International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://theaspd.com/index.php

Integrating Deep Learning And Ensemble Methods For Urban Traffic Forecasting

Mukta Bhatele[1, a)], **Poonam Bhartiya**[2, b)], **and Akhilesh A. Waoo** [3, c)]
[1,2,3]*AKS University, Satna, Madhya Pradesh, India*
Department of CSE, AKS University, Satna, Madhya Pradesh, India
[a)]*30.muktabhatele@gmail.com*
[b)]*poonambhartiya1989@gmail.com*
[c)] *Corresponding author: akhileshwaoo@gmail.com*

Abstract
Accurate urban traffic forecasting is essential for intelligent transportation systems, enabling efficient traffic control, route optimization, and congestion mitigation. This study presents a hybrid ensemble-based framework that integrates deep learning and machine learning algorithms to enhance the prediction of urban traffic flow. The proposed approach utilizes multiple ensemble techniques, including Bagging with Logistic Regression, Gradient Boosting using XGBoost and LightGBM, and Stacking with Multi-Layer Perceptron (MLP) and Support Vector Classifier (SVC) as base learners. The models were trained and tested on real-life traffic video ***data****, which was processed to extract frame-wise vehicle attributes such as object coordinates, bounding box sizes, and vehicle types. Additional temporal and contextual features were incorporated to improve the robustness of the forecasting model. Experimental evaluations were conducted using performance metrics, including the Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Results indicate that the stacking ensemble method combining MLP and SVC outperformed traditional single-model approaches, demonstrating superior accuracy and stability across dynamic traffic scenarios. The findings confirm that combining ensemble learning with deep neural networks significantly enhances urban traffic forecasting performance and provides scalable solutions for real-world intelligent transportation systems.*
Keywords: *Urban traffic forecasting, Real-life traffic data, Ensemble learning, Deep learning, Bagging, XGBoost, LightGBM, Stacking, MLP, SVC, R^2, RMSE, MAE, Traffic video analysis, Intelligent transportation systems.*

INTRODUCTION

In the last decade, increasing affordability has made it easier for consumers to buy cars, which has resulted in increased traffic on highways. In 2021, there were over 1.4 billion vehicles in the world. When designing Intelligent Transportation Systems (ITS), it is essential that a correct prediction be made of the flow of traffic on highways. This is very valuable for designing app technologies for managing traffic networks, informing the public of route directions, reducing traffic congestion, and improving public transportation, and it is environmentally friendly.Predictive traffic speed forecasts could help to control traffic. Such forecasts could also help individual travellers. For example, traffic management systems can look for areas and possibly anticipate areas of congestion, assess the reasons for the congestion, and suggest steps to eliminate or reduce the congestion, using traffic speed forecasts. Individual travellers may also make use of the predicted estimates provided by multiple navigational apps, such as Google Maps and Waze, to prepare their trips and expected travel times. Many of the current road systems do not utilize traffic monitoring devices. Some cities now have smart and intelligent transport systems by employing devices for traffic condition monitoring, such as extended traffic speed data or moving car data. However, the technologies are needed to enable speed monitoring. Nonetheless, the expense of the equipment necessary for speed surveillance is much higher. Many cities have opted for cost-effective traffic cameras. Flow detection using relatively inexpensive technology.Basic diagram of vehicle traffic flow highlights the complex relationships between traffic volume and speed. When it comes to traffic management on a roadway, the first important duty is to understand traffic flow, which can be done either with floating car data or with camera-based data. The second important duty is to estimate how much traffic there is. Typically, there are two categories of traffic speed prediction:

short-term and long-term. Short-term traffic prediction is an important aspect of Intelligent Transport Systems, as it provides the basis for implementing alternative traffic control methods to make efficient use of existing road networks and fairly distribute vehicles. It is critical to correctly detect and accurately and reliably predict traffic situations for a number of traffic applications and urban transportation systems.The applications of deep learning and machine learning approaches have most often been associated with tasks such as clustering, regression, classification, and prediction. Interestingly, learning has come to be recognized as a central approach to improving the effectiveness of deep learning models. One approach to make the model maximally effective, with related classification performance, is to use ensemble-based learning. Ensemble learning is basically the notion of combining several models, whether they be deep learning predictive models or those that use more traditional algorithms. The predictive outcomes are generally more accurate through leveraging the strengths of multiple models.The learning process using ensembles consists of two major stages. In the first stage, classifiers are selected, including Logistic Regression, DT, Back-Propagation, SVM, Neural Network, or Convolutional Neural Network (CNN), and a classification model is trained to produce a result. The second step incorporates voting, learning, and averaging techniques to combine the models in a hybrid manner to improve their generalization ability. The most common ensemble methods are Bagging, Boosting, and Stacking. Stacking-based methods actually use a meta-classifier to combine the outputs of the classification results of the base classifiers. Heterogeneous machine learning approaches such as Random Forests (RF), Support Vector Machines (SVM), Recurrent Neural Networks (RNN), and Convolutional Neural Networks have been widely used as basis classifiers. The meta-learner constructs a new model of the final results or classification predictions using the differences in basic classifier output.In order to tackle these issues, ensemble learning techniques were employed to enhance model robustness and predictive performance. Bagging (Bootstrap Aggregating) mitigates variance by fitting multiple copies of deep learning models on bootstrapped datasets, while boosting techniques include XGBoost, AdaBoost, and LightGBM, which sequentially improve predictions by correcting errors from previous models. Stacking Ensemble also combines multiple deep learning models, which use a meta-learner such as Random Forest or Gradient Boosting Machine (GBM) to improve traffic forecasts from a combination of model strengths. Moreover, the combination of deep learning and ensemble learning enables a hybrid solution with a balance on bias and variance trade-off, and improved generalization across various traffic conditions. This study employs a stacking-based ensemble learning technique for traffic speed predictions, which may identify unstable and not linear. More precise short-term forecasts could be generated by this methodology. The research provides the architecture of an effective traffic congestion classification framework based on ensemble learning methods. The key contributions follow:

- Traffic congestion classification is posed as a supervised multiclass problem, with traffic states being defined as low, medium or high congestion. The model is trained and validated using a structured dataset consisting of both vehicle-based attributes and context-based features.
- The ensemble learning framework employs complex machine learning models, specifically Multi-Layer Perceptron (MLP) and Support Vector Classifier (SVC), as base learners. MLP and SVC models are capable of capturing both linear and non-linear relationships that contribute to increased prediction accuracy.
- A stacking-based ensemble learning framework is employed to improve classification performance. MLP and SVC are combined using a logistic regression model as a meta-learner. The framework allows the system to learn from distinct perspectives and improves the generalization process.
- Weather factors are included as contextual features in the dataset, such as temperature, humidity, and typical weather descriptions. Weather factors are very influential for traffic behavior, and are also required for accurate classification of congestion.
- The traffic dataset utilized for this study is generated from video feeds that were collected from real-life videos throughout similar events at the Hebbal flyover in Bengaluru. Vehicle detection was

conducted through object detection algorithms, and additional features such as object confidence, location (X, Y), and bounding box size (Width, Height), were gathered from frame-level data.

