Natural Resources for Human Health

eISSN: 2583-1194

Original Research

Received 25 May 2026
Revised 29 June 2026
Accepted 17 July 2026

Natural Resources for Human Health 2026, Vol. 6 (10s): 239–255

KEYWORDS:
traffic congestion; denoising; contrast enhancement; transformer based fusion; CNN; normalization; temporal synchronization.

Multi-Model Traffic Congestion Prediction Using Diffusion Convolutional Recurrent Neural Networks with Transformer Based Fusion and Dual-Channel CNN for Multimodal Learning

Anil Kumar[1,2*], Raju Ranjan[1] , Shiv Kumar Verma[3]

[1]School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, 203201, India.
[2]Department of Computer Science, Deen Dayal Upadhyaya College, University of Delhi, Delhi, India.
[3]Department of Computer Science and Applications, Sharda University, Greater Noida, Uttar Pradesh – 201310, India.

ABSTRACT: Traffic congestion prediction focuses on evaluations upcoming road traffic states by analyzing past and real-time data such as vehicle velocity, traffic throughput and road occupancy. However, many existing models have limited adaptability to rapidly changing urban traffic conditions. To address these limitations, the proposed framework introduces a multimodal traffic prediction mechanism that integrates sensor time-series data with surveillance video streams. Initially, multi-source data are collected where traffic sensors provide time-series values (vehicle count, speed, and density) and roadside cameras provide continuous video frames. The sensors data are pre-processed using ImputeGAN (IGAN) for missing value imputation, Quantile Transformation for Normalization (QTN) and Time2Vec encoding (T2VE) to capture rush-hour patterns. Video frames are enhanced using Deep Unsupervised Blind Image Denoising (DUBID) to remove noise and Self-supervised Low-Light Enhancement (SSLLE) to improve visual quality. After preprocessing, both modalities are temporally aligned using a Sliding Window Synchronization (SWS). Feature-level multi-modal representations are learned using a Dual-Channel Convolutional Neural Network (DCCNN), where ResNet50 extracts detailed traffic density patterns and MobileNetV4 learns lightweight structural features. The combined features are fused through a Transformer-Based Fusion (TBF), to model relationships between numerical trends and vehicle congestion. Finally, the fused representation is fed into a Diffusion Convolutional Recurrent Neural Network (DCRNN) to predict the traffic state and the system classifies the road condition into four levels: low, normal, high, and heavy congestion. By using this methodology achieves the accuracy of 98.22%, precision of 96.48%, and the recall of 96.53%, demonstrating improved prediction efficiency, reliability with sparse data and responsiveness to real-world traffic dynamics.

1. INTRODUCTION

Traffic congestion prediction has become a research issue of critical concern in the context of urban intelligent transportation. With information about traffic, critical areas where congestion is likely to occur to identified and controlled in advance thus, vast economic losses, and road accidents can be avoided [1]. The traffic systems are enormous spatial networks whose structures are complicated and dynamic. Traditional methods are not enough to process the real-world traffic information properly. The predicted results may sometimes deviate from actual traffic conditions. Thus, the identification of credible patterns in large and complex traffic data is a core problem in the intelligent transportation systems. Furthermore, congestion control is another fundamental part of traffic management, and it requires the introduction of the appropriate approaches to control based on the properties of road structures to disperse congested areas within a short time and avoid the worsening of traffic [2].

The classical techniques of predicting traffic mostly use statistical and mathematical forecasting models including historic averaging, autoregressive models, and Kalman filtering. The methods assume linear depiction and steady traffic flow, which can hardly be achieved in the real world urban setting where the traffic conditions are constantly changing owing to accidental instances, peak hours, weather changes, and human driving habits [3]. They do not adequately represent nonlinear space-temporal correlations and frequently are inaccurate predictors. Although machine learning algorithms such as Support Vector Machine (SVM), k-Nearest Neighbor (KNN), and Random Forest (RF) have demonstrated superior performance, they suffer from limitations such as a heavy reliance on hand-crafted features and an inability to automatically capture complex patterns using diverse traffic data [4].

Deep learning methods have been widely used to solve these weaknesses in traffic analysis. The recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Graph Convolutional Networks (GCN), Gated Recurrent Units (GRU), can additionally encode road connectivity and long-range time dynamics. However, most of these methods continue to use one type of data- usually sensor data or video data- therefore decreasing resilience to noise, missing values, or uncertainty [5]. To avoid this limitation the combination of multimodal traffic data and deep learning models has been developed an effective approach to predict traffic and congestion. Below is the primary contribution of the suggested work.

- Multi-model prediction framework that combines Transformer based fusion and DCRNN to achieve reliable traffic congestion prediction.
- ImputeGAN, Quantile Transformation and Time2Vec improve sensor data reliability by handling missing values, normalization and temporal patterns, while DUBID is used for image denoising and SSLLE to improve video quality.
- SWS is used to synchronize the data temporally and ensure consistent multi-model correspondence.
- DCCNN (ResNet50+ and MobileNetV4) extracts complementary spatial features, while Transformer based fusion adaptively combines heterogeneous traffic representations.
- DCRNN learns dynamic spatio-temporal dependencies and accurately predicts and classifies traffic conditions.

The rest of this paper is organized as follows: Section 2 reviews related work on traffic prediction and congestion analysis; Section 3 presents the proposed multi-model framework and methodology; Section 4 discusses the simulation results and evaluating the performance of proposed model; and Section 5 concludes the work with future scope.

2. LITERATURE REVIEW

Several AI based traffic prediction and congestion analysis approaches have been developed, some of which are reviewed below.

shin et al. [6] have described a long short-term memory (LSTM) for Prediction of Traffic Congestion. Before prediction, the algorithm processes the traffic information by removing outliers with median absolute deviation and altering the temporal-spatial values. Mean absolute percentage error (MAPE) of the approach was found to be better than that of the comparison models with a value of about 5%.

