SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I11P119

Volume 13 Issue 11, 316-329, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Long Term Traffic Congestion Prediction using Hybrid Deep Learning

Mohandu Anjaneyulu[1], Kaveripakam Thyagarajan[2]

[1]*School of Technology, Computer Science Engineering, The Apollo University, Chittoor, India.*
[2]*Department of Computer Science Engineering, Sri Venkateswara College of Engineering and Technology, Chittoor, India.*

[1]*Corresponding Author: manjaneyulu.cse@gmail.com*

Received: 26 June 2026 Revised: 30 October 2026 Accepted: 06 November 2026 Published: 27 November 2026

Abstract *- Precise estimation and forecasting of traffic congestion across roadway sections are essential for effective route planning and intelligent traffic management in urban areas. Rapid urbanization and the continuous growth in the number of vehicles keep typically expanded traffic congestion, making traffic monitoring and forecasting important components of Intelligent Transportation Systems (ITS). Accurate forecasts of traffic congestion during peak periods, recurring traffic conditions, and unexpected incidents can assist traffic management systems in identifying congestion patterns and directing vehicles toward suitable alternative routes. To address the temporal characteristics of Long-Term Traffic Congestion (LTTC) forecasting, this research recommends a unique hybrid Xception–Convolutional Long Short-Term Memory (ConvLSTM)–K-Nearest Neighbor (XCL-KNN) model for predicting traffic congestion over the next 17 hours. Traffic images of roadway sections in Bangalore, Karnataka, were collected from Google Maps using Selenium-based automated data acquisition. The Xception network is employed as a spatial feature extractor to learn high-level traffic representations from individual traffic images using depthwise separable convolutions and nonlinear activation functions. The extracted feature sequences are subsequently supplied to a ConvLSTM network, which captures both spatial and temporal dependencies among successive traffic observations. The resulting spatio-temporal representations are then provided to the K-Nearest Neighbor (KNN) classifier, which identifies the most similar historical traffic patterns using Euclidean distance and determines the congestion class based on the neighboring observations. The proposed model XCL-KNN generates multi-horizon traffic congestion predictions from one hour up to 17 hours ahead. Model conduct is calculated proving F1-score, recall, confusion matrix, precision, and accuracy. The suggested model architecture enables to explicitly incorporate temporal dependencies and therefore provides a more methodologically appropriate framework for long-term, multi-hour traffic congestion forecasting.*

Keywords *- Intelligent Transportation System, Traffic congestion, Long-term traffic congestion, Xception, Convolutional neural network, Deep learning.*

1. Introduction

Expeditious urbanization and a continuous rise in the quantity of automobiles have drawn a lot of attention to traffic congestion. A range of strategies is utilized to deal with the issue of traffic. At present, there is a growing utilization of Advanced Traffic Management Systems (ATMS), Dynamic Route Guidance Systems (DRGS), Advanced Traffic Information Systems (ATIS), and a range of add-on Intelligent Transportation Systems (ITS). Precise and dependable traffic forecasting, being the fundamental necessity and essential technology of these systems, has garnered increasing attention. Studying traffic flow predictions holds considerable theoretical importance and practical worth [1]. Traffic forecasting is the process of anticipating future traffic circumstances over the examination about the movement of transit data. Forecasts are categorised as short-term or long-term according to their time [2]. The original reflects a brief forecast interval, generally lasting less than 30 minutes, while the latter represents an extended forecast timeframe, such as a day, week, or beyond [3]. The prediction of traffic congestion is crucial for smart transportation systems. For instance, GPS navigation systems that include a feature for forecasting traffic congestion may offer more efficient routing alternatives. In addition, the capacity to forecast the occurrence of traffic congestion empowers the department in charge of traffic management to significantly improve its management.

In the context of predicting traffic congestion, three critical issues arise: the estimation of journey time, the prediction of the bottleneck itself, and the forecasting of the amount of traffic. This study focuses on forecasting traffic congestion, conceptualized as a unilateral decision issue concerning how much the quantity of traffic would imminently exceed a monitoring criterion [4].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

In recent times, expansion counts seemed to be piling up worldwide. As per UN forecasts, through 2030, the comprehensive urban community is normal to hit approximately by 2050, the global population is projected to reach 4.9 billion, with nearly 70% inhabiting urban areas [5]. Intelligent Transportation Systems (ITS) improve how transit systems work and make traffic flow better, playing a key role in Internet of Things (IoT) for smart mobility. A basic goal concerning ITS aims for accomplish traffic flow and ensure safety while reducing travel time and fuel usage. Owing to swift economic growth and urban development, the quantity of motor vehicles is rising quickly. This poses significant difficulties and pressures for city road traffic. The shortcomings of traditional road planning and design have started to become apparent, leading to progressively worse traffic congestion problems. It also adds to environmental pollution [6]. As a vital component of ITS, traffic congestion forecasting seeks to predict upcoming traffic scenarios (i.e., rapid, sluggish, and congested) on roadways, which is essential for human well-being and urban management. For instance, precise forecasting of traffic congestion allows drivers to make knowledgeable trip planning choices ahead of time, significantly decreasing travel duration and fuel usage. Route planning, public transit timetabling, and emergency response strategies. Consequently, congestion prediction has been widely researched in both scholarly and commercial fields [7]. The phenomenon of traffic congestion is considered to be a particularly severe occurrence that is not strictly related to space and time, and a modelling technique that is centred on occurrences has been developed. The event-driven modelling technique can better capture how congestion starts and fades away over time and space, while many earlier studies have overlooked how individual congestion events develop. Techniques based on Deep Learning (DL) and driven by data have been widely used with short-term forecasting. This appears to be achievable the aforementioned widely recognized methods to predict traffic congestion. Congestion is not inherently happened at identical time as well as place, even though many models driven by data can detect spatiotemporal traffic patterns after being trained on enough data. As a result, these vanilla frameworks have limitations in how they can be used for the whole road system, which shows different patterns of congestion over time and in different locations. Methods based on machine learning generally necessitate a substantial quantity of samples to tackle this heterogeneity problem, since local road conditions (such as design of the system, architecture of the arteries, current moment and the environment, and the features of the movement of traffic) must be considered. Conversely, intricate machine learning models pose a major computational difficulty during both training and inference, since short-term congestion forecasting is a time-sensitive task [8]. The time frame for Long-Term Traffic Congestion Prediction (LTTCP) extends against diverse hours toward days or continuous weeks into a future. The goal regarding LTTCP entails aid in architecture design, policy development, occasion organization, as well as strategies for long-term traffic control. Features about LTTCP involve low temporal resolution that emphasizes overarching trends instead of short-term variations. Additionally, it frequently necessitates taking into account variables such as seasonal patterns, economic influences, and scheduled occasions.

Typically, LTTCP implementations feature time series modelling, statistical approaches, and machine learning frameworks that can handle relationships that last a long time [9]. Congestion forecasting is a key focus for traffic management centers globally to guarantee prompt incident response. Up to now, predicting traffic congestion has garnered significant interest within both civil engineering and information technology fields. Initial studies concentrate on predicting individual sites using traffic with one dimension, like the Autoregressive Integrated Moving Average (ARIMA) technique as well as nearest neighbour approach. Lately, the focus has shifted to forecasting using spatial temporal correlations in traffic flows. A significant challenge of sophisticated traffic management is accurately forecasting traffic congestion. The total land area in urban regions and the quantity of vehicles on the roads have risen to a level where they are now at full capacity. As a result, road transportation became very challenging, causing ecological disruption and posing a danger to human life on the roads. Some issues were addressed through the use of recent innovations. Anticipating traffic circumstances ahead of time helps with trip planning, estimating journey duration, and avoiding congestion. In many instances, outside influences account for the erratic and chaotic nature of traffic behaviour. Several issues were managed once different types of technology were implemented. The need for traffic forecasting has resulted in the creation of various procedures can be categorized via two fundamental types: parametric and nonparametric.