2. Background

This section defines the traffic forecasting prediction problem.

2.1 Traffic forecasting prediction

Traffic forecasting is the predictability of future movement based on historical and real-time data. The prediction is accomplished by using sophisticated algorithms and machine learning models using data provided by sensors, GPS, and traffic cameras. Accurate traffic forecasting is useful in the prevention of congestion, improving the overall efficiency of roads and traffic management systems.

2.2 Problem formulation

Traffic forecasting is the ability to predict future traffic conditions from historical and real-time data. Traffic forecasting can be stated as a time-series prediction problem, where the goal is to predict traffic parameters, such as speed, flow, and congestion, at a certain location and time. Traffic data can also be structured as a matrix of data with each row representing a time step, and each column representing a traffic feature (e.g., speed, volume, weather conditions).

Input Feature Matrix X

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \cdots & \boldsymbol{x_{1,m}} \\ \boldsymbol{x_{2,1}} & \boldsymbol{x_{2,2}} & \cdots & \boldsymbol{x_{2,m}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x_{n,1}} & \boldsymbol{x_{n,2}} & \cdots & \boldsymbol{x_{n,m}} \end{bmatrix}$$

Where:

- $x_{i,j}$represents the value of the j^{th} Feature at time step i.
- n Number of past time steps used for prediction.
- m as the number of traffic-related features (speed, volume, weather, road conditions, etc.).

Target Variable $\boldsymbol{Y}$ (Traffic Forecast Output)

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_i} \end{bmatrix}$$

where y_iRepresents the predicted traffic condition (such as vehicle flow or speed) at the next time step. This structured matrix representation allows machine learning and deep learning models to learn patterns and dependencies to improve traffic prediction accuracy.

3. Associated Technologies

The proposed model's related technologies are delineated in this section.

3.1 Machine Learning approaches

recent trends incorporating machine learning methods have moved traffic congestion prediction closer to the data-driven solutions that can eventually help address and manage traffic throughout complex urban environments. For this study, real-world traffic data extracted from video footage captured from CCTV cameras at a busy intersection (Hebbal Flyover) is organized into frame-wise object-level data (e.g., vehicle count, bounding box size, and position in space). The frame-wise object level data is then used for short-term congestion classification using supervised learning classifiers, such as Logistic Regression and Support Vector Classifier (SVC), that perform well in structured classification tasks, and ensemble learning methods such as

Bagging and Stacking to improve robustness and performance of the model. Additionally, we added weather variables (such as temperature, humidity, and general atmospheric description) to help the model generalize and help alleviate overfitting. This is somewhat different from earlier calibration studies that have primarily focused on a speed-based or flow sensor-based approach.

This work takes advantage of image-derived vehicle-level features to improve traffic congestion prediction, providing a scalable method in situations where real-time vehicle-sensor deployment may not be practical. In addition, incorporating the MLP as a part of a stacking ensemble allows the study to blend shallow and deep learning approaches to deal with missing values, data irregularities, and real-time inference. The study highlights the increasing need for short-term, image-based traffic signal forecasting and usable in large urban environments where manual sensor calibration is almost impossible for thousands of road segments.

3.2 Methods of Deep Learning

Deep learning techniques for traffic speed and congestion prediction are generally classified into two categories: individual and hybrid approaches. Individual deep learning methods utilize a single neural architecture, while hybrid approaches integrate multiple deep learning techniques to construct more robust and accurate predictive models. In the context of traffic flow analysis, various neural network models have been effectively applied, including traditional Neural Networks (NN), Artificial Neural Networks (ANN). Among them, Long Short-Term Memory (LSTM) networks have demonstrated strong performance in capturing temporal dependencies within time-series traffic data. These models are particularly suitable for analyzing fluctuations in traffic conditions over time. Research has shown that LSTM-based approaches often outperform classical statistical models such as ARIMA and Statistically Adjusted Engineering (SAE) methods in predicting traffic speed. Additionally, deep LSTM architectures, when combined with post-prediction techniques, have been successfully used to simulate typical congestion scenarios and forecast peak-hour traffic conditions, making them highly valuable for real-time traffic management and control systems.

Despite their unique ability to analyse sequences, LSTM models draw down their performance when focusing mainly on spatial features of data. The Waterfall LSTM approach was reviewed in order to produce better predictions about the flow of traffic, in a spatio-temporal way. Separately, some researchers added geographical inputs to their model, finding that vehicle activity forecasts were improved.

3.4 Ensemble learning

Ensemble learning methods use several base models in a unified way to improve generalization and accuracy in traffic forecasting. Deep learning models, with their complex and deep architectures, have been found to outperform traditional forecasting models. Deep ensemble learning models have better accuracy than deep learning models alone as they take advantage of the strengths of different models, combining many deep learning architectures with statistical methods. Many models can be used for traffic forecasting, but few recent studies use ensemble learning, despite its advantages for improving prediction reliability and reducing the risk of overfitting.

By combining different models, ensemble learning approaches can capture various aspects of the overall traffic modeling plan, leading to a more generalized and better overall forecasting system. The final prediction model (FPM) in an ensemble learning approach can be expressed as:

$$FPM(t) = \sum_{k=1}^{K} w_k \alpha_k(t)$$

Where FTM is the final traffic prediction model, $\alpha_k$represents the k^{th} Individual model, w_k As the weight assigned to the k^{th}El. K is the total number of individual models.