Zaki et al. [7] have developed a Hidden Markov Models-contrast model (HMM-C) for Traffic congestion prediction. The model variability of the traffic patterns in terms of mean speed and contrast. The first method which researchers use to forecast traffic jams involves using HMM. The error of predicting patterns was 8.8%.

Li et al. [8] have introduced a Machine Learning and Fuzzy Comprehensive Evaluation based Framework (MF-TCPV) for Traffic Congestion Prediction. MF-TCPV categorizes congestion based on six levels, through a fuzzy overall assessment which was grounded by average speed, occupancy rate, and density of traffic-flow. The model achieved a prediction accuracy of 92.36%.

Natural Resources for Human Health

eISSN: 2583-1194

Gao et al. [9] have described a convolutional neural network (CNN) framework for traffic congestion analysis. The trained CNN was tested on a 2,400 image dataset after being trained on a 1,400 image dataset. The model had a precision of 98.74% and F-measure of 98.91% when predicting the level of congestion.

Ramana et al. [10] have described a Vision Transformer (ViT) Approach for Traffic Congestion Prediction in Urban Areas. A CNN teaches the feature maps to traffic images, and then a ViT operates on the feature maps and uses tokenization and projection schemes to predict congestion in the city, with the accuracy of 86.27%.

2.1 Research Gap

LSTM methods struggle to capture spatial road interactions and network connectivity [6]. HMM-C models have low adaptability to dynamic traffic changes [7]. MF-TCPV has limited ability in capturing difficult complex spatio-temporal traffic relationships [8]. CNN based image methods ignore numerical sensors data, leading incomplete traffic understanding [9]. CNN-ViT models are sensitive to noise and missing information [10]. To overcome this challenges, this paper employs DCCNN and DCRNN frameworks for predicting and classifying the traffic conditions.

3. PROPOSED METHODOLOGY

Intelligent Transportation Systems (ITS) integrate sensing, communication and computational technologies with transportation infrastructure to supplement traffic safety, mobility and operational efficiency. However, the existing approaches are limited by lack of sensor data and distorted surveillance images, which undermine the credibility of the models. To overcome these issues, the suggested model develops a multi-modal intelligent traffic prediction system, which combines heterogeneous data provided by IoT sensors with video surveillance data.

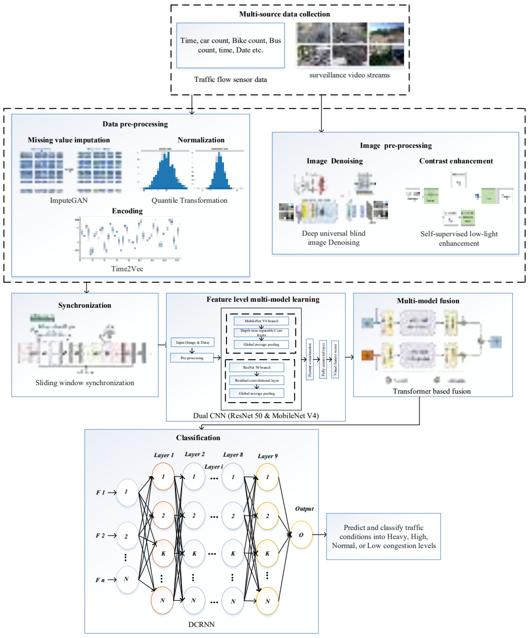

Figure 1. Architecture for Multi-model traffic prediction.

Figure 1 represents the traffic prediction framework using traffic sensors and surveillance video data. Multi-source data are first gathered for traffic prediction. These datasets include time-series values of vehicle count, speed and density variables derived by traffic flow sensors and continuous video frames of roadside cameras which represent congestion. The data sources are noisy and inconsistent, the pre-processing of both data modalities is performed separately. The sensor data is subjected to ImputeGAN (IGAN) to support missing-value imputation, and then Quantile Transformation Normalization (QTN) to standardize the distribution, and lastly, Time2Vec encoding (T2VE) to support time-aware encoding of sensor data to facilitate the model to learn rush-hour dynamics. At the same time, the frames of the video are pre-processed using Deep Unsupervised Blind Image Denoising (DUBID) for noise in the environment, as well as Self-Supervised Low-Light Enhancement (SSLLE) for minimized visibility in the dark. After that, the two modalities are synchronized in time through a sliding window process to make sensor readings and video frames reach the same time intervals. Then, the learning of multimodal representation at the features level is performed using a dual-channel CNN, in which the ResNet50 learns the patterns of spatial traffic density in images, and MobileNetV4 learns the patterns of lightweight structural features. The generated visual characteristics are combined with the encoded sensor characteristics and inputted into a TBF module implemented as a transformer that learns associations between numerical trends in congestion and visual vehicle aggregation. The DCRNN receives the combined representation which it uses to learn the roads connect throughout space and traffic patterns change over time to produce traffic predictions. It's categorizes the road conditions as low, normal, high, and heavy congestions.

i) Time series data pre-processing: ImputeGAN, QTN and Time2Vec

The sensor readings that are missing are reconstructed through the use of an ImputeGAN. Where a generator estimates the missing traffic values given the sensory information. A discriminator evaluates the validity of the generated values against actual traffic measurements [11]. This plays an adversarial role that enables the generator to learn realistic temporal patterns instead of using the simple interpolation. The adversarial task is defined in Eqn. (1).

$$\min G \max D\ V(D,G) = E_{x \sim P_{real}}[logD(x)] + E_{z \sim P_{noise}}\left[\log\left(1 - D\big(G(z)\big)\right)\right] \tag{1}$$

where G is a Generator network (predicts missing traffic value), D is a discriminator network (checks real or fake value), x denotes a real traffic sensor data, P_{real} is a distribution of real traffic data, z is a random noise input, P_{noise} is a noise distribution, $V(D,G)$ denotes adversarial loss. The final imputed value is obtained as Eqn. (2).

$$\hat{x}_m = G(x_{obs}, m) \tag{2}$$

Here m represents the missing value mask, x_{obs} denotes the available sensor readings, m is a mask (1=available, 0=missing) and $\hat{x}_m$ is reconstructed missing value.

