm obtains significant performance with an RMSE of 07.87. However, the ETFP-KOAHDL approach performs well, with a reduced RMSE of 05.86.

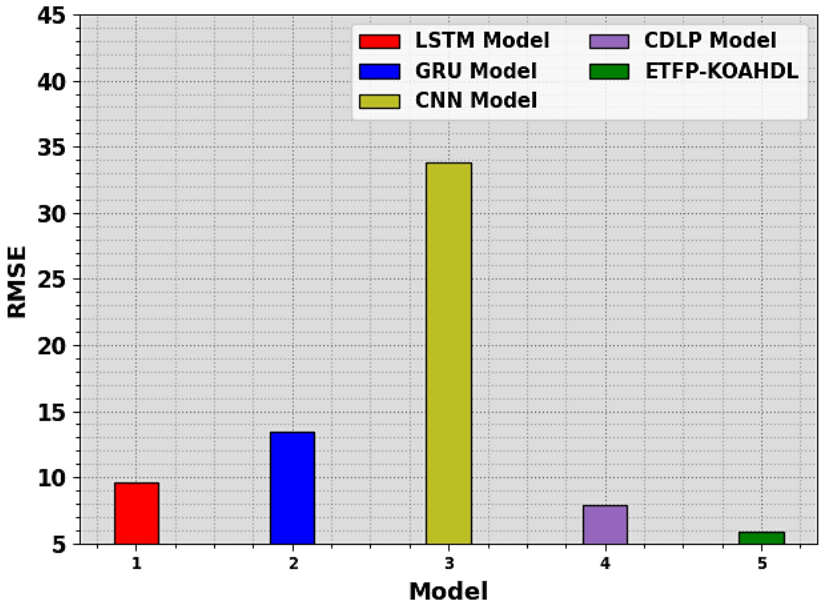

Figure 6: RMSE Analysis of ETFP-KOAHDL System with Other Methods

In Table 2 and Figure 7, a comparison of training time (TRT) analysis of the ETFP-KOAHDL methodology with recent models is represented. The obtained outcome indicated that the LSTM system gets worse outcomes with a maximum TRT of 50 min. Simultaneously, the GRU and CNN models have reported slightly closer TRT values of 46 min and 48 min, respectively. On the other hand, the CDLP algorithm obtains considerable performance with a TRT of 32 min. However, the ETFP-KOAHDL technique performed better with minimal TRT of 15 min. These achieved outcomes confirmed the enhanced predictive outcomes of the ETFP-KOAHDL technique over other models.