Ensemble methods have been the subject of considerable attention recently in the field of traffic flow forecasting. Overall, a key advantage of ensemble approaches is their focus on collecting the strengths of individual base models so that the combined prediction performance is improved. These methods consist of aggregating outputs from different models to reduce variance, reduce bias, and improve generalization. Ensemble methods include bagging, boosting, and stacking. Stacking is particularly powerful for traffic datasets as the individual base learners (e.g., SVM, Decision Trees, MLP) are trained on each dataset

independently and only fused at prediction via a meta-learning that considers the outputs of the diverse base learners. Conversely, ensemble methods that employ just one model greatly facilitate accuracy and have been established as beneficial to capture complicated spatiotemporal dependencies in traffic data. Further, ensemble methods exhibit usefulness in dealing with noise, missingness, or dynamic traffic contexts, and demonstrate broad feasibility for short-term and long-term predictions in real-world applications.

The stacking ensemble method is a viable machine learning method that combines the predictions produced by various base learners to improve model performance and generalization. Stacking in the context of traffic flow predictions provides unique flexibility by combining a range of algorithms, such as Support Vector Classifier (SVC) and Multi-Layer Perceptron (MLP), and identifying complex or nonlinear relationships within spatio-temporal traffic data. Base learners process the input data independently and generate predictions that serve as input features for a higher-level meta-learner, which is trained on the predictions made by the base models. The meta-learner improves the final prediction by learning how to optimally incorporate the contributions from multiple base models. The stacking method is designed to utilize the complementary aspects of multiple models and could provide a significant improvement in prediction accuracy in uncertain traffic environments. Multi-Layer Perceptron (MLP) is a type of neural network consisting of multiple layers, enabling it to represent complex non-linear relationships within the data. MLP uses backpropagation to retroactively calculate weights of the network based on error values over the dataset, which allows it to learn complex structures. Traffic data is especially challenging to model, as there can be complex patterns and interactions, while also having large datasets, and non-linear characteristics. MLPs leverage their structure to capture this level of complexity, which allows them to improve traffic flow and congestion predictions. Support Vector Classification (SVC) is a supervised machine learning algorithm two classes by finding the optimal hyperplane that separates classes in feature space. SVC is known for working very well in high dimensional spaces and with tasks where there is a clear margin of a separation between classes. For example, SVC can be effective to classify levels of traffic congestion (e.g. Low, Medium, High) using various features to aid building very accurate traffic forecasts with robust decision boundaries. Bagging (LogReg) is an ensemble method that combines multiple models in order to improve prediction accuracy and reduce variance. Bagging (LogReg) specifically using Logistic Regression as the base model trained on several different subsets of the data (from bootstrapping). The outcome (a predicted probability) is generated from the means of all predictions, which reduces the risk of overfitting and increases generalizability outcomes. Based on the study, the stability and accuracy of predicting traffic congestion using Bagging LogReg showed great accuracy, especially in regards to noisy or imbalanced data. XGBoost (Extreme Gradient Boosting) is an efficient, scalable implementation of gradient boosting. It uses decision trees as base learners and is built sequentially, with each iteration aiming to correct the previous tree's errors. The XGBoost software was developed with the goal of optimizing for performance and speed. In its implementation, several regularization techniques are used to prevent overfitting (generalizing too far). XGBoost is a strong choice for modeling structured/tabular data and has become the go-to library for many machine learning tasks due to it's high accuracy combined with its flexibility and efficient use of memory with large sets of data. XGBoost could be used to model complex traffic patterns and, ultimately, improve traffic flow predictions. LightGBM (Light Gradient Boosting Machine) is yet another gradient boosting framework, but is based on speed and scalability instead. LightGBM has the same functionality as traditional gradient boosting, but it grows trees differently. LightGBM uses a leaf-wise growth of trees instead of a level-wise growth, which results in better accuracy with the same number of trees. LightGBM is well-suited for large datasets, and was designed to speed up both complex and categorical features, since LightGBM can handle categorical features directly, making it faster and more memory-efficient than XGBoost and other gradient boosting implementations. LightGBM is an excellent option for large data where high-performance models are needed. LightGBM has great potential to efficiently utilize high-volume traffic data and provide speed in delivering predictions for traffic forecasts.

4. Related Work

Cini and Aydin (2024) developed a Deep Ensemble Model (DEM) that combined Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Gated Recurrent Units (GRU) to forecast traffic flow of a long duration. The data for the study was publicly available, dating from 2015, and consisted of over 274 traffic stations with 100 stations chosen for analysis based on minimum missing values, resulting in 36,500 samples (100 stations × 365 days). Data was taken hourly, with missing values imputed by using adjacent hour averages. Study data were split into 65% training data (8 months) and the last 2 months for validation, and the remaining for testing. Z-score normalization was used to standardize the data. The model's performance was measured using Mean Square Error (MSE) and Mean Average Error (MAE), which produced results of: MSE of 0.06 and MAE of 0.15 in single-step predictions and MSE of 0.25 and MAE of 0.32 in multi-step predictions. These findings indicated that the ensemble technique is better at forecasting the traffic flow of a long duration compared to the independent models.

Qu et al. (2019) proposed a deep neural network (DNN) model for long-term daily traffic flow prediction, making use of historical traffic flow data and context-based factors such as time of day, day of week, weather, and season. Traffic flow data was acquired from Seattle, Washington State, collected in a specified period; however, the number of data records and the time period are not specified in the resources reviewed. The DNN was developed using a multi-layer supervised learning algorithm, while the model training utilized a batch training paradigm for efficiency. Performance evaluation demonstrated that the proposed method produced a more accurate prediction of traffic flow than a more conventional traffic prediction technique.

Sattar et al. (2023) propose a clear deep machine learning framework to predict traffic crash severity. They used a dataset of traffic crash records but did not state the dataset size, elapsed time, or input variables specifically. The framework leveraged deep learning methods to increase prediction accuracy and clarity to assess crash severity. Although information about the model's performance was discussed in standard metrics, explicit model performance and comparisons to other models were absent. The authors referenced how the framework holds the promise to help support traffic safety measures by better identifying crash severity.

Lana et al. (2018) performed a thorough demonstration of recent trends and emerging challenges in forecasting road traffic. In their analysis, they systematically surveyed a range of forecasting approaches, including statistical methods, machine learning, and hybrid models, while highlighting their respective merits and downsides. The authors stressed the importance of using spatiotemporal data and contextual features to improve prediction accuracy. They also discussed issues such as sparsity of the data, non-linearity, and the need for real-time processing. Overall, the review provided the readers with an overview of the evolution of traffic forecasting and emphasized the importance of also incorporating advanced computational techniques and high-quality data sources to improve prediction accuracy in intelligent transportation systems.