Once missing values are imputed using QTN, empirical sensor data distribution is brought to the normal distribution as there are strong skewness values of quantitative values for traffic flow during peak and off-peak hours [12]. The observations are ordinally ranked, and hence a cumulative probability is assigned based on the empirical distribution, and then transformed to fit to a target distribution, the most common being a uniform or Gaussian distribution. This process reduces the heterogeneity in the variance and hence stabilizes the training of neural-networks. Formal transformation is defined as in Eqn. (3).

$$x_{scaled} = F_{target}^{-1}\big(F_{data}(x)\big) \tag{3}$$

where x is a original traffic value, $F_{data}(x)$ denotes the empirical cumulative distribution function, F_{target}^{-1} is a inverse target distribution.

Time2Vec encoding helps the model understand the periodicity of traffic in a day by transforming the individual timestamps to a vector of linear and periodic elements [13]. The linear term is proposed to represent the time-course, whereas sinusoidal terms are designed to represent periodicity (rush hours). The encoding is expressed as Eqn. (4).

$$t2v(t) = [w_0 t + \phi_0, \sin(w_1 t + \phi_1), \sin(w_2 t + \phi_2), \dots] \tag{4}$$

where t timestamp (min/hour), w denotes the learnable frequency weight, ϕ is a phase shift and $t2v(t)$ is an encoded time feature vector.

ii) Time series image pre-processing: Deep universal blind image denoising and Self-supervised low-light image enhancement

Video images often contain environmental noise, e.g. rain, fog, motion blur. A neural reconstruction model is conditioned to decouple a clean structural representation of an image without the aid of clean reference images [14]. The model attains image reconstruction by minimising the reconstruction error in Eqn. (5).

$$L = ||I_{clean} - F_\theta(I_{noisy})||^2 \tag{5}$$

Where I_{noisy} is a noisy traffic frame, I_{clean} denotes the reconstructed clean frame, F_θ is a neural network with parameters θ and L reconstruction loss.

Night-time frames are brightened using illumination estimation. Self-supervised low-light image enhancement, an image separated into two parts which include reflectance and illumination elements [15]. The process increases illumination levels while maintaining the integrity of existing structures using Eqn. (6).

$$I_{enhanced} = R(x).\gamma L(x) \tag{6}$$

where γ is an adaptive brightness factor, $L(x)$ is an illumination and $R(x)$ is a reflectance (object details).

iii) Temporal Synchronization: Sliding window synchronization

The system uses a sliding time window method to synchronize both modes which then show identical traffic times. The system divides continuous data into predetermined time slots and matches sensor data with the closest video footage to maintain temporal consistency in multimodal learning [16]. The synchronized windows in Eqn. (7).

$$X_t = \{x_{t-k+1}, \dots, x_t\} \tag{7}$$

Where X_t is a input sequence at time t, k denotes the window size and x_t is a traffic reading at time t.

iv) Feature level multi-model learning: Dual CNN (ResNet50 & MobileNetV4)

The system uses two separate CNN networks to analyze one traffic frame because the two networks will learn different visual information from the frame [17]. The network uses ResNet50 to detect intricate traffic patterns because it identifies both vehicle clustering and traffic congestion zones [18]. MobileNetV4 extracts lightweight structural features like lane boundaries and general road occupancy [19]. The network uses convolution filters which move across the image to identify important patterns. The convolution operation computes each feature value as Eqn. (8).

$$F_{i,j} = \sigma\left(\sum_{m,n} W_{m,n} \, . \, I_{i+m,j+n} + b\right) \tag{8}$$

where I is a input image, W denotes the convolution filter, b is a bias, σ denotes a activation function and $F_{i,j}$ is a extracted feature map value. The feature maps that ResNet-50 and MobileNet-V4 generate are then combined and merged with the sensor features that have been processed in advance, and a single, coherent multimodal representation is obtained and summarizes both roadway conditions and quantitative traffic.

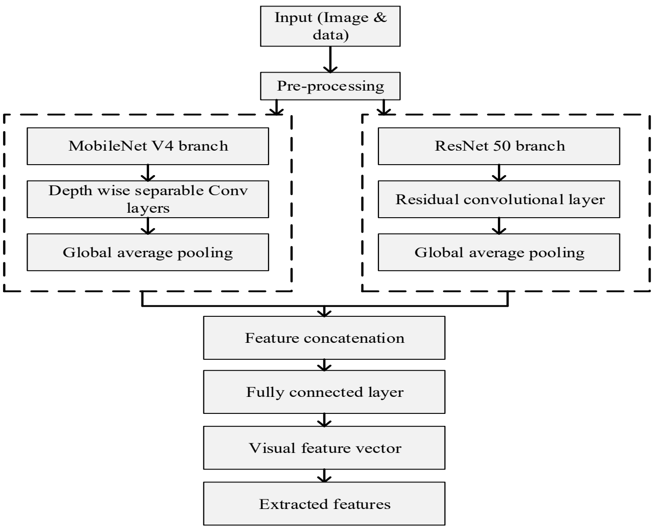

Figure 2. Architecture for Dual CNN (ResNet 50 & MobileNet V4).

Figure 2 shows that Architecture for Dual CNN (ResNet 50 & MobileNet V4). The Dual-Channel CNN employs two parallel branches ResNet50 and MobileNet V4 to extract deep features and lightweight visual features from the inputs. The extracted components are combined via feature concatenation and a Fully Connected Layer (FCL) to generate a feature vector for traffic analysis.

v) Multi-model Fusion: Transformer based fusion

The model uses an attention mechanism to combine different features because it needs to determine which video and sensor data provides more valuable information during specific accident scenarios and sensor operation periods [20]. The system first transforms features into three different vectors which include Q is a Query to indicate present information K denotes the Key for holding reference data and V is a Value to store actual feature information. The similarity between Query and Key determines the importance of each feature, and Softmax converts it into attention weights. The fused representation is computed as Eqn. (9).