Generally, Kalman techniques for time-series analysis laid the foundation for parametric methods. The ARIMA approach is widely employed for road traffic prediction due to its simplicity and efficacy in estimating continuous and stable period data. An essential element of ARIMA technique is adjusting approach characteristics, like the way (p, d, q) standards for autoregressive, differencing, and moving average components. ARIMA models are capable of identifying linear trends but might struggle with non-linear trends. Models employing ARIMA can manage seasonal data as well, either by utilizing seasonal variation or by explicitly integrating seasonal modules. The ARIMA model is harder to interpret because it emphasizes details regarding the time series' mathematical characteristics. Analyzing the ARIMA strategy features and diagnostics requires a solid understanding of basic statistics: the KARIMA technique, a combined strategy for executing immediate travel predictions. ARIMA constitutes a kind of parametric approach. The approach utilizes Kohonen's self-organizing map as basic framework; in addition, the ARIMA model is optimized for

each category individually [10]. Seasonal Auto Regressive Integrated Moving Average (SARIMA) technique utilized dataset by immediate forecasting of journey times. The SARMIA approach considers not solely the prior uninterrupted delays but also figure from equivalent times on previous occasions for the purpose of predicting journey duration. This is accomplished by implementing a seasonality component that improves the accuracy of trip time predictions in linear approaches [11].

1.1. Background

Traffic congestion prediction has become a critical component of Intelligent Transportation Systems (ITS) for enhancing transportation efficiency, road safety, and traffic management. Traditional machine learning approaches, including Support Vector Machines (SVM), linear regression, and decision trees, have been widely employed to discern and predict traffic congestion employing automotive speed, traffic volume, weather parameters, and accident records. Advances in deep learning have integrated GRU, BiLSTM, CNN, YOLO and attention-based techniques to capture spatial and temporal traffic characteristics. In contrast, the precise capture of intricate spatial-temporal dependencies and the prediction of congestion situations across extensive future time horizons continue to be difficult. Consequently, proactive traffic management necessitates the development of a hybrid deep learning framework that is integrated and capable of predicting long-term traffic congestion.

2. Literature Survey

In past study, an SVM-based method for Predicting Real Highway Traffic Congestion (SRHTCP) was developed that aggregates roadway data of Taiwan's Area National Freeway Bureau, including reports traffic accidents caused by drivers and weather conditions. The congested condition of a section of overhead expressway is assessed in real time by the fuzzy theory, which considers road velocity, volume of traffic, and weather. While ML is frequently linked to more sophisticated methods, conventional methods for instance linear regression as well as decision trees, are widely utilized for predicting congestion. Numerous researchers have explored DL as it appears to be a powerful method for identifying patterns in data [12]. Researchers previously developed a collaborative traffic congestion detection method, a new cooperative approach utilizing Vehicle-to-Vehicle (V2V) infrastructure aimed at identifying traffic congestion, achieving congestion detection probabilities of 90% [13]. In past study introduced You Only Look Once (YOLO) as well as Deep Convolutional Neural Network (DCNN) approaches for recognizing traffic congestion illustration. The benefits of DL approach were evaluated using Support Vector Machine (SVM). In the dataset, visuals of congested roads comprised annotations, and simulations were created employing square footage data from adjacent sensor equipment [14] and in past study introduced convolutional bidirectional long short-term memory method that combines CNN with BiLSTM as well as spatial and temporal features. An initial travel rapidity data has been reduced based on its spatio-temporal characteristics, resulting in the formation of a matrix with 3 dimensions prediction method. Subsequently, CNN incorporates geometric attributes, while the Bi-LSTM extracts temporal elements; ultimately, the predictive outcomes are produced as a throughput [15]. Researchers invented in the past introduced an effective forecasting framework for traffic congestion by leveraging the advantages of Gated Recurrent Unit (GRU) approach. Temporal and spatial features were developed to reflect daily as well as past traffic patterns, while categorical data incorporated using attribute integration. The concentration-driven GRU strategy employs a method of concentration to focus on relevant past information, thereby improving its forecasting abilities by specifically emphasizing critical times. By effectively handling temporal relationships and modifications, this design model, which features multiple concealed levels and attention mechanisms, enables improved traffic projections [16]. In past study introduced an innovative method, Pulse-Coupled Neural Network (PCNN), created from DCNN using 2 key strategies: historical data movable as well as multi-grained knowledge. Historical data remains firstly modified toward reflect appropriate transit circumstances as well as historical traffic patterns. Subsequently, the order of inputs is subjected to an array of convergence, resulting in the construction of a 2-dimensional matrix [17]. In past study suggested utilizing a deep learning approach to thoroughly examine modes of transit. They depended on automobile GPS information, Recurrent Neural Network (RNN), as well as deep Restricted Boltzmann Machine (RBM). The RNN-RBM model was subsequently used towards investigate the growth of traffic congestion. A numerical analysis remained performed in Ningbo, China, toward verify the efficacy of the proposed method [18]. In past study created a method to assess traffic congestion that uses fuzzy logic and combines multiple measurements, focusing on three anticipated vehicle volume factors for 5G IoT. The method for forecasting has the ability to change the weights for the simulation model and the smoothing coefficient. The adaptive critic method sets the weights for analysing traffic congestion data, while the trapezoidal membership function looks at how much traffic congestion is grouped in the ranking [19].

The primary research gap defined in the existing literature is the absence of a viable integrated framework that is capable of predicting congestion conditions over an extended forecasting horizon and learning intricate spatial and temporal traffic patterns concerning simulated traffic records. Current solutions usually consider geographical feature extraction and temporal dependency modeling individually, but cannot capture the dynamic congestion patterns and retain consistent prediction capability over numerous future time steps. However, the successful use of traffic congestion photographs as a replacement resource of data for long-term congestion prediction has been comparatively less explored. This

introduces the demand for a strong hybrid deep learning foundation for extracting the discriminative visual features, capturing sequential relationships, and enhancing classification for subsequent traffic congestion instances.

This research article presents a supervised hybrid method-based strategy for highway congestion prediction. Hybrid approaches improve the performance of different filters in DL. Consequently, hybrid approaches have shown enhanced effectiveness regarding gridlock in roadway scenarios.

2.1. Contributions

The main suggestions put out in the article are summarized below:

- We suggested a hybrid Xception–Convolutional Long Short-Term Memory K-Nearest Neighbor (XCL-KNN) classifier model to forecast LTTC for a duration of 17 hours.
- We present Xception, a tool that predicts traffic congestion by using a dataset and combining different methods like separable convolution, ReLU, and convolution.
- The extracted feature sequences are subsequently supplied to a ConvLSTM network, which captures both spatial and temporal dependencies among successive traffic observations.
- KNN determines the ultimate traffic flow by averaging the K Nearest Neighbor (with k=5) utilizing the Euclidean Distance method.

The succeeding remains the organization about the study's subsequent sections: Section 3 contains the Methodology. Results Discussion covers in section 4. Conclusion displayed in section 5.

3. Methodology

3.1. XCL-KNN Approach

This part presents the formulation of an XCL-KNN methodology for assessing LTTC. The XCL-KNN methodology aims to predict LTTC every three hours over a span of seventeen hours, potentially contributing to the reduction of travel expenses, duration, air pollution, and global warming. When employing Xception as a feature extractor, you transform raw images into concise, distinctive feature vectors. Utilizing KNN on those vectors generally yields favorable results. The subsequent subsections offer further details regarding the suggested XCL-KNN architecture. Figure 1 illustrates the suggested XCL-KNN framework.

3.2. Xception

Xception Model was first presented by Francois Chollet. Xception provides enhanced version about Inception design that uses depthwise separable convolutions in place of conventional Inception modules. Convolutional Neural Networks (CNN) use the Xception approach, which provides a enhanced revision of Inception approach. This model, broader convolution layers as well as deep convolution layers work together.