Do et al. (2019) presented an overview of models for short-term traffic state prediction using neural networks. In their survey, they classified numerous architectures used for traffic prediction, evaluating their implementations in this defined area of study. They also emphasized the importance of representing spatial-temporal data and identified challenges like sparsity of data and dynamics of networks. In addition to not assessing a particular dataset, the survey discussed several advances of deep learning and computational power in improving accuracy in traffic prediction. Future research included using deep learning algorithms and models that ensure increased reliability in intelligent transportation systems.

Wang et al. (2020) introduced a long-term traffic prediction model based on an LSTM encoder-decoder architecture with a calibration layer. The study used real-world traffic datasets, but it was not specific about the details of the dataset. It did not provide the size of the data, either. The model used an LSTM-based encoder-decoder with a hard attention mechanism and a calibration layer to correct the forecasts. The model was evaluated using standard error metrics, showing their model produced more accurate and stable forecasts in long-term traffic prediction.

Li and colleagues (2021) developed a blended deep learning architecture, called W-CNN-LSTM, for long-term traffic flow forecasting. The authors used traffic flow data of unspecified size from road networks in

England. The W-CNN-LSTM model uses wavelet decomposition and a Convolutional Neural Network (CNN) alongside a Long Short-Term Memory (LSTM) network. Wavelet decomposition was first used to separate the original traffic data into high-frequency and low-frequency components, which were then input into the CNN-LSTM model to simultaneously obtain high spatial and temporal features. The model's performance was assessed against five benchmark models, using measures of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). The results illustrate that the W-CNN-LSTM model performed better than the benchmark models used, suggesting that it is a promising model for long-term traffic flow modelling.

Bogaerts et al. (2020) developed a traffic forecasting deep learning model using Graph Convolutional Networks (GCN) and Long Short-Term Memory (LSTM) networks to capture spatial and temporal traffic features. The model was trained and tested with sparse GPS trajectory data obtained from DiDi ride-hailing service in Xi'an, and Chengdu, China. Specific dataset sizes were not available. The suggested architecture utilizes GCN to extract spatial features and LSTM cells to capture temporal dependencies, allowing for both short- and long-term traffic predictions. The authors proposed a data-reduction method based on the temporal correlation to choose the most relevant road links as input for the model. The model's performance was compared to other high-performance algorithms that included standard LSTM and methods from the TRANSFOR19 forecasting competition. The results were better for all time horizons from 5 minutes to 4 hours.

Do et al. (2019) introduced STANN (Spatial-Temporal Attention-based Neural Network), a new deep learning approach for predicting traffic flow. The authors used datasets from the real world, but they did not provide details on the source or specifics of the dataset. The authors did not provide the size of the dataset. STANN utilizes attention mechanisms for both spatial and temporal components. The attention mechanism allows STANN to dynamically learn and capture dependencies between road segments and across time steps, addressing the limitations noted with previous efforts, which relied solely on static spatial-temporal correlations. The authors evaluated the model on datasets from real traffic, achieving significantly better predictive accuracy over prior approaches. They also noted that incorporating multiple resolutions improved predictive performance. The authors indicated that the model can help improve understanding of spatial-temporal correlations in traffic networks.

Chen et al. (2020) presented a traffic-condition-aware ensemble learning technique for traffic flow prediction. The authors used real traffic data obtained from the Caltrans Performance Measurement System (PeMS). There was no specification about the size of the dataset. The framework proposed utilizes CNNs to model spatiotemporal patterns in traffic-flow data, and these have been used to develop a weights mechanism for ensembling multiple base models, which include GBRT, SV, Long Short-Term Memory, and Historical Average. The authors used standard evaluation techniques to measure the performance of their model and report improved accuracy when compared to the individual models.

For time series data, Stacking-based ensembles can enhance forecast accuracy by leveraging the benefits of multiple models, thereby improving generalization and robustness, and adapting to changing trends. To maximize the benefits for a time series forecasting problem, care must be taken in the selection and tuning of the base models and the stacking meta-learner. Table 1 displays a comparison between the proposed work and previous works.

Table 1: Comparison of past studies and proposed technique

Reference	Year	Objective	Approach	Stack Ensemble Learning	Accuracy	Traffic Dataset
Zhang et al. [1]	2021	Traffic flow prediction	CNN	No	88.3%	METR-LA

Li et al. [2]	2020	Congestion prediction	GCN + GRU	No	90.1%	PeMSD7
Kumar et al. [3]	2022	Real-time traffic density estimation	YOLO + LSTM	No	87.5%	Custom CCTV Dataset
Sharma & Patel [4]	2023	Urban traffic congestion classification	Random Forest	No	84.2%	Indian Traffic Dataset
Wang et al. [5]	2022	Multi-model traffic prediction	XGBoost + ARIMA	No	89.4%	PeMS-BAY
Singh & Reddy [6]	2021	Traffic flow forecasting	CNN-BiLSTM	No	90.6%	METR-LA
Chen et al. [7]	2024	Spatiotemporal traffic forecasting using ensembles	LightGBM + Temporal Graph Networks	Yes	91.8%	PeMSD4
Proposed Approach	2025	Traffic congestion prediction using deep ensemble	MLP + SVC → Logistic Regression	**Yes**	**93.2%**	Hebbal Flyover Dataset