Table 2: TRT Outcome of ETFP-KOAHDL Algorithm with Other Models

Model	**Training Time (min)**
LSTM	50
GRU	46
CNN	48
CDLP	32
ETFP-KOAHDL	15

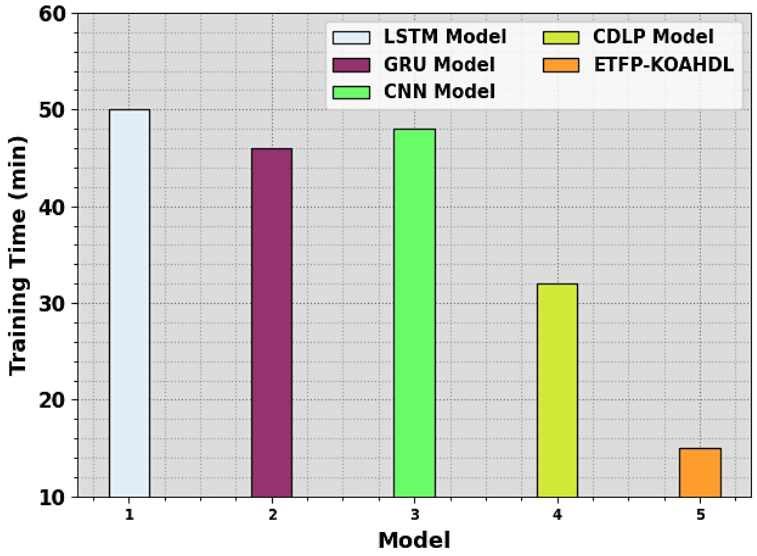

Figure 7: TRT Analysis of ETFP-KOAHDL System with Other Models

4 Conclusion

In this article, an automated and accurate ETFP-KOAHDL model is introduced. The objective of the ETFP-KOAHDL method is to provide a reliable and efficient solution for real-time traffic flow prediction using advanced DL models. It follows a three-stage process: min-max normalization, CLSTM-AE-based prediction, and a KOA-based hyperparameter tuning model. Primarily, the ETFP-KOAHDL technique follows a min-max scaler to pre-process the input data. The prediction process is also implemented using the CLSTM-AE model. Finally, the ETFP-KOAHDL technique designs the KOA for the optimal hyperparameter selection. An extensive set of experiments was conducted to examine the improvised prediction results of the ETFP-KOAHDL technique. The stimulation analysis stated that the ETFP-KOAHDL technique can be exploited to predict the traffic flow at a consecutive time, which helps in effective travel planning for autonomous vehicles. The limitations of the ETFP-KOAHDL technique are in its potential threats to precisely comprehend complex spatial-temporal patterns inherent in traffic data. Future work can focus on designing a feature selection strategy to handle the high dimensionality problem in real-time traffic data.

References

[1] Abdullah, S. M., Periyasamy, M., Kamaludeen, N. A., Towfek, S. K., Marappan, R., Kidambi Raju, S., ... & Khafaga, D. S. (2023). Optimizing traffic flow in smart cities: Soft GRU-based recurrent neural networks for enhanced congestion prediction using deep learning. *Sustainability*, *15*(7), 5949. https://doi.org/10.3390/su15075949

[2] Azgomi, H. F., & Jamshidi, M. (2018, November). A brief survey on smart community and smart transportation. In *2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI)* (pp. 932-939). IEEE. https://doi.org/10.1109/ICTAI.2018.00144

[3] Bharathi, D., Sopeña, J. M. G., Clarke, S., & Ghosh, B. (2024). Travel Time Prediction Utilizing Hybrid Deep Learning Models. *Transportation Research Record*, *2678*(4), 56-65. https://doi.org/10.1177/03611981231182964

[4] Chehri, A., & Mouftah, H. T. (2019). Autonomous vehicles in the sustainable cities, the beginning of a green adventure. *Sustainable Cities and Society*, *51*, 101751. https://doi.org/10.1016/j.scs.2019.101751

[5] Chen, X., Chen, H., Yang, Y., Wu, H., Zhang, W., Zhao, J., & Xiong, Y. (2021). Traffic flow prediction by an ensemble framework with data denoising and deep learning model. *Physica A: Statistical Mechanics and Its Applications*, *565*, 125574. https://doi.org/10.1016/j.physa.2020.125574

[6] Contreras, L.M., Bernardos, C.J., & Soto, I. (2011). RAMS: A Protocol Extension to PMIPv6 for Improving Handover Performance of Multicast Traffic. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *2*(2), 67-82.

[7] Dehghani, M., Montazeri, Z., Bektemyssova, G., Malik, O. P., Dhiman, G., & Ahmed, A. E. (2023). Kookaburra optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics*, *8*(6), 470. https://doi.org/10.3390/biomimetics8060470

[8] Duhaim, S. M., Taleb, D., & Ghadhban, Z. H. (2024). The Quality Learning System Interface Factor of Success Blended Learning. *International Academic Journal of Science and Engineering*, *11*(1), 262–264. https://doi.org/10.9756/IAJSE/V11I1/IAJSE1130

[9] Gao, H., Jia, H., Huang, Q., Wu, R., Tian, J., Wang, G., & Liu, C. (2024). A hybrid deep learning model for urban expressway lane-level mixed traffic flow prediction. *Engineering Applications of Artificial Intelligence*, *133*, 108242. https://doi.org/10.1016/j.engappai.2024.108242