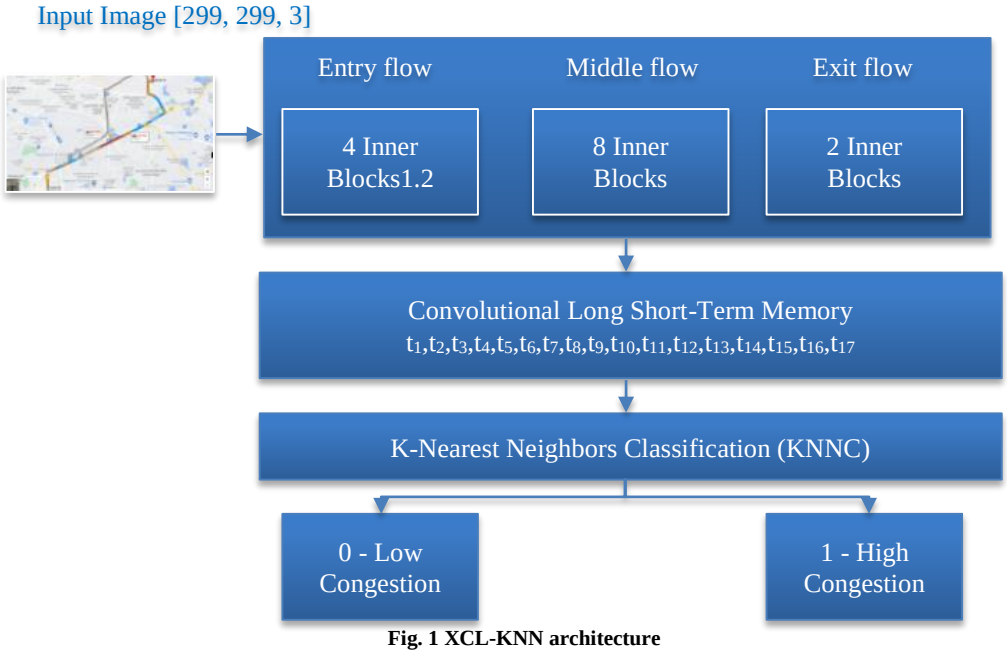


Fig. 1 XCL-KNN architecture

It provides a deep CNN with 71 layers and is based on an enhanced version of Google's Inception model. This architecture consists of layers that incorporate depthwise separable convolution levels. A prelearnt distinctive approach was developed using millions about photos against ImageNet repository. The approach can recognize hundreds of different

classifiers and offers thorough examples of its effectiveness across a large variety of photos. Xception is a powerful Deep Convolutional Neural Network that was recently trained on ImageNet, accurately capturing high-level, intangible characteristics from photos. This technique is exceptionally proficient in identifying images and segmentation. The input in this configuration is 299x299 pixels in size, with three color channels (RGB), and it goes through the entry, middle, and exit flows. With extra batch normalization, every convolution and separable convolution layer is considered. The Xception structure includes 36 convolutional layers that act as the network's backbone for feature extraction. The model adds a dense layer to the convolutional source to emphasize picture division. Along with the starting as well as finishing works, all 14 convolutional layers have linear residual connections. Xception system uses special convolution blocks that separate depth and max pooling, along with residual neural networks. It comprises a sequential configuration of residually connected depthwise separable convolutional layers. The effectiveness of the Xception system relies on 2 key concepts: 1. depthwise separable convolution, as well as 2. Convolution blocks like those in ResNet. Depthwise separable convolutions function as substitutes for traditional convolutions and are recognized for their considerable efficiency in processing time [20]. Two external dense layers are integrated into the Xception framework for multi-class categorization. The Xception classifier is described in Figure 2 below.

3.3. Temporal Preprocessing

The traffic photos shown below in chronological order

$K_1, K_2, K_3 \ldots . K_t$

Where

K_t – represents Google Maps traffic image at time t.

Xception extracts spatial features described as below

$$Y_t = f_{xception}(K_t)$$

A chronological series of length L is subsequently formed as

$$Y_t = [Y_{t-L+1}, Y_{t-L+2}, \ldots . Y_t]$$

The resultant sequence of length L is subsequently provided to ConvLSTM.

$$C_t = ConvLSTM(Y_t)$$

The resultant temporal depiction is provided to KNN:

$$\hat{P}_{t+h} = KNN(H_t)$$

Where

Y_t - spatial feature map extracted from the image Kt

L - sequence length

Ht –prediction horizon (H=1,2,3,…17)

$\hat{P}_{t+h}$ - predicted traffic congestion class.

3.4. Convolutional Long Short-Term Memory

The (ConvLSTM) network constitutes the spatio-temporal learning component of the proposed Xception–ConvLSTM-K-Nearest Neighbor (XCL-KNN) model. The principal purpose of ConvLSTM is to learn the temporal evolution of spatial traffic features extracted by the Xception network. Unlike conventional LSTM networks, which generally operate on one-dimensional feature vectors, ConvLSTM incorporates convolutional process into the input-to-state and state-to-state transitions. Consequently, it can preserve the spatial structure of traffic feature maps while simultaneously learning their temporal dependencies. In the proposed architecture, a sequence of historical traffic images is initially processed by Xception to extract spatial feature maps representing vehicle distribution, road occupancy, traffic density, and other congestion-related visual characteristics. These feature maps are arranged as a temporal sequence and provided to the ConvLSTM layer. At each time step (t), ConvLSTM receives current Xception feature map Xt, together with the previous hidden state H{t-1} and cell state C{t-1}. The input gate Equation 1, forget gate Equation 2, and output gate Equation 3 are regulating the information flow and enable the network to retain important historical traffic information while eliminating irrelevant information. The ConvLSTM operations are represented by:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f \quad (2)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \quad (3)$$

Where

i_t – input gate,

f_t – forget gate,

o_t – output gate,

X_t – input at time step (t),

H_{t-1} - denotes the hidden state,

C_t - represents the cell state,

* - denotes convolution,

σ is the sigmoid activation function,

W_{xi} - Weight matrix connecting the input X_t to the input gate,

W_{hi} - Weight matrix connecting the previous hidden state, H_{t-1} to the input gate.

b_i - Bias vector,

W_{xf} - Weight matrix mapping the input X_t to the forget gate,

W_{hf} - Weight matrix mapping the previous hidden state H_{t-1} to the forget gate,

W_{xo} - Weight matrix connecting the input X_t to the output gate,

W_{ho} - Weight matrix connecting the previous hidden state H_{t-1} to the output gate.

3.5. K-Nearest Neighbor

The supervised machine learning algorithm known as KNN manages both classification and regression tasks. KNN

operates efficiently in low-dimensional, pertinent feature spaces, where analogous data points are nearby. Distance measurements in the feature space help KNN component classify data by using the features from the training data to infer outcomes for test dataset. The acquired ConvLSTM representation is converted into a feature vector via the necessary flattening or pooling procedure. This feature vector is provided to the KNN classifier. Throughout the training process, KNN retains the feature representations along with their associated congestion-category labels. In the prediction phase, the Euclidean distance is computed among the input feature vector and the archived training feature vectors. For instance, the KNN algorithm computes the distinction between training and testing information to identify features of a newly added data point. After that, KNN will use the separation of the new data point from the existing training data points with similar properties to classify the new data point. The KNN is illustrated in Figure 3 below. A technique that is most frequently employed for determining this distance is Euclidean [21]. The length that connects two points is known as Euclidean distance, employed to calculate the distance separating them. The Euclidean Distance method computes the square root of sum of the squared discrepancies among an original data point (c) and a pre-existing data point (d), as demonstrated in Equation 4.