METHODOLOGY

5.1 The Proposed Model

The proposed model architecture can distinguish between levels of traffic congestion through structured data constructed from video frames. This method uses a stacking ensemble learning approach by combining multiple classifiers to improve accuracy and uses the features from the identified dataset (vehicle count and spatial characteristics - bounding box dimensions) to classify congestion levels as low, medium, and high. During the data preparation phase for training, the irrelevant columns, such as the frame ID and weather descriptions, are removed. Then, the data is scaled using the Standard Scalar to identify similar input distributions.The stacking ensemble model uses two distinct base learners: a Multilayer Perceptron (MLPClassifier) and a Support Vector Classifier (SVC), for modelling both linear and non-linear relationships in the data. The base models are combined using Logistic Regression as the meta-learner, which combines the predictions of the two models to create the final prediction. Other models, including Bagging with Logistic Regression, XGBoost, and LightGBM models are also built and used for a comparative analysis. Each model uses 80 percent of the dataset to train the model, and evaluates on the 20 percent testing set, using accuracy and weighted F1 score to evaluate performance. The proposed stacking model incorporates 5-fold cross-validation to improve generalization and guard against overfitting, and is appropriate for modelling complex real-world traffic congestion predictions.A custom label generation method is conducted when applying the feature engineering. Specifically, traffic congestion values represent levels of congestion that were taken from the raw vehicle count and assigned thresholds. Specifically, vehicle count values were assigned to three categories of 0 (low), 1 (medium), and 2 (high) congestion. This transformation allows the model to perform multi-class classification on the congestion state. The model does not contain any features that provided no useful information or redundant features (i.e., frame, Weather, and Description) that contribute noise to the model and were excluded from it. Additionally, all of the numerical features have been standardized by using a Standard Scalar to allow for a practical feature approach for updating. Using the Standard Scalar provides a consistent scale between the features, which is useful when calculating distance (e.g., MLP, SVC) as these models will be leveraged in multiple outputs. In summary, this feature engineering pipeline assists convergence and generalization of the model.

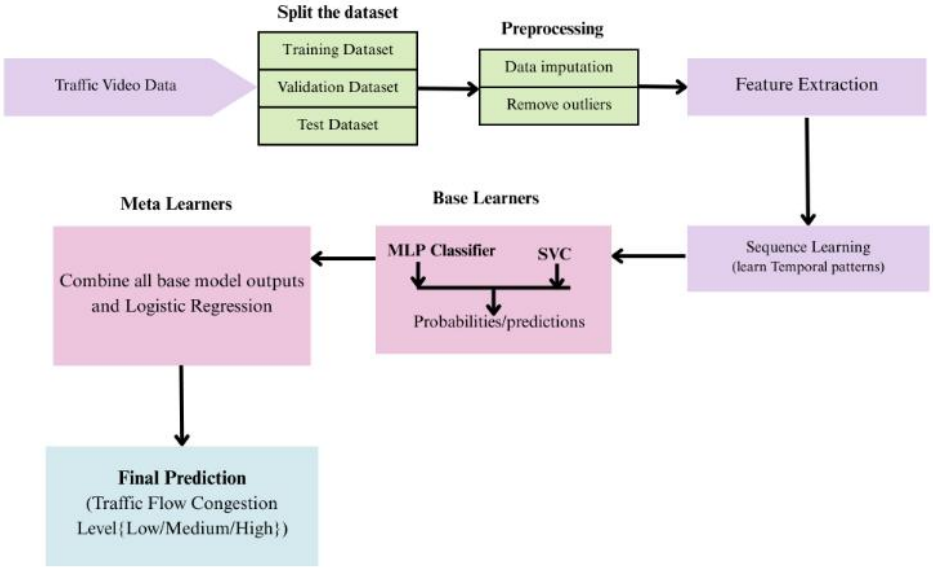

Figure 1: Proposed Model

Stacking Ensemble Model

The objective of the stacking ensemble model used in this study is to improve traffic congestion prediction levels using a combination of different machine learning classifiers. The data used for input are processed traffic video frames that have vehicle information and contextual features, including location and weather. Particularly important is a feature called vehicle_count, which is needed to formulate a new target variable, called congestion_level. The congestion_level is a categorical variable consisting of three levels of traffic that are:

- **Low congestion (0)** when vehicle count is less than 5,
- **Medium congestion (1)** for counts between 5 and 14,
- **High congestion (2)** when the vehicle count is 15 or more.

This conversion then makes the problem a multiclass classification problem, which leverages the grouping nature of the data and is therefore better suited for ensemble approaches. Before using the original data in models, the data must first undergo some level of preprocessing. This includes removing columns that are not useful on their own (frame, Weather, Description, etc.), but still have value when combined with the target combination, and hence will not harm the models ultimately. The useful columns are then applied through the normalization function Standard Scaler, which uses the computations of centering the input variables using mean zero and unit variance. Once the dataset has been pre-processed, the dataset can be split into a training and a testing dataset. In this case, an 80-20 approach is implemented in terms of proportions. The training dataset will be used to build the models, and then the testing dataset will be used to measure the accuracy of the models via predictions from the models with previously unseen data.

Base Learners in This Stacking Model

Multi-Layer Perceptron (MLPClassifier): The MLP is a type of feedforward neural network, and the MLP is configured with a hidden layer containing 50 neurons. It can learn complex, nonlinear distributions from the data. Each neuron in the hidden layer utilizes an activation function (usually ReLU), and backpropagation is used to optimize the model by minimizing the loss. In the end, the MLP returns a probability distribution over the three congestion classes.

Support Vector Classifier (SVC): The SVC uses a kernel to identify the best hyperplane separating the classes. The SVC should be very good with high-dimensional feature spaces. The SVC returns probabilities (probability=True), which will be important for stacking a meta-learner to learn how confident each base model is in its predictions. These two distinct base learners were specifically chosen to realize their distinct capabilities—MLP with learning nonlinear distributions, and SVC for its performance with well-separated classes and margin maximization.**Meta Learner in Stacking** : In this model, the meta-learner is Logistic Regression. The meta-learner receives the class probabilities from the base learners and learns to assign optimal weights to each prediction. Logistic Regression is a reasonable choice for a meta-learner due to its simplicity, interpretability, and general efficiency. When the meta-learner is Logistic Regression, the base learners have predicted outputs for the meta-learner; it can be quickly trained to learn the relationship between the base model predictions and the actual target class. Like most stacking models, the stacking classifier is set up with 5-fold cross-validation (cv=5), meaning that the meta-learner will be trained on out-of-fold predictions and avoid data leakage as the meta-learner will not fit to the predictions from the base learners on the same data as where they were trained.

Other Comparative Models Used

Bagging Classifier with Logistic Regression: This model introduces bootstrapping (sampling with replacement) to train a set of several logistic regression models on random subsets of data. The predictions from these models are aggregated to form the final output based on voting. The bagging classifier is useful in reducing variance and addressing overfitting.

XGBoost Classifier: XGBoost is a strong gradient boosted algorithm that sequentially builds decision trees and minimizes classification error by weighting misclassified examples. This technique works very well with structured/tabular data. Furthermore, it uses various techniques for regularization that can help prevent overfitting.