[10] Gu, Y., Lu, W., Xu, X., Qin, L., Shao, Z., & Zhang, H. (2019). An improved Bayesian combination model for short-term traffic prediction with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 1332-1342. https://doi.org/10.1109/TITS.2019.2939290

[11] Hu, H. X., Hu, Q., Tan, G., Zhang, Y., & Lin, Z. Z. (2023). A multi-layer model based on transformer and deep learning for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *25*(1), 443-451. https://doi.org/10.1109/TITS.2023.3311397

[12] Jia, T., & Yan, P. (2020). Predicting citywide road traffic flow using deep spatiotemporal neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *22*(5), 3101-3111. https://doi.org/10.1109/TITS.2020.2979634

[13] Jogunola, O., Adebisi, B., Hoang, K. V., Tsado, Y., Popoola, S. I., Hammoudeh, M., & Nawaz, R. (2022). CBLSTM-AE: a hybrid deep learning framework for predicting energy consumption. *Energies*, *15*(3), 810. https://doi.org/10.3390/en15030810

[14] Kostić, S., Nikolić, N., & Malbašić, V. (2021). Development of A Methodology for Stability Monitoring of a Defense Embankment Loaded with Frequent Traffic: The Example of the Kovin Mine. *Archives for Technical Sciences*, *2*(25), 29–42. https://doi.org/10.7251/afts.2021.1325.029K

[15] Kuru, K., & Khan, W. (2020). A framework for the synergistic integration of fully autonomous ground vehicles with smart city. *IEEE Access*, *9*, 923-948. https://doi.org/10.1109/ACCESS.2020.3046999

[16] Lai, Z., Zhang, D., Li, H., Jensen, C. S., Lu, H., & Zhao, Y. (2023). Lightcts: A lightweight framework for correlated time series forecasting. *Proceedings of the ACM on Management of Data*, *1*(2), 1-26. https://doi.org/10.1145/3589270

[17] Li, Y., Chai, S., Ma, Z., & Wang, G. (2021). A hybrid deep learning framework for long-term traffic flow prediction. *IEEE Access*, *9*, 11264-11271. https://doi.org/10.1109/ACCESS.2021.3050836

[18] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. https://doi.org/10.48550/arXiv.1707.01926

[19] Liu, Y., Wang, X. K., Hou, W. H., Liu, H., & Wang, J. Q. (2022). A novel hybrid model combining a fuzzy inference system and a deep learning method for short-term traffic flow prediction. *Knowledge-Based Systems*, *255*, 109760. https://doi.org/10.1016/j.knosys.2022.109760

[20] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187-197. https://doi.org/10.1016/j.trc.2015.03.014

[21] Manipriya, S., Mala, C., & Mathew, S. (2020). A collaborative framework for traffic information in vehicular adhoc network applications. *Journal of Internet Services and Information Security, 10*(3), 93-109.

[22] Miao, H., Shen, J., Cao, J., Xia, J., & Wang, S. (2022). MBA-STNet: Bayes-enhanced discriminative multi-task learning for flow prediction. *IEEE Transactions on Knowledge and Data Engineering*, *35*(7), 7164-7177. https://doi.org/10.1109/TKDE.2022.3179781

[23] Mohammed Malik, C. K. (2022). Web Mining Using Improved Apriori Algorithm. *International Academic Journal of Innovative Research, 9*(1), 52–60. https://doi.org/10.9756/IAJIR/V9I1/IAJIR0917

[24] Nguyen, D. D., Rohacs, J., & Rohacs, D. (2021). Autonomous flight trajectory control system for drones in smart city traffic management. *ISPRS International Journal of Geo-Information*, *10*(5), 338. https://doi.org/10.3390/ijgi10050338

[25] Ren, C., Chai, C., Yin, C., Ji, H., Cheng, X., Gao, G., & Zhang, H. (2021). Short-Term Traffic Flow Prediction: A Method of Combined Deep Learnings. *Journal of Advanced Transportation*, *2021*(1), 9928073. https://doi.org/10.1155/2021/9928073

[26] Seuwou, P., Banissi, E., & Ubakanma, G. (2020). The future of mobility with connected and autonomous vehicles in smart cities. *Digital twin technologies and smart cities*, 37-52. https://doi.org/10.1007/978-3-030-18732-3_3

[27] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.