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{9}$$

where d is the feature dimension. The weighted values are summed to achieve a single feature vector expressing may both sensor patterns and visual road conditions numerically.

vi) Classification: DCRNN

The DCRNN uses spatial traffic patterns and temporal traffic trends to forecast upcoming traffic conditions. The road network is represented as a graph in which each junction serves as a node and the road connections from the adjacency matrix A [21]. Diffusion convolution enables traffic data to spread through interconnected roadways which allows traffic jams at one intersection to affect nearby intersections. The recurrent unit processes input traffic feature X_t at every time step to update its memory together with the previous hidden state H_{t-1} in Eqn. (10).

$$H_t = \sigma(W_x X_t + W_h H_{t-1}) \tag{10}$$

where H_t is a hidden state at current time step (updated traffic memory), W_x is a weight matrix for current input features, W_h denotes the weight matrix and σ is an activation function. This also combines temporally evolved features like slow traffic build-up with the spatial propagation, assuming diffusion across the road graph is given in Eqn. (11).

$$X_{t+1} = D^{-1} A X_t W \tag{11}$$

Here $D^{-1}A$ normalizes traffic flow according to the number of connected roads which enables the model to forecast future traffic patterns across all locations. X_{t+1} denotes the predicted traffic state for next time step and W is a learnable spatial weight matrix. DCRNN training requires two objectives because it predicts traffic values before making classifications. The first prediction loss function between the actual traffic flow and the estimated traffic flow in Eqn. (12).

$$L_{pred} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{12}$$

where y_i and $\hat{y}_i$ is a actual and predicted traffic flow. After the prediction is converted to a category, cross-entropy loss is applied in Eqn. (13).

$$L_{cls} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{13}$$

where C is a total number of classes (4 congestion levels), c is a class index (1 to C), y_c denotes the true label for class c and $\hat{y}_c$ is a predicted probability of class c from softmax output. The predicted traffic value is finally converted into congestion categories: low, normal, high, or heavy.

Table 1. Layer Description for DCRNN approach.

Layer	**Output size**	**Parameter**
Input layer	(None, T, 256)	0
Diffusion convolution layer	(None, T, 256)	32,678
DCGRU layer 1	(None, T, 64)	49,152
DCGRU layer 2	(None, T, 64)	24,576
Fully connected layer	(None, 128)	8320
Output	(None, 4)	516

Table 1 describes the neural network using traffic data form 10 time steps across 5 road segments. Diffusion convolution captures spatial interdependencies between roads, but the DCGRU models temporal dynamics. T is a time steps of the input sequence. The decoder estimates the next state of traffic; the next fully-connected layer produces a value of absolute traffic, and a softmax classifier takes each road one of four congestion classes are low, normal, high or heavy.

4. RESULT AND DISCUSSION

The proposed work presented a multimodal traffic prediction framework that integrates sensor time-series data with surveillance video information. The system was testing using Python 3.12 on a setup with an Intel Core i5 processor, an NVIDIA GeForce GTX 1650 GPU, and 16GB of RAM, ensuring smooth and efficient execution. Table 2 depicts the Simulation parameters for proposed approaches.

Table 2. Simulation parameters for proposed ResNet50, Dual channel CNN and DCRNN.

Methods	Parameters	Range
ResNet50	Default size	224
	Minimum size	197
	Stride	(2,2)
	Target size	(224,224)
	Axis	0
Dual channel CNN	Batch size	206
	Learning rate	0.005
	Epochs	250
	Axis	1
	Activation	Relu
	Stride	2
	Pool size	2
	Factor	0.5
DCRNN	Dimension	1
	Number of nodes	5
	Input dimensions	1
	Output dimensions	1
	Hidden dimensions	32
	Sequence length	10

Table 2 summarizes the simulation parameters used for the proposed techniques, including batch size, learning rate, dimensions, input dimensions, output dimensions, sequence length, factor, pool size, axis, epochs and target size.

4.1.Dataset description

The Kaggle Traffic Prediction Dataset (85 kB) contains time-series data which tracks traffic volume together with traffic congestion indicators [24]. The dataset includes timestamp data and day information and vehicle count numbers and congestion assessment criteria which support machine learning and deep learning traffic prediction. The Mendeley Traffic Congestion Dataset (396 MB) contains 500×500 pixel images of traffic scenes labelled as congested or uncongested, designed for computer vision tasks such as CNN-based congestion detection [25].

Table 3. Data split for training and testing in Traffic Congestion Prediction.

Dataset	Class	Overall samples	Total samples	Training	Testing
Traffic congestion Dataset	0	596	2384	1907	477
	1	596			
	2	596			
	3	596			

Table 3 presents the total number of samples used in this study. The sample of image is 2384 and sample of data is 2384, which are fused to split the training and testing process. The dataset for traffic congestion prediction is split such that 80% of the samples are used for training and the remaining 20% are used for testing.

Table 4. Sample data for pre-processed outcome.

Original data				
Car Count	**Bike Count**	**Bus Count**	**Truck Count**	**Total**
14	0	0	34	48
17	5	0	23	nan
19	1	1	18	39
15	5	nan	12	33
13	2	1	26	42
nan	3	0	32	50
Data imputation				
14	0	0	34	48
17	5	0	23	45
19	1	1	18	39
15	5	1	12	33
13	2	1	26	42
15	3	0	32	50
15	0	1	13	29
Normalized data				
-1.269664385	-5.199337583	-5.199337583	1.602210318	-0.908974922
-0.935852086	-0.586473213	-5.199337583	0.637923432	-1.029375402
-0.716838953	-1.442543358	-0.798768904	0.278301688	-1.295362537
-1.135284209	-0.586473213	-0.798768904	-0.129582314	-1.689670077
-1.397836529	-1.15461469	-0.798768904	0.846278503	-1.15461469
-1.135284209	-0.963422907	-5.199337583	1.41129327	-0.837338361
-1.135284209	-5.199337583	-0.798768904	-0.053972647	-2.11966823

The sample data of pre-processed results is given in Table 4. The gaps in the traffic data are used to imputation and then all the attributes pertaining to vehicles are normalized to scaled values and thus, make the data responsive to further model training.