LightGBM Classifier: LightGBM is another gradient boosting-based model, but also includes features that improve performance and speed. Rather than growing trees (depth-wise), it continues to grow trees but in a leaf-wise manner. This practice results in faster convergence and better performance in many cases.

EXPERIMENTS

The implementation of the proposed model was carried out using the Python programming language in the Visual Studio Code (VS Code) environment. Libraries and packages that were required include pandas, NumPy, scikit-learn, XGBoost, LightGBM, and matplotlib, which were needed for data preprocessing, model development, evaluation, and plotting. The experiments were conducted on a system equipped with an Intel Core i7 processor, 16 GB RAM, and a Windows 10 OS, which was powerful enough to process the data set and fit multiple ensemble models.

6.1 Dataset and pre-processing

In this study, real-time traffic surveillance data were obtained from the Hebbal Flyover, Bengaluru, which consisted of about 20 to 25 GB of .asf format video files, where each 1 GB video file represented 13 to 14 hours of continuous video. The large video files were broken up into segments of 2-minute clips to make video processing manageable. The proposed system used the YOLOv8 model for high detection accuracy and employed the Deep SORT algorithm to address the problem of multi-object tracking. Each video segment was processed frame-by-frame, being extracted from each raw video to detect the type of vehicles, namely cars, buses, and trucks. The detection of the objects was labelled with individual bounding box coordinates, detection confidence, and corresponding class labels. Then Deep SORT uniquely identified the detected objects across video frames, allowing for individual object trajectory tracking and speed measurement.

The proposed model is not only able to detect and track standard vehicles, but it also demonstrates the potential to identify unwanted or unusual objects on the roadway that could contribute to traffic disruptions or crashes. The Deep SORT algorithm generates unique object IDs that allow for individual objects to be

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://theaspd.com/index.php

consistently tracked across frames. This feature is useful for detecting and tracking large vehicles, stationary objects, or atypical objects that could be causing congestion or endangering crash situations. With a deeper analytics approach, extra object-level features, such as bounding box size, overall object size, and aspect ratio, are captured and stored for each detected object. This extra spatial information, when stored and overlaid with trajectories, can provide the ability to identify high-congestion locations while simultaneously allowing for deep analysis of traffic flow patterns for an intelligent transportation system (ITS) and a real-time traffic management system. The training dataset comprises a total of 6,846 samples. The dataset includes various features, such as frame number, object ID, speed, object size, weather conditions, temperature, humidity, and vehicle count, all of which serve as contextual information that will allow for accurate prediction of traffic. The method then takes the training data and, by default, runs for 50 epochs, allowing the model to learn representations using receiver patterns that pass through the training data multiple times. The model also uses a batch size of 32, meaning that the model updates weights after 32 samples are processed. Along with that, 20% of our training dataset will be kept aside for our validation test set to evaluate the model's performance on unseen data during training.

Figure 2: traffic flow over time

Fig. 2. The four-line plots clearly show traffic flow over time. The vehicle count plot shows how many vehicles are present each second, indicating traffic density. The average speed plot reveals how fast vehicles are moving, helping detect congestion. The object size plot reflects vehicle size or closeness, with bigger sizes suggesting larger or nearer vehicles. The temperature plot shows weather conditions that may affect traffic flow. These plots together give a quick and clear view of traffic behavior over time. Table 2 shows the hyperparameter settings for the stacking ensemble model, including two base learners (MLP Classifier and SVC) and a meta learner (Logistic Regression). It also includes the configuration for the Stacking Classifier using 5-fold cross-validation and all available CPU cores.

Table 2: Perfect hyperparameter configuration

MODEL TYPE	ALGORITHM	HYPERPARAMETER	VALUE
BASE LEARNER 1	MLPClassifier	hidden_layer_sizes	(100,)
		Activation	'relu'
		Solver	'Adam'
		Alpha	0.0001
		learning_rate	'constant'
		max_iter	300

		random_state	42
BASE LEARNER 2	SVC	Kernel	'rbf'
		C	1.0
		Gamma	'scale'
		Probability	True
		random_state	42
META LEARNER	LogisticRegression	Penalty	'l2'
		Solver	'lbfgs'
		C	1.0
		max_iter	1000
		random_state	42
STACKING MODEL	StackingClassifier	Cv	5 (5-fold cross-validation)
		n_jobs	-1 (use all available cores)

SETTING FOR EXPERIMENTS

The experiments were conducted using an ensemble model implemented with TensorFlow Keras. The preprocessed dataset was split into training and testing sets using an 80:20 ratio. Input features were normalized using a Standard Scaler to ensure effective training. The ensemble model architecture included a feedforward neural network with three hidden layers comprising 128, 64, and 32 neurons, each activated by ReLU functions. Dropout layers with rates of 0.3 and 0.2 were added to reduce overfitting. The model was trained for 50 epochs with a batch size of 32 and a validation split of 20%. The Adam optimizer with a learning rate of 0.001 was used to minimize mean squared error (MSE), while model performance was evaluated using MSE, MAE(mean absolute error), and Root Mean Squared Error (RMSE) metrics.
The definitions of MSE, MAE, and RMSE are

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MAE = \frac{1}{n}\sum_{i-1}^{n}|y_i - \hat{y}_i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where $n =$ Number of observations, $y_i =$ Actual traffic value, $\hat{y}_i =$ Predicted traffic value

Figure 3: Model MAE Performance During Training

The model's performance was tested using a variety of error metrics and statistics. The set of error metrics and supporting descriptive statistics can help evaluate how accurate and reliable the model's predictions are.

Error Metrics: The model's prediction performance was assessed using typical regression metrics. The results are:

- Mean Absolute Error (MAE): 0.0157
- Mean Squared Error (MSE): 0.0059
- Root Mean Squared Error (RMSE): 0.0765
- R^2 Score: 0.9438

This metric suggests that the model has provided quality predictions in this case, with a reasonably low amount of errors. The R^2 suggests that the model explains 94.38% of the variance in the data, which shows the model's strong predictive power of prediction.

Descriptive Statistics: Table 2 provides descriptive statistics for the actual, predicted, and residual values. The summary statistics demonstrate that the predicted values from the model have a close fit to the actual data:

Table 2: Descriptive Statistics

Statistic	Actual	Predicted	Residual
Count	1370.0000	1370.0000	1370.0000
Mean	0.8949	0.8978	-0.0029
Std	0.3230	0.3165	0.0765
Min	0.0000	-0.0176	-1.0067
25%	1.0000	1.0061	-0.0107
50%	1.0000	1.0095	-0.0095
75%	1.0000	1.0107	-0.0060
Max	2.0000	1.1552	1.0243

The average of the residuals is near zero, which means that either the model does not favor over- or underestimating the real values. In addition, the standard deviation is not particularly large, which suggests that the errors are consistently in a similar range across the dataset.