[28] Sinsomboonthong, S. (2022). Performance Comparison of New Adjusted Min-Max with Decimal Scaling and Statistical Column Normalization Methods for Artificial Neural Network Classification. *International Journal of Mathematics and Mathematical Sciences*, *2022*(1), 3584406. https://doi.org/10.1155/2022/3584406

[29] Wang, S., Cao, J., & Philip, S. Y. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, *34*(8), 3681-3700. https://doi.org/10.1109/TKDE.2020.3025580

[30] Wen, Y., Xu, P., Li, Z., Xu, W., & Wang, X. (2023). RPConvformer: A novel Transformer-based deep neural networks for traffic flow prediction. *Expert Systems with Applications*, *218*, 119587. https://doi.org/10.1016/j.eswa.2023.119587

[31] Xia, D., Chen, Y., Zhang, W., Hu, Y., Li, Y., & Li, H. (2024). RSAB-ConvGRU: A hybrid deep-learning method for traffic flow prediction. *Multimedia Tools and Applications*, *83*(7), 20559-20585. https://doi.org/10.1007/s11042-023-15877-x

[32] Yağız, E., Ozyilmaz, G., & Ozyilmaz, A. T. (2022). Optimization of graphite-mineral oil ratio with response surface methodology in glucose oxidase-based carbon paste electrode design. *Natural and Engineering Sciences*, *7*(1), 22-33. http://doi.org/10.28978/nesciences.1098655

[33] Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. https://doi.org/10.48550/arXiv.1709.04875

[34] Zhang, J., Zheng, Y., & Qi, D. (2017, February). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1). https://doi.org/10.1609/aaai.v31i1.10735

[35] Zhang, X., Zhao, Z., & Li, J. (2022). ARDE-N-BEATS: an evolutionary deep learning framework for urban traffic flow prediction. *IEEE Internet of Things Journal*, *10*(3), 2391-2403. https://doi.org/10.1109/JIOT.2022.3212056

[36] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, *21*(9), 3848-3858. https://doi.org/10.1109/TITS.2019.2935152

[37] Zhou, Q., Chen, N., & Lin, S. (2022). FASTNN: a deep learning approach for traffic flow prediction considering spatiotemporal features. *Sensors*, *22*(18), 6921. https://doi.org/10.3390/s22186921

[38] Zhou, T., Han, G., Xu, X., Han, C., Huang, Y., & Qin, J. (2019). A learning-based multimodel integrated framework for dynamic traffic flow forecasting. *Neural Processing Letters*, *49*, 407-430. https://doi.org/10.1007/s11063-018-9804-x

Author Biography

R.Y. Aburasain received the B.Sc. degree in computer science from the College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia, in 2008. She obtained the M.Sc. degree in computer science specializing in Intelligent Systems from the Department of Computer Science and Information Technology, Malaya University, Kuala Lumpur, Malaysia in 2012 and the Ph.D. degree in computer science from the University of Loughborough, Leicestershire, UK in 2020. She is currently working as an Assistant Professor in the Faculty of Engineering and Computer Science in Jazan University, Jazan, Saudi Arabia. Her research interests are Artificial intelligent, Machine Learning, and Deep Learning specially for Drone based applications including object detection, tracking, and recognition. Some of her research findings are used by Falcon Eye Drones Ltd, Dubai, UAE for detection and recognition of objects using high scale images captured by drones.