Original image | Denoised image | Contrast enhancement image

Figure 3. Sample and pre-processed images.

Figure 3 shows the Sample and pre-processed images. The first, second and third columns represent the original, denoised and enhanced image, respectively.

4.2. Comparison of traffic congestion prediction process

Proposed DCRNN approach is compared with existing algorithms, including the Long Short Term Memory (LSTM), Deep Brief Network (DBN), Deep Neural Network (DNN), Gated Recurrent Unit (GRU).

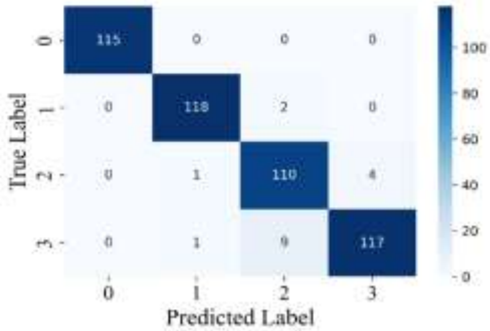

Figure 4. Confusion matrix for proposed approach.

Figure 4 shows the confusion matrix used to evaluate the classification model's performance. The model correctly predicts 115 class 0 (heavy), 118 class 1 (high), 110 class 1 (normal) and 117 class 2 (low) cases, while only 17 instances are misclassified.

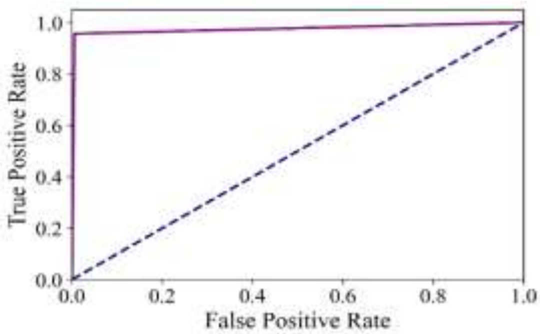

Figure 5. ROC curve for proposed approach.

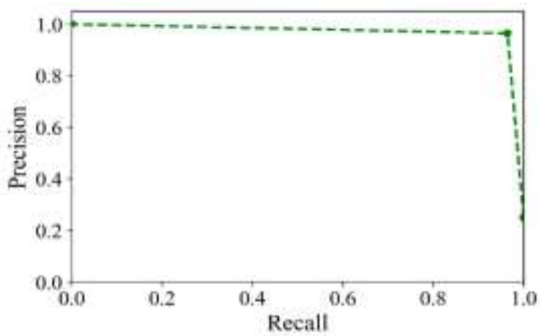

Figure 6. Precision-recall curve for proposed approach.

Natural Resources for
Human Health
eISSN: 2583-1194

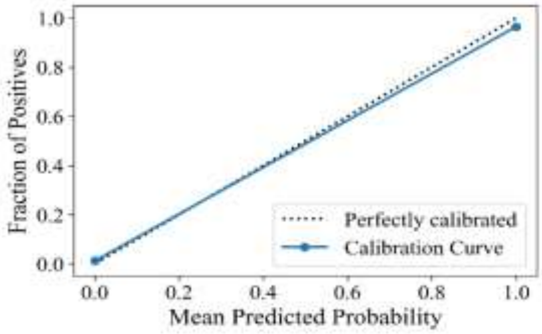

Figure 7. Calibration curve for proposed approach.

Figure 5 presents the ROC curves for all classes, demonstrating the classification performance of the proposed model. The AUC value is near to 1 (97%), so this proposed DCRNN provide a better traffic congestion classification. Figure 6 displays the precision-recall curve of the proposed model, demonstrating that it maintains high precision at 1.00 across most recall values, starting from a recall of 0.0. Figure 7 presents the calibration curve for the proposed model, which closely tracks the diagonal reference line; the fraction of positives increases from 0.02 to nearly 1.0 as the predicted probability rises from 0.0 to 0.97.

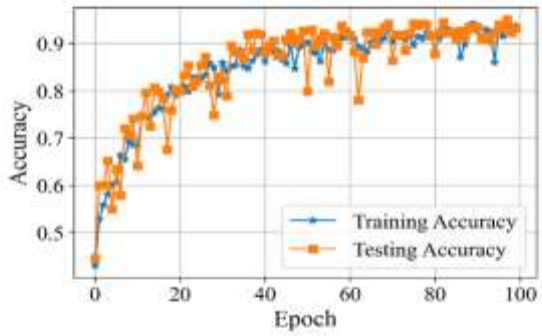

Figure 8. Training and testing accuracy curve.

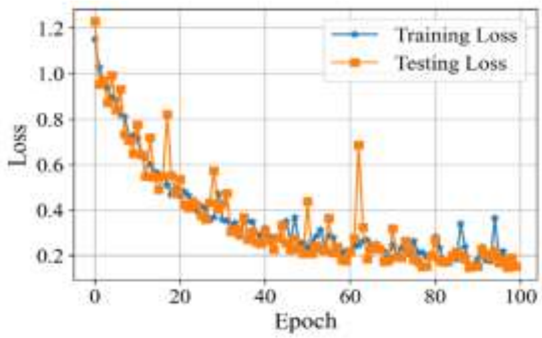

Figure 9. Training and testing loss curve.

Figure 8 displays the training and testing accuracy curves. The training accuracy begins at approximately 0.43 during the initial stages (epochs) and improves to nearly 0.93 by the 100th epoch; this indicates successful learning and a reduction in overfitting. As shown in Figure 9, the loss continues to decrease as the epoch's progress; specifically, the training loss drops from 1.18 to approximately 0.20. This points to the proper optimization of the model and its robust generalization performance.