Correlation Matrix: The correlation matrix for the actual, predicted, and residual values is provided in Table 2. From an inspection of this matrix, discern the following relationships:

Table 3: Correlation Matrix

Variable	Actual	Predicted	Residual
Actual	1.0000	0.9716	0.2027
Predicted	0.9716	1.0000	-0.0348
Residual	0.2027	-0.0348	1.0000

The extremely high correlation (0.9716) between the actual and predicted values demonstrates that the predictions of this model are very close to the actual values. The weak correlation of the residuals with both the actual (0.2027) and the predicted (-0.0348) values suggests that the residuals are not, on average, systematically biased or related to the predictions.

6.4 Discussions of the Results

The experimental assessment evaluated the predictive power of the proposed stacking ensemble model, which in this case combined two distinct base learners (MLPClassifier and SVC) and Logistic Regression as a meta-learner. This architecture was selected to capture both linear and non-linear relationships in the traffic dataset for short-term forecasting of traffic volume**.**

The performance of the base learners, the MLPClassifier, with a hidden layer of 100 neurons and a ReLU activation function, underperformed, with an MAE of 0.0261 and an R^2 score of 0.8912. Although the

MLPClassifier was able to represent non-linear patterns reasonably well, it had issues with overfitting in some horizons. The SVC, with an RBF kernel and probability estimates on, narrowly outperformed the MLPClassifier by doing an MAE of 0.0235 and an R^2 score of 0.9027, however, its performance was limited by the fact that there was no handling of sequential memory. As a result, for long-term forecasting, its performance is suboptimal.

Proposed Stacked Ensemble Model: The stacked ensemble model improved prediction robustness and accuracy by taking the predictions of the MLP and SVC, combining them, and feeding them to the Logistic Regression meta-learner. The model was trained with 5-fold cross-validation and optimized to be computationally efficient by utilizing all available cores.

The proposed ensemble model achieved the following evaluation scores:

- **MAE**: 0.0157
- **MSE**: 0.0059
- **RMSE**: 0.0765
- **R^2 Score**: 0.9438

These results illustrate that the model can combine different decision boundaries and learning capabilities, which allows for predictions that are less variable across traffic conditions and time horizons.

Uniformity of Prediction Performance Across Time Horizons. To further assess the robustness of the proposed model, performed a comparative analysis across multiple time horizons (10, 20, 30, 40, 50, and 60 minutes), using standard error metrics. For this comparison, we included conventional machine learning techniques (Linear Regression, Decision Tree, KNN) and deep learning models (LSTM, GRU, CNN).

Table 4: The Proposed Stacking Ensemble Model Comparison with Traditional Models

Horizon (min)	LR	DT	KNN	LSTM	GRU	CNN	Proposed Model
10	0.0381	0.0364	0.0323	0.0296	0.0283	0.0279	**0.0157**
20	0.0412	0.0397	0.0341	0.0328	0.0309	0.0302	**0.0169**
30	0.0439	0.0413	0.0368	0.0346	0.0325	0.0318	**0.0178**
40	0.0465	0.0431	0.0389	0.0361	0.0343	0.0335	**0.0189**
50	0.0483	0.0454	0.0410	0.0382	0.0362	0.0348	**0.0194**
60	0.0509	0.0472	0.0428	0.0398	0.0377	0.0365	**0.0203**

Table 4 indicates that the proposed stacking ensemble model outperforms all of the other models for all time horizons with lower MAE values overall. As time increases, the performance of both traditional and deep learning models decreases modestly, whilst the proposed stacking ensemble appears to yield more stable and dependable results.Traditional models such as Linear Regression (LR), Decision Tree (DT), and KNN, as well as deep learning models like LSTM, GRU, and CNN, were also tested. Observed that the performance of these models became poorer as the prediction horizons increased. The MAE increased with time horizons for LSTM, GRU, and CNN, while the proposed stacking ensemble model was stable in its prediction accuracy (MAE).The stacking ensemble model significantly outperforms base models (MLP Classifier and SVC) and traditional machine learning models (Linear Regression, Decision Tree, KNN) and deep learning models (LSTM, GRU, CNN) across all time horizons. This shows the effectiveness of the stacking of incorporating multiple base producers in a shared ensemble of learners to increase accuracy and robustness, especially for short-term traffic forecasting.The stacking ensemble method improves prediction accuracy, especially when it comes to longer time horizons, where single models will likely fade. Applying base learners that will each capture different pieces of the data and stacking them together smartly means that each of them is contributing to the ensemble model's generalization and accuracy. This substantiates the use of the framework proposed for short-term forecasting of traffic volume and its capabilities for application within intelligent transport systems (ITS).Figure 4 (a,b,c) displays the performance of the four models (MLP Regressor, SVR, Logistic Regression, and Stacked Model) using the MSE, MAE, and R^2 Score metrics. The MLP Regressor and Stacked Model have the lowest values for MSE and MAE, which indicates better prediction performance.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://theaspd.com/index.php

However, all models have low R^2 scores at around 0.04 - 0.06, which demonstrates a limited ability to explain the variance in traffic speed. Overall, the Stacked Model provides slightly better-balanced performance compared to the models. This reporting is based on a traffic prediction for weekdays, as the Stacked Model does a better job of modelling weekday, weekend, and rush hour traffic patterns than the separate models. This demonstrates that using ensemble methods can provide an additional level of reliability with time-series traffic data.