$$\sqrt{\sum_{a=1}^{b}(c_i - d_i)^2} \tag{4}$$

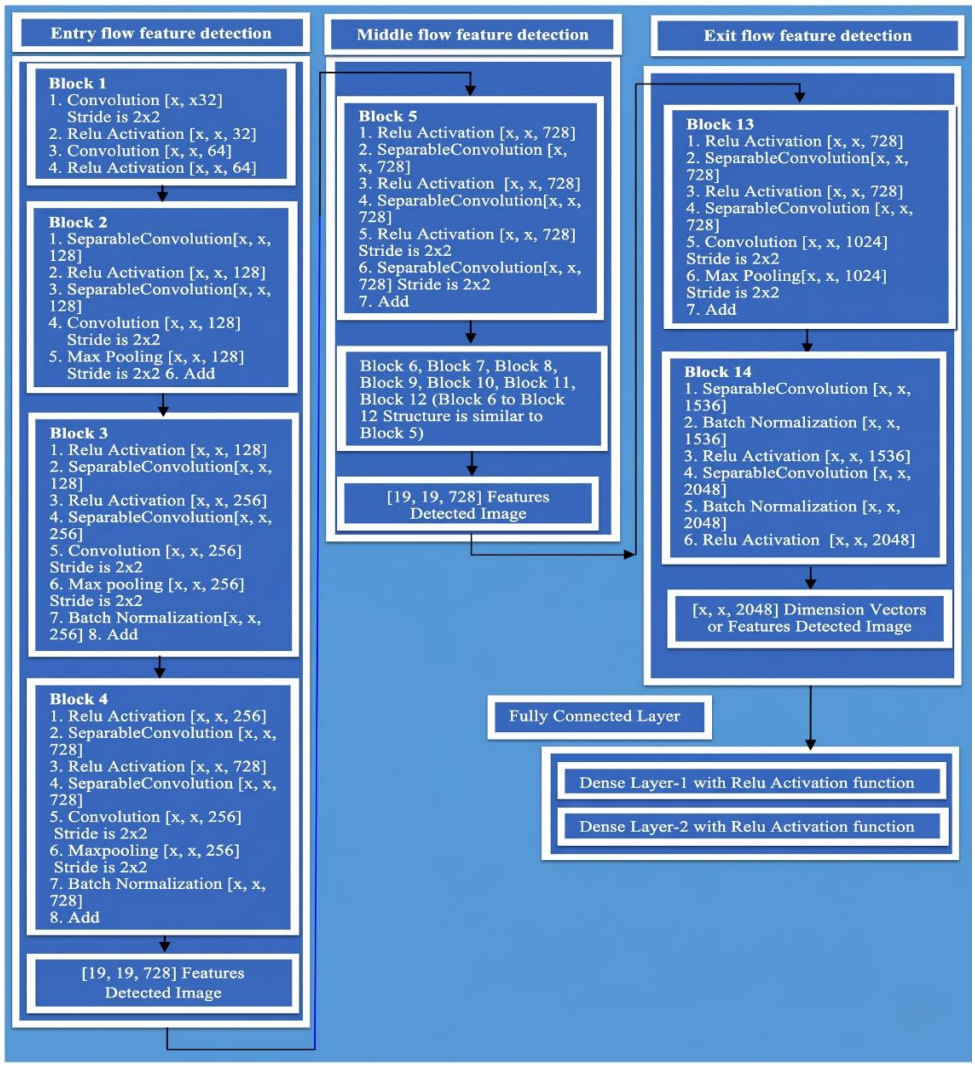


Fig. 2 Xception classifier

3.5.1. Distance Metric

The distance metric plays an essential role in KNN, as it identifies how near a point is to its surrounding points. The selection of metrics can significantly influence performance, particularly in high-dimensional or structured datasets.

The Euclidean distance is the metric most frequently utilized in KNN, particularly when your data is continuous and normalized. It quantifies the direct distance between two points in Euclidean space.

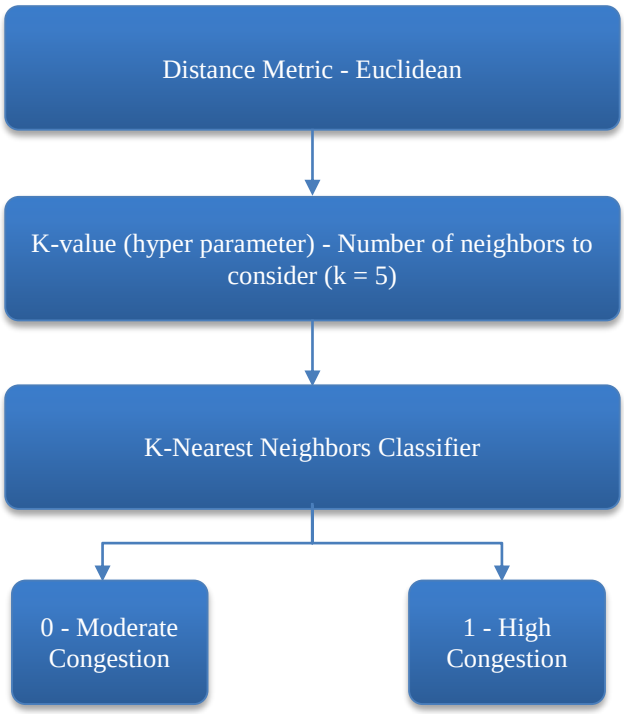


Fig. 3 K-Nearest Neighbor

3.5.2. K-Value

The k in K-Nearest Neighbor is an essential hyperparameter that specifies the number of Neighbor the algorithm considers to generate a prediction. For ordering, the prediction relies on the predominant class among the k nearest Neighbor. If the aspect involves binary classification, it is advisable to use odd numbers for k.

3.6. Implementation

3.6.1. XCL-KNN

1. Generate an ImageDataGenerator to import picture batch size of 32, class mode set to "categorical," and each image rendered as a 2D array of dimensions were all taken from the Train directory [299, 299].
2. Likewise, set up an ImageDataGenerator to import picture data with a batch size of 32, class mode set to "categorical," and each image formatted as a 2D array from the Test directory [299, 299].
3. Generate an instance of the pre-trained Xception Model using Keras.
4. Deactivate the fourteen blocks (layers) of Xception that were previously learned.
5. Remove the traditional output sections from the Xception model's normal outcome layers.
6. Incorporate all the interconnected layers of the outcome.
   a. Combine the Flatten layer
   b. Include 2 Dense layers followed by neurons set to 50, 100 and 200 using "relu" (activation function)
7. Incorporate a single-neuron output layer with K set to 5, where K represents the Nearest Neighbor.
8. Include Euclidean distance.
9. Train the model using the input information for 25 epochs and measure the loss.
10. Forecast traffic congestion.

3.7. Hyperparameter Tuning

Table 1 provides an explanation of the XCL-KNN features.

Table 1. XCL-KNN parameters

Hyperparameter	Range
Count of input parameters	3D array (299, 299, 3)
Loss Function	binary_crossentropy
Count of Neurons	50, 100, 200
Fully Connected Layer	3
Optimizer	Adam
Activation Function	Relu
Count of Separable Convolution Layers	34
Count of Maxpooling Layers	4
Count of Convolution Layers	6
Count of Flatten Layers	1
Count of Epochs	25
Neurons in output layers	1
K Nearest Neighbor	5
Distance	Euclidean

4. Results and Discussion

4.1. Data Acquisition

This study involved collecting transit congestion information by utilizing the Selenium tool to get snapshots from the website provider Google Maps. Approximately 2500 traffic photos were collected throughout peak morning and evening intervals. Images of traffic in Bengaluru, India, were taken between the ITI Bus Depot and the Halsuru Traffic Police Station. The author applied XCL-KNN model to evaluate its efficiency with actual traffic congestion datasets collected from 11-05-2023 to 13-06-2025.

4.2. Evaluation of Model Effectiveness

Figure 4 illustrates confusion matrix formula for assessing the performance of classifiers. It presents an N x N matrix that is utilized to assess the classification approach effectiveness, where N remains number of target classes. The matrix contrasts the real objective rates with projected methods for learning. The confusion matrix comprises two categories: Type 1 error and Type 2 error. If a method shows a lower rate of Type 1 and Type 2 mistakes, the data deems it optimally fitted. Table 2 provides the classification report for the models under comparison, which encompasses F1-score, precision and recall. It is important to note that the classifier's precision in identifying positive classes is evaluated by the ratio of true positives to false positives, which is estimated by tp / (tp + fp). Recall assesses classifier's competence to identify all positive patterns, expressed as tp / (tp + fn), wherein fn signifies false negatives.

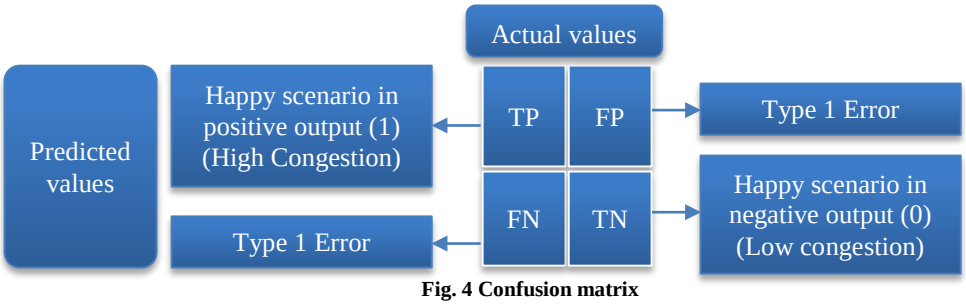


Fig. 4 Confusion matrix

F1 score, which implies the inverse mean of recall as well as precision, is adopted by applying the principle F1 score = 2 × (recall × precision) / (recall + precision) [22]. Note that TP signifies true positives, and FP signifies false positives.