Natural Resources for
Human Health
eISSN: 2583-1194

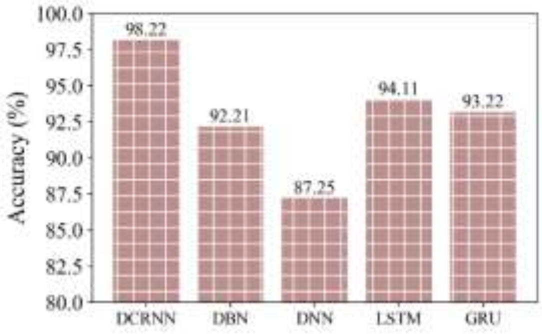

Figure 10. Accuracy comparison of DCRNN with other current algorithms.

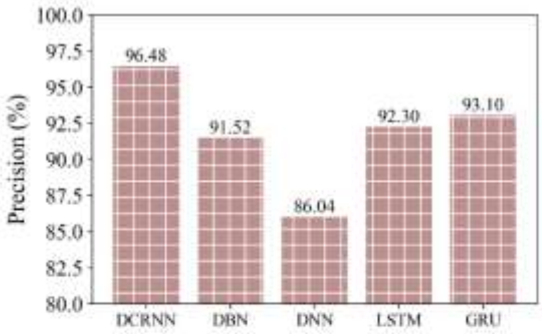

Figure 11. Precision comparison of suggested and different models.

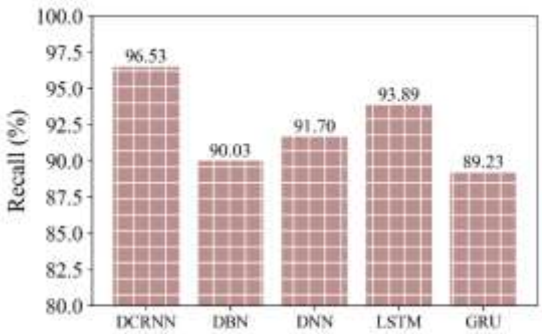

Figure 12. Recall comparison of DCRNN and the current algorithm.

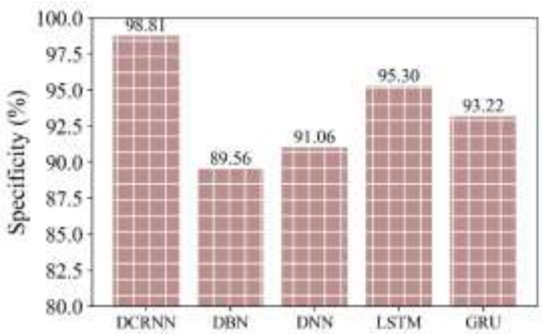

Figure 13. Comparison of specificity for DCRNN and the current algorithm.

The performance of the proposed model in terms of accuracy, precision, recall, and specificity is compared to the existing models as depicted in Figures 10-13. The proposed DCRNN has the highest accuracy of 98.22%, whereas the DBN, DNN, LSTM, and GRU models have lower accuracy values. The DCRNN also attained a precision of 96.48% and a recall of 96.53%

and specificity of 98.81% thus better than all baseline approaches. Generally, DCRNN is capable of detecting actual traffic congestion events; furthermore, due to its high specificity, it effectively identifies false scenarios related to normal traffic conditions.

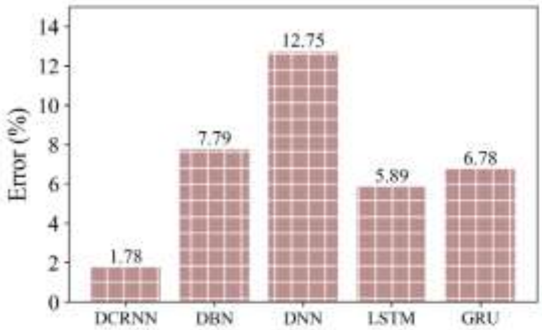

Figure 14. Evaluation of error for proposed and existing models.

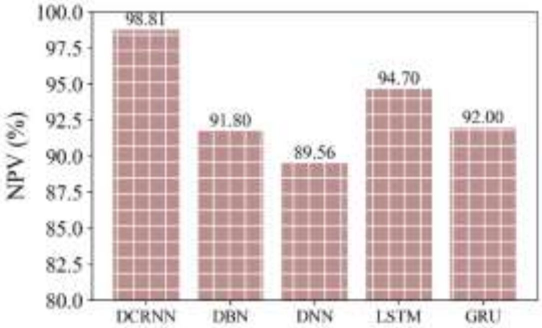

Figure 15. Comparison of NPV.

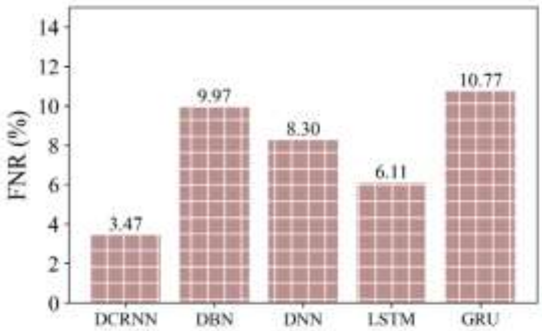

Figure 16. FNR comparison of DCRNN with Existing models.

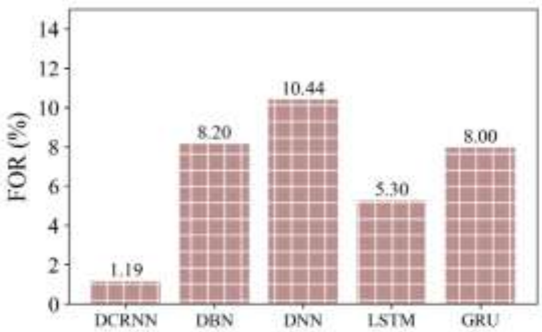

Figure 17. FOR comparison of proposed with Existing models.