Table 2. Category report

Methods	F1 score (%)	Precision (%)	Recall (%)
XCL-KNN	98.75	98.9	98.6
XBi-LSTM	97.35	97.5	97.2
CNN-LSTM	96.50	96.8	96.2
ResNet-LSTM	95.55	95.7	95.4
CNN+MLP	93.6	93.7	93.5

Table 3. Error metric rates

Methods	Training		Testing	
	Type 1 Error	Type 2 Error	Type 1 Error	Type 2 Error
CNN+MLP	7.4	5.4	6.4	4.5
ResNet-LSTM	6.8	4.8	5.7	3.1
CNN+LSTM	5.7	4.5	3.9	3.4
XBi-LSTM	3.8	3.1	2.4	2.1
XCL-KNN	1.6	1.4	0.9	0.6

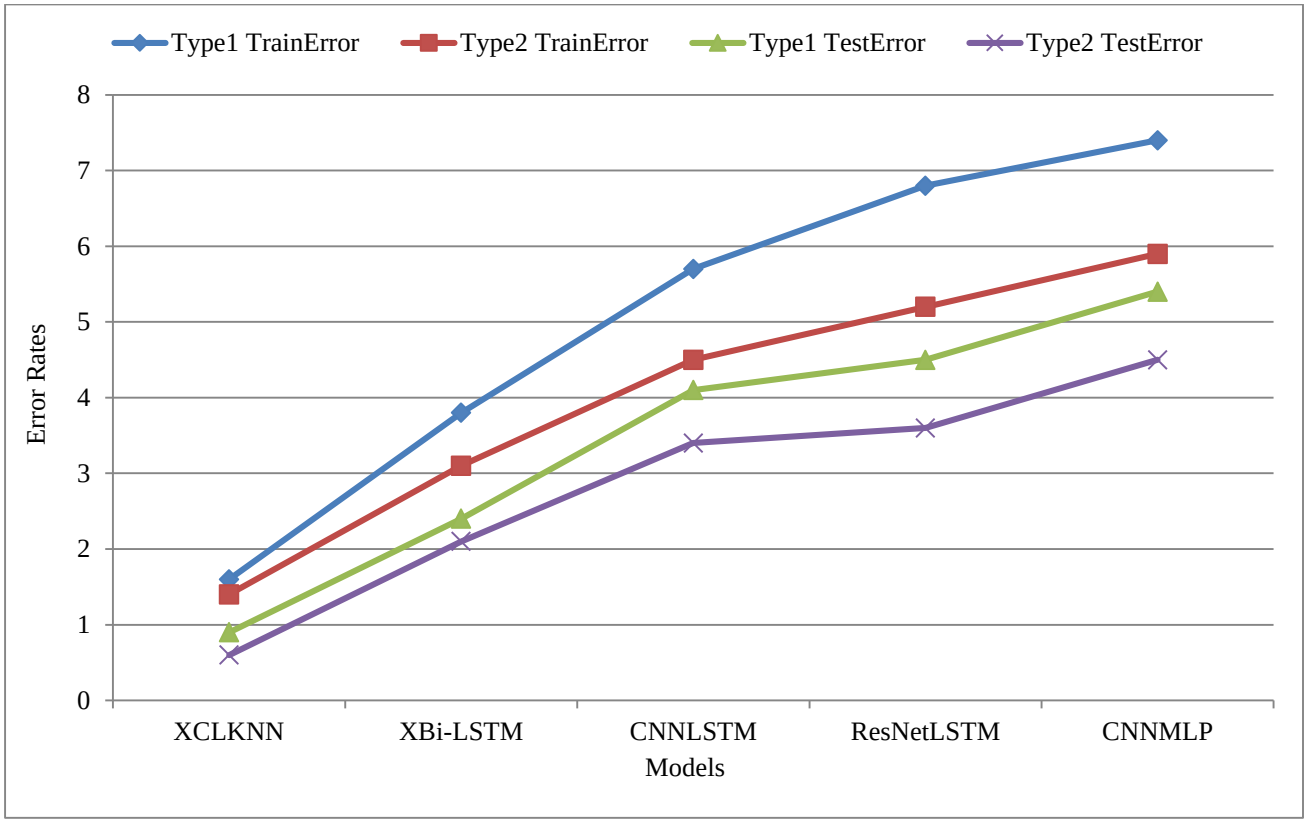
4.3. Discussion

As exposed in Table 2, the proposed XCL-KNN approach achieved the best results in the overall performance, with an F1-score of 98.75%, precision of 98.9%, and recall of 98.6%. This meant that the classification of traffic congestion was highly accurate and balanced. XBi-LSTM with F1-score of 97.35% and precision and recall of 97.5% and 97.2%, respectively, which proves the effectiveness of the combination of deep Xception-based feature extraction with bidirectional temporal learning. CNN-LSTM obtained the third-best performance with an F1-score of 96.50%, a precision of 96.8%, and a recall of 96.2%, confirming the capability of CNN and LSTM to detect spatial and temporal congestion patterns. ResNet-LSTM obtained an F1-score of 95.55%, precision of 95.7% and recall of 95.4%, indicating relatively lower performance but learning spatial and temporal features. CNN+MLP achieved the lowest performance, with F1-score of 93.6%, precision of 93.7%, and recall of 93.5%. This is probably due to the lack of specific sequential learning, which reduces the capability of capturing the temporal congestion patterns. Typically, binary classification jobs use Type 1 as well as Type 2 errors to measure the failures. A false positive is a Type 1 error that happens when forecast is inaccurate under positive or beneficial preferences. When predictions were incorrect due to unfavourable or negative circumstances, this type of Type 2 error is called a false negative. To measure or assess the efficacy of a model, one may examine the rates of Type 1 along with Type 2 faults. This is extremely critical to diminish Type I as well as Type 2 mistake rates, which are obtained via the confusion matrix. In this study, for all peer methods, generated a confusion matrix for the suggested approach and computed Type 1 and Type 2 error stages, as indicated in Error Table 3. The suggested system demonstrates minor error counts during each training and testing, as evidenced by the fault metric rates in Figure 5(a). Overall, while testing several algorithms for accuracy, loss values, and error rates, the recommended model turned out to be the most effective one. They selected the Bengaluru, India traffic congestion image dataset, which consists of 1GB and 2500 images, as the basis for the proposed method of dividing the data into testing and training datasets in a 70% to 30% ratio. The training dataset contains 1750 photos, while the testing dataset consists of 750 photos. First, train the hybrid approach on training dataset and then assess it on training as well as testing datasets to determine accuracy as well as loss. In the training dataset, the accuracy increased

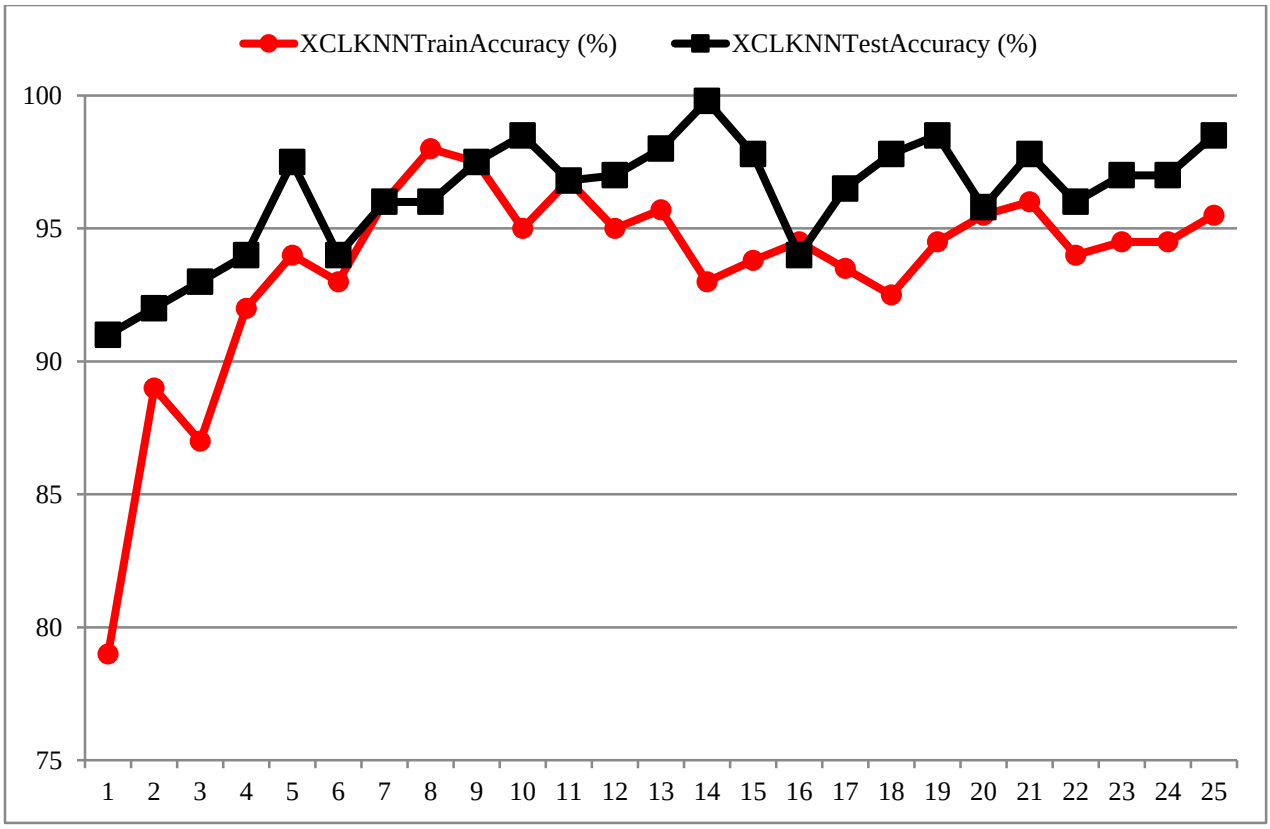
from 79% to 95.5%, as illustrated in Figure 5(b). In the testing dataset, the accuracy reached 98.7%, starting at 91%. The model combined 5 points when the training and testing accuracy were the same before epoch 25, indicating it understood the input data well; as a result, the accuracies became stable from epoch 7 and are likely to change before epoch 26. In this study, found that using 25 epochs yields the best results. Consequently, the overall accuracy of the hybrid approach is 98.7%. The ensemble procedure required fewer epochs, taking a total of 14 minutes and 27 seconds to analyze the training dataset (25 epochs at 30 seconds each), while each epoch lasted 28 seconds. The efficiency is acceptable for the chosen 1750 photos illustrating traffic congestion, and loss ratios within the training and testing datasets in a similar manner. Initial loss in the training dataset was 8 (800%), which progressively decreased with additional epochs, reaching 0.11. In the testing dataset, the loss rate commenced at 0.6 and diminished to 0.1 at epoch 25. The proposed approach produced a loss value of 10% within testing phase, exhibiting one convergence point between training and testing. Figure 5(c) shows that, for 98.7 percent of the time, the projected and actual levels of traffic congestion are similar. The predicted value of the suggested approach is close to the actual traffic congestion, with the exception of one specific time point (between 11:00 and 13:00) when it differs. In Figure 5(d) below, it is clear that forecasted traffic congestion aligns perfectly with real congestion for 98.7% of the time. In this research, examined peer algorithms namely to review the ability of offered model. Each of these algorithms employs data on traffic congestion in Bengaluru, utilizing a testing dataset consisting of 1000 photos and fed into a training dataset consisting of 1500 images. They have clearly discovered that the hybrid technique suggested outperforms other comparable algorithms. When compared to other designs, the proposed model shows higher accuracy, as illustrated below Figure 5(e) below. In the Figure 5(f) below, the same observation is noted. There were multiple situations where the expected value was not met and did not align with the actual value; this mainly occurred repeatedly in peer models, unlike the suggested method. As a result, found that the peer techniques are not superior to the proposed paradigm. After observing each method in traffic congestion forecasting, we measured its accuracy; Table 4 displays the accuracy scores. Because of its highly tuned internal structure, which uses the Xception model for feature recognition, LSTM for and the KNN approach for output prediction, the proposed hybrid model performed exceptionally well and took the top spot with a score of 98.7%.

Table 4. Accuracy values

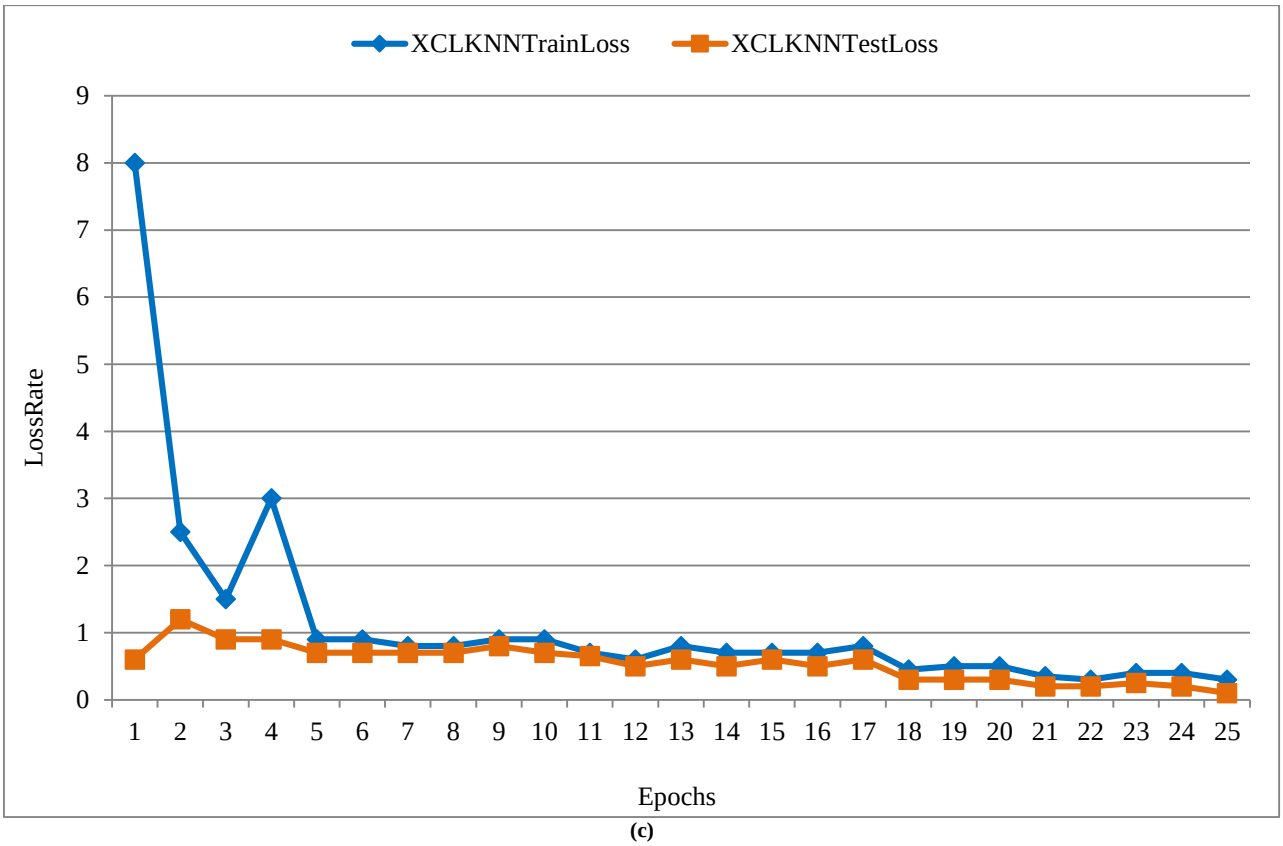
Algorithm	Accuracy (%)
XCL-KNN	98.7
XBi-LSTM	97.4
CNN-LSTM	95.6
ResNet-LSTM	94.8
CNNMLP	93.5