Figures 14-17 show the performance of the proposed DCRNN model as compared to the existing models such as Error rate, NPV, FNR and FOR. The Error of 1.78% is achieved by the DCRNN model, which is significantly lower than that of the existing models. The NPV, the DCRNN has a value of 98.81% as compared to other strategies. The FNR is 3.47% which means that a very small amount of congestive events are missed. Moreover, the DCRNN has the lowest FOR at 1.19%. It's especially troublesome in Intelligent Transportation Systems, where a missed congestion event may lead to timely traffic control responses.

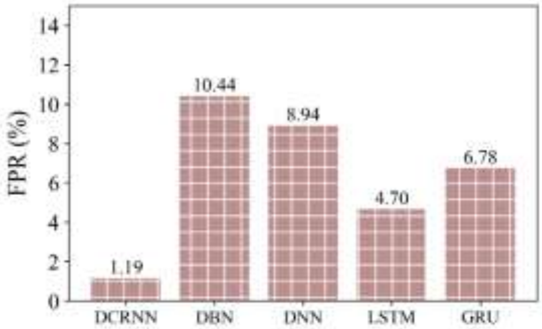

Figure 18. Comparison of FPR.

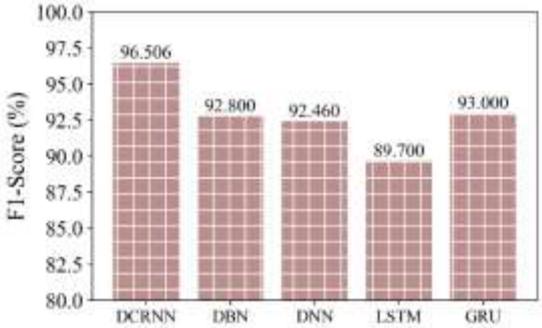

Figure 19. Comparison of F1-score for proposed and existing models.

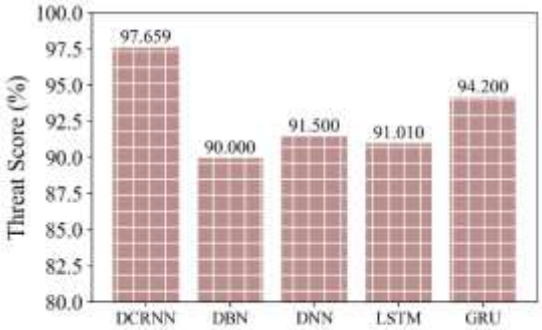

Figure 20. Comparison of Threat score.

The comparative analysis of FPR, F1-score, and Threat Score of proposed DCRNN with existing models were presented in Figures 18, 19 and 20. The DCRNN attains a FPR of 1.19% which shows that the device is prone to placing false alarms of congestion. An F1-score of 96.50% is achieved simultaneously. Similarly, the Threat Score gets 97.65% confirming the accurate observation of true congestion events. Together, these measures prove that the DCRNN is better than the DBN, DNN, LSTM, and GRU networks, which makes it a potential choice in the context of implementing the architecture in the real-time traffic monitoring system.

Table 5. Ablation study for the proposed approaches.

Performance metrics (%)	Accuracy	Precision	Recall
IGAN+QTN+T2VE+DCRNN	89	87	88.3
DUBID+SSLLIE+ DCRNN	89.1	89.3	90
SWT+ DCRNN	87.2	88.3	88.4
DCCNN (ResNet50)+ DCRNN	90.7	91.2	90.4
DCCNN (MobileNetV4)+ DCRNN	91.4	92.7	90.8
DCCNN (ResNet50+MobileNetV4)+TBF+ DCRNN	92.8	93.47	94.5
IGAN+QTN+T2VE+DCRNN+ DUBID+SSLLIE+DCCNN (ResNet50+MobileNetV4)+TBF+ DCRNN	98.22	96.48	96.53

As shown in the table 5 is illustrated the ablation study for the proposed approaches. Complete interconnectedness of all elements is lead to a best-performance outcome hence prove that the multimodal architecture and learning strategy significantly increase the reliability of traffic congestion prediction.

Table 6. State-of-art for proposed with existing models.

Performance metrics (%)	Proposed [DCRNN]	Gaussian Mixture Model (GMM) [22]	Fusion-based machine learning approach [23]
Accuracy	98.22	96	93.3
Precision	96.48	92	84.4
Recall	96.53	93	91.1

Table 6 demonstrates a State-of-art for the proposed with existing models. The proposed model performs with significantly greater reliability and accuracy than the method currently employed in traffic congestion prediction.

5. CONCLUSION

The suggested framework anticipates the state of traffic based on the collective learning of sensor indicators on vehicles and surveillance videos, thus, obtaining both the quantitative and visual road conditions. The vehicle congestion is captured both visually by the use of cameras and through vehicle sensors that provide vehicle count, vehicle speed and density. The individual modalities are pre-processed, sensor data is imputed with ImputeGAN, standardised distribution with Quantile Transformation Normalization and Time2Vec for temporal awareness. Video denoised using DUBID and low-light enhanced employed SSLLE. The two modalities are then aligned in time using a SWS. Feature level multi-model learning through DCCNN, (ResNet50 extracts the spatial patterns and MobileNetV4 for lightweight structural features) and combined with a transformer based fusion module that generates the relationship between visual and numerical traffic dynamics. The merged representation is then fed through a DCRNN which not only gets the spatial connectivity of the roads but also the temporal evolution of traffic. The system categorizes the levels of congestion into four levels, namely low, normal, high, and heavy. The proposed approaches attained an accuracy of 98.22%, a precision of 96.48%, a recall of 96.53%, a threat score of 97.65% and the error rate is of 1.78%. Improving generalizability may require retraining the model for specific urban situations and traffic patterns. The future directions include cross-model representation learning, modality dropout in order to reduce cases of missing data and model-compression methods including knowledge distillation and quantisation so that the model can be deployed in real time.