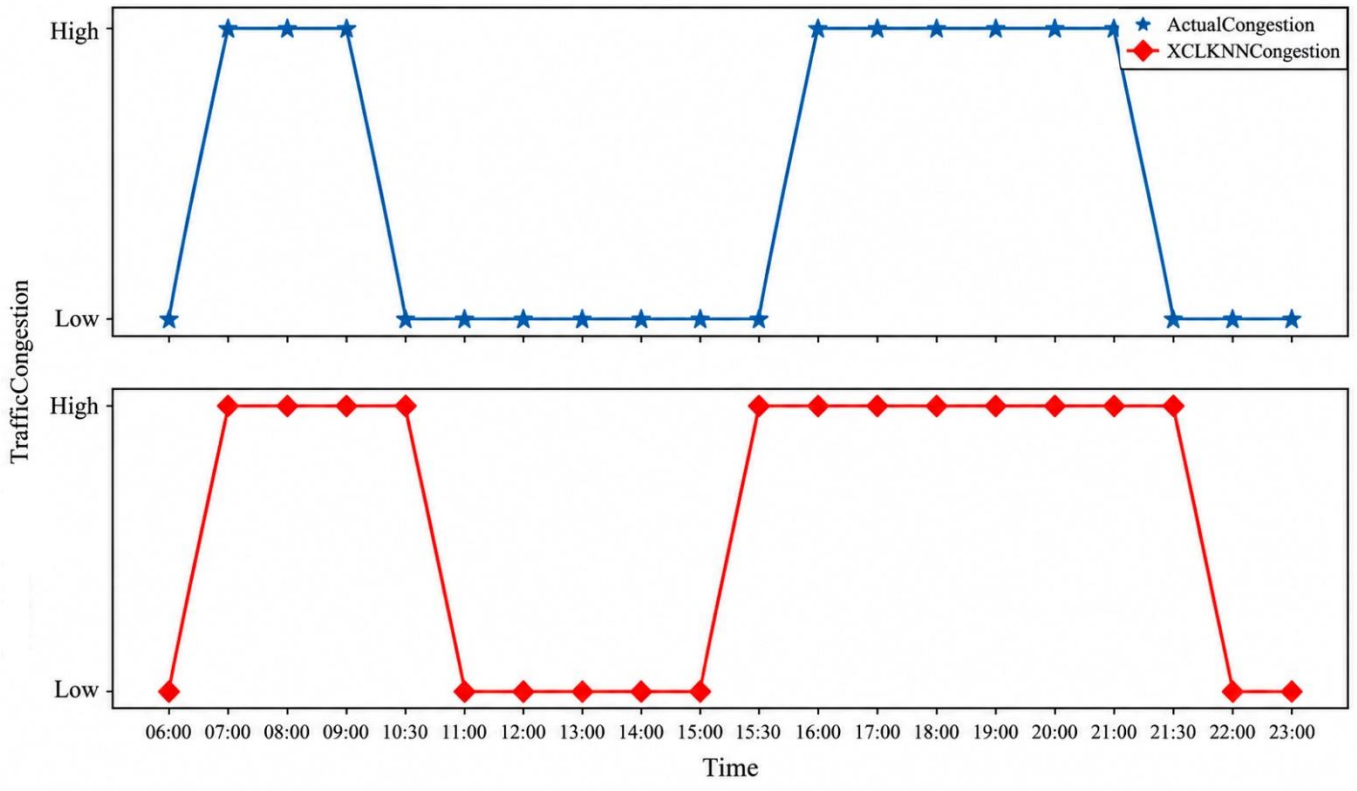
(a)


XCLKNNTrainAccuracy (%)
XCLKNNTestAccuracy (%)
100
95
90
85
80
75
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

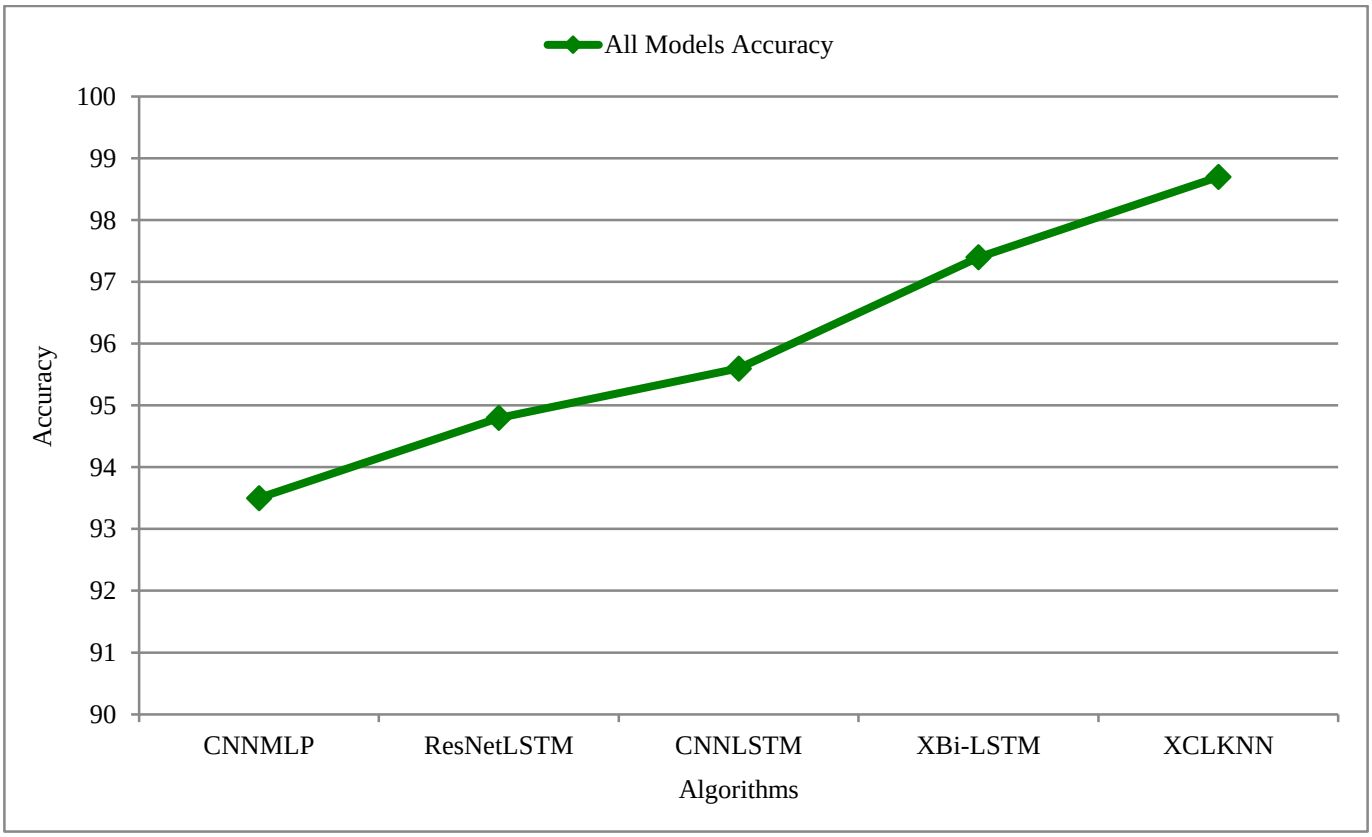
(b)


XCLKNNTrainLoss
XCLKNNTestLoss
LossRate
9
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Epochs