ACKNOWLEDGEMENT

AUTHOR CONTRIBUTIONS

Anil Kumar: Conceptualization, Investigation, Supervision

Raju Ranjan: Methodology, Laboratory Analysis, Writing—Review & Editing

Shiv Kumar Verma: Data Curation, Formal Analysis

FUNDING

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

CONFLICT OF INTEREST

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

AVAILABILITY OF DATA AND MATERIAL

Not applicable

CODE AVAILABILITY

Not applicable

COMPLIANCE WITH ETHICAL STANDARDS

This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

CLINICAL TRIAL NUMBER

Not applicable

ETHICS STATEMENT

Not applicable

REFERENCES

[1] Lin, C., Hu, X., Zhan, Y., Hao, X., 2024. *MobileNetV2 with spatial attention module for traffic congestion recognition in surveillance images*. Expert Systems with Applications 255, 124701.

[2] Liu, B., Lam, C.T., Ng, B.K., Yuan, X., Im, S.K., 2024. *A graph-based framework for traffic forecasting and congestion detection using online images from multiple cameras*. IEEE Access 12, 3756–3767.

[3] Ahn, H., Lee, Y.H., 2022. *Vehicle classification and tracking based on deep learning*. Journal of Web Engineering 21(4), 1283–1294.

[4] Rafique, A.A., Al-Rasheed, A., Ksibi, A., Ayadi, M., Jalal, A., Alnowaiser, K., Park, J., 2023. *Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images*. IEEE Access 11, 2993–3007.

[5] Barbosa, R., Ogobuchi, O.D., Joy, O.O., Saadi, M., Rosa, R.L., Al Otaibi, S., Rodriguez, D.Z., 2023. *IoT-based real-time traffic monitoring system using image sensors by sparse deep learning algorithm*. Computer Communications 210, 321–330.

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

[7] Zaki, J.F., Ali-Eldin, A., Hussein, S.E., Saraya, S.F., Areed, F.F., 2020. *Traffic congestion prediction based on hidden Markov models and contrast measure*. Ain Shams Engineering Journal 11(3), 535–551.

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

[9] Gao, Y., Li, J., Xu, Z., Liu, Z., Zhao, X., Chen, J., 2021. *A novel image-based convolutional neural network approach for traffic congestion estimation*. Expert Systems with Applications 180, 115037.

[10] Ramana, K., Srivastava, G., Kumar, M.R., Gadekallu, T.R., Lin, J.C.W., Alazab, M., Iwendi, C., 2023. *A vision transformer approach for traffic congestion prediction in urban areas*. IEEE Transactions on Intelligent Transportation Systems 24(4), 3922–3934.

[11] Yang, H., Lian, C., Xu, B., Ding, R., Zhao, P., Zeng, Z., 2025. *Self-supervised latent feature-guided multi-step diffusion model for electricity theft detection with imbalanced and missing data*. IEEE Transactions on Smart Grid 16, 2439–2450.

[12] Skubleny, D., Ghosh, S., Spratlin, J., Schiller, D.E., Rayat, G.R., 2024. *Feature-specific quantile normalization and feature-specific mean–variance normalization deliver robust bi-directional classification and feature selection performance between microarray and RNA-seq data*. BMC Bioinformatics 25(1), 136.

[13] Geng, D., Wang, B., Gao, Q., 2023. *A hybrid photovoltaic/wind power prediction model based on Time2Vec, WDCNN and BiLSTM*. Energy Conversion and Management 291, 117342.

[14] Lyu, Z., Chen, Y., Sun, H., Hou, Y., 2024. *A dual fusion deep convolutional network for blind universal image denoising*. Signal Processing: Image Communication 120, 117077.

[15] Liang, J., Xu, Y., Quan, Y., Shi, B., Ji, H., 2022. *Self-supervised low-light image enhancement using discrepant untrained network priors*. IEEE Transactions on Circuits and Systems for Video Technology 32(11), 7332–7345.

[16] Xia, W., Wei, C., Li, Z., Wang, X., Zou, X., 2022. *NetSync: A network adaptive and deduplication-inspired delta synchronization approach for cloud storage services*. IEEE Transactions on Parallel and Distributed Systems 33(10), 2554–2570.

[17] Tian, C., Xu, Y., Zuo, W., Du, B., Lin, C.W., Zhang, D., 2021. *Designing and training of a dual CNN for image denoising*. Knowledge-Based Systems 226, 106949.

[18] Elpeltagy, M., Sallam, H., 2021. *Automatic prediction of COVID-19 from chest images using modified ResNet50*. Multimedia Tools and Applications 80(17), 26451–26463.

[19] Huang, X., Yang, S., Xiong, A., Yang, Y., 2024. *Enhanced YOLOv8 with VTR integration: A robust solution for automated recognition of OCS mast number plates*. IEEE Access 12, 179648–179663.

[20] Hu, J.F., Huang, T.Z., Deng, L.J., Dou, H.X., Hong, D., Vivone, G., 2022. *Fusformer: A transformer-based fusion network for hyperspectral image super-resolution*. IEEE Geoscience and Remote Sensing Letters 19, 1–5.

[21] Tang, Y., He, H., Wang, Y., 2024. *Hierarchical vector transformer vehicle trajectories prediction with diffusion convolutional neural networks*. Neurocomputing 580, 127526.

[22] Chetouane, A., Mabrouk, S., Jemili, I., Mosbah, M., 2022. *Vision-based vehicle detection for road traffic congestion classification*. Concurrency and Computation: Practice and Experience 34(7), e5983.

[23] Alnawayseh, S.E., Al-Sit, W.T., Ghazal, T.M., 2022. *Smart congestion control in 5G/6G networks using hybrid deep learning techniques*. Complexity 2022, 1781952.

[24] Hasibullahaman, 2026. *Traffic prediction dataset*. Kaggle. Available at: https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset/data (accessed 24 February 2026).

[25] Bedadabekelemendeleydata, 2026. *Traffic prediction dataset*. Mendeley Data. Available at: https://data.mendeley.com/datasets/wtp4ssmwsd (accessed 24 February 2026).