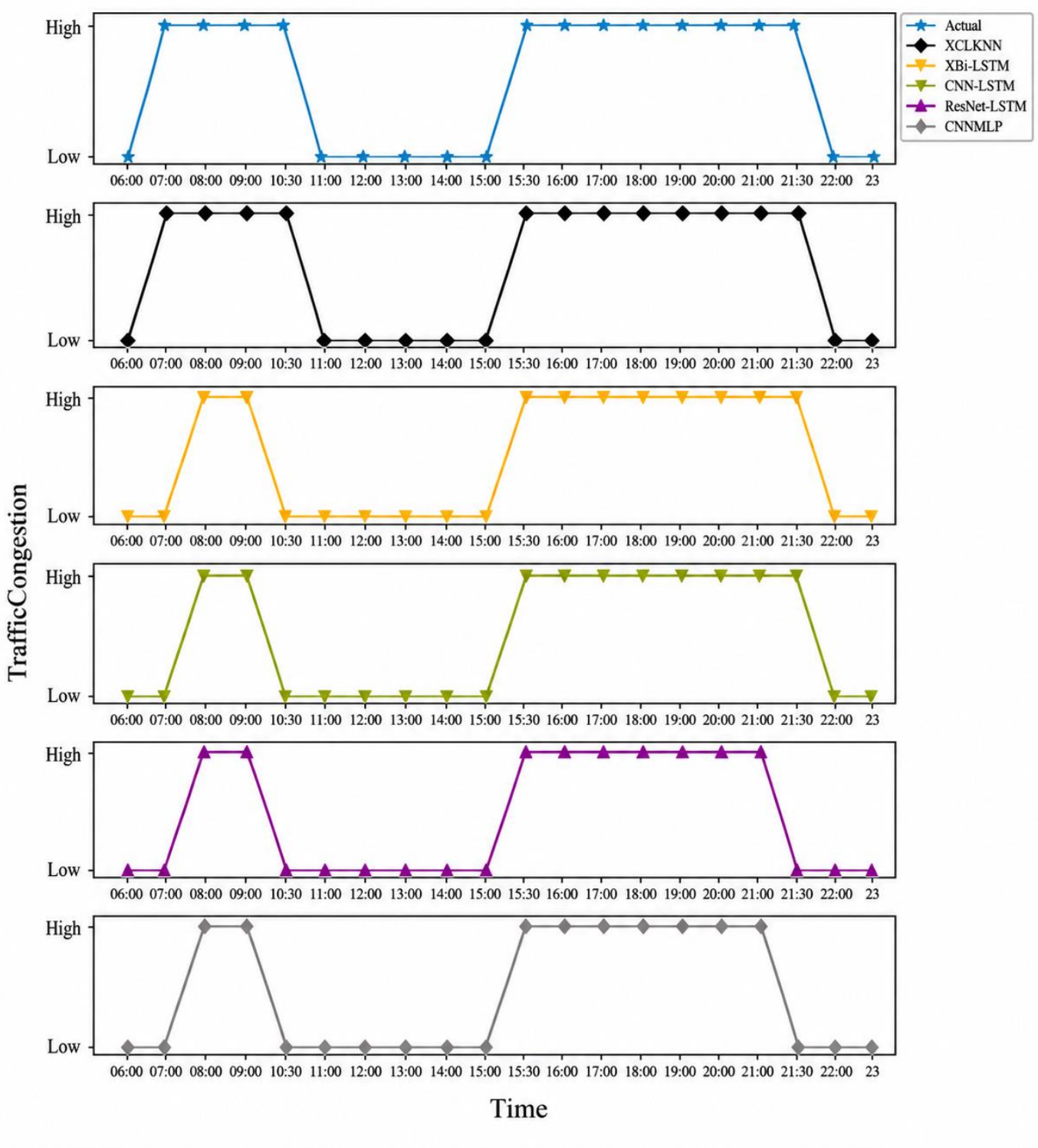
(c)


ActualCongestion
XCLKNNCongestion
TrafficCongestion
High
Low
High
Low
06:00 07:00 08:00 09:00 10:30 11:00 12:00 13:00 14:00 15:00 15:30 16:00 17:00 18:00 19:00 20:00 21:00 21:30 22:00 23:00
Time

(d)


All Models Accuracy
100
99
98
97
96
95
94
93
92
91
90
Accuracy
CNNMLP
ResNetLSTM
CNNLSTM
XBi-LSTM
XCLKNN
Algorithms

(e)



(f)

Fig 5. (a) All models Type1, Type2 error graphs, (b) Training Testing accuracy of suggested method, (c) Training Testing loss of suggested method, (d) Proposed model traffic congestion prediction, (e) Accuracy graph for all models, and (f) All models traffic congestion prediction comparison.

5. Conclusion

In this study, presented a hybrid algorithm, XCL-KNN, which can precisely forecast long-term traffic congestion over the subsequent seventeen hours. Primarily, the Xception classifier predicts features in the dataset by using convolution, separable convolution and ReLU techniques. Secondarily, KNN uses features chosen from Xception algorithm to forecast long-term traffic congestion by averaging the 5

closest Neighbor based on the Euclidean distance. Therefore, when compared to similar models, the proposed hybrid model produced impressive results. In future efforts to improve the performance and stability of hybrid approach, replace Google Maps screenshots with raw CCTV camera streams, induction loop detector data, or speed time-series. KNN replace with a more efficient model that processes input data and forecasts result values more quickly, accommodating both linear and non-linear input data. To evaluate the hybrid model using a data set that includes weather conditions such as rainy, snowy, and windy seasons.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] Fei Su et al., "Long-Term Forecasting Oriented to Urban Expressway Traffic Situation," *Advances in Mechanical Engineering*, vol. 8, no. 1. pp. 1-16, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Brian L. Smith, Billy M. Williams, and R. Keith Oswald, "Comparison of Parametric and Nonparametric Models for Traffic Flow Forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303-321, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jianhua Guo, and Billy M. Williams, "Real-Time Short-Term Traffic Speed Level Forecasting and Uncertainty Quantification using Layered Kalman Filters," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2175, no. 1, pp. 28-37, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[4] Su Yang, "On Feature Selection for Traffic Congestion Prediction," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 160-169, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[5] United Nations, World Urbanization Prospects: The 2018 Revision, United Nations, Technical Report, 2019. [Google Scholar]

[6] Juan Lu et al., "Expansion of City Scale, Traffic Modes, Traffic Congestion, and Air Pollution," *Cities*, vol. 108, pp. 1-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Keji Wei, Vikrant Vaze, and Alexandre Jacquillat, "Transit Planning Optimization under Ride-Hailing Competition and Traffic Congestion," *Transportation Science*, vol. 56, no. 3, pp. 725-749, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Wenqing Zheng et al., "Integrating the Traffic Science with Representation Learning for City-Wide Network Congestion Prediction," *Information Fusion*, vol. 99, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yiannis Kamarianakis, and Poulicos Prastacos, "Space-Time Modeling of Traffic Flow," *Computers and Geosciences*, vol. 31, no. 2, pp. 119-133, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mascha Van Der Voort, Mark Dougherty, and Susan Watson, "Combining Kohonen Maps with ARIMA Time Series Models to Forecast Traffic Flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307-318, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[11] R. Bauza, and Javier Gozalvez, "Traffic Congestion Detection in Large-Scale Scenarios using Vehicle-to-Vehicle Communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295-1307, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[12] Fan-Hsun Tseng et al., "Congestion Prediction with Big Data for Real-Time Highway Traffic," *IEEE Access*, vol. 6, pp. 57311-57323, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Amir Mohammad Khoei, Ashish Bhaskar, and Edward Chung, "Travel Time Prediction on Signalised Urban Arterials by Applying SARIMA Modelling on Bluetooth Data," *Australasian Transport Research Forum 2013 Proceedings*, Australasian Transport, Research Forum, Australia, pp. 1-18, 2013. [Google Scholar] [Publisher Link]

[14] Pranamesh Chakraborty et al., "Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 45, pp. 222-231, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Tao Li et al., "Short-Term Traffic Congestion Prediction with Conv-BiLSTM Considering Spatio-Temporal Features," *IET Intelligent Transport Systems*, vol. 14, no. 14, pp. 1978-1986, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] M. Sreelekha, and Midhunchakkaravarthy Janarthanan, "Enhanced Traffic Congestion Prediction using Attention-based Multi-Layer GRU Model with Feature Embedding," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, pp. 969-983, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[17] Meng Chen, Xiaohui Yu, and Yang Liu, "PCNN: Deep Convolutional Networks for Short-Term Traffic Congestion Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3550-3559, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xiaolei Ma et al., "Large-Scale Transportation Network Congestion Evolution Prediction using Deep Learning Theory," *PLOS One*, vol. 10, no. 3, pp. 1-17, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[19] Li Liu et al., "TCSA: A Traffic Congestion Situation Assessment Scheme based on Multi-Index Fuzzy Comprehensive Evaluation in 5G-IoV," *Electronics*, vol. 11, no. 7, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] François Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 1800-1807, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Haiyang Yu et al., "A Special Event-based K-Nearest Neighbor Model for Short-Term Traffic State Prediction," *IEEE Access*, vol. 7, pp. 81717-81729, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mohammadreza Heydarian, Thomas E. Doyle, and Reza Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, pp. 19083-19095, 2022. [CrossRef] [Google Scholar] [Publisher Link]