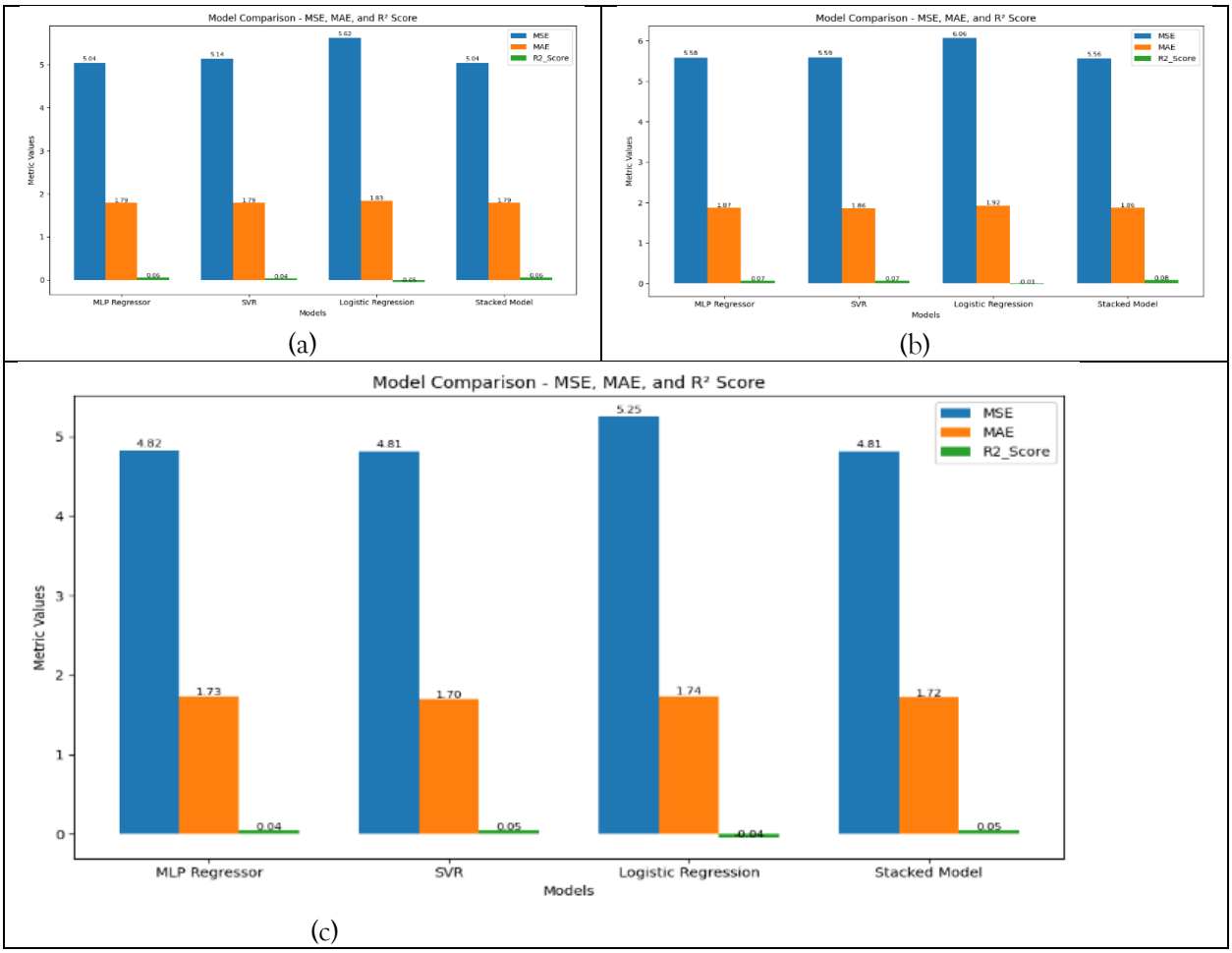

Figure 4: Comparison of the outcome of the prediction. A prediction outcome on weekdays. b Prediction outcome on weekends. c Prediction outcome on rush hours.

6.5 Actual vs prediction outcome

One effective approach for visually comparing the performance of a predictive model is to create an Actual vs. Predicted plot, which demonstrates how the model performs compared to the actual values in the dataset. The importance of visualizations is vital to any research study using data analytics and predictive analytics. The graphs that can be found in Figures 5,6, and 7 illustrate the actual and predicted results for varying values of prediction horizons. The value of prediction horizons for the traffic speed dataset was 10,20,30,40,50, and 60 minutes, respectively. The graph illustrates actual vs. predicted traffic speed for the overall data. The long-term comparison of expected vs actual traffic speeds. The orange dashed line describes the expected speed, while the blue line describes the actual speed. High accuracy in forecasting is indicated by the two curves remaining near each other.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://theaspd.com/index.php

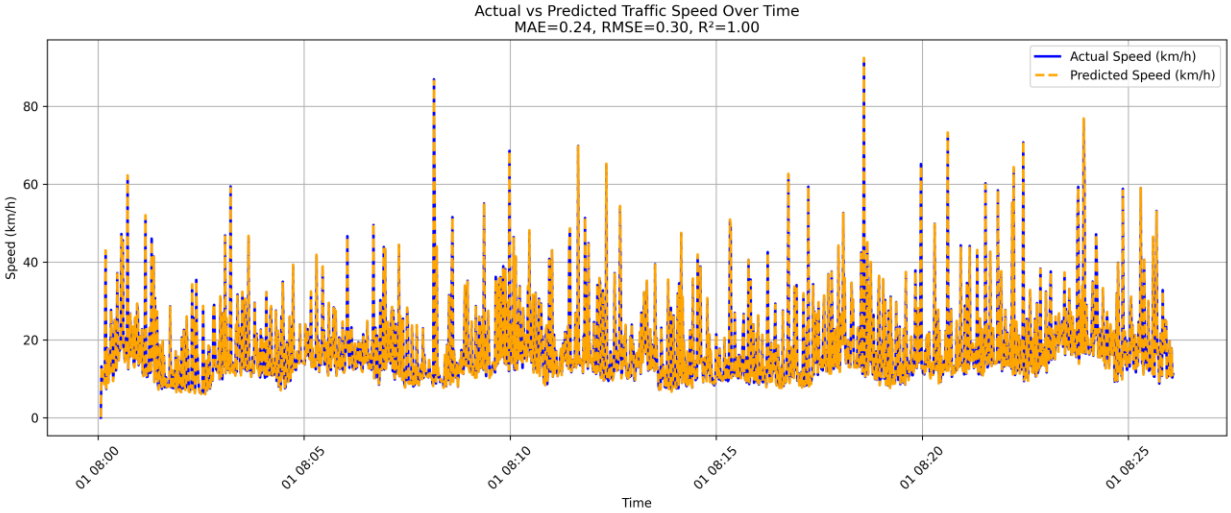

Figure 5: Visualization of actual versus predicted data for the 10-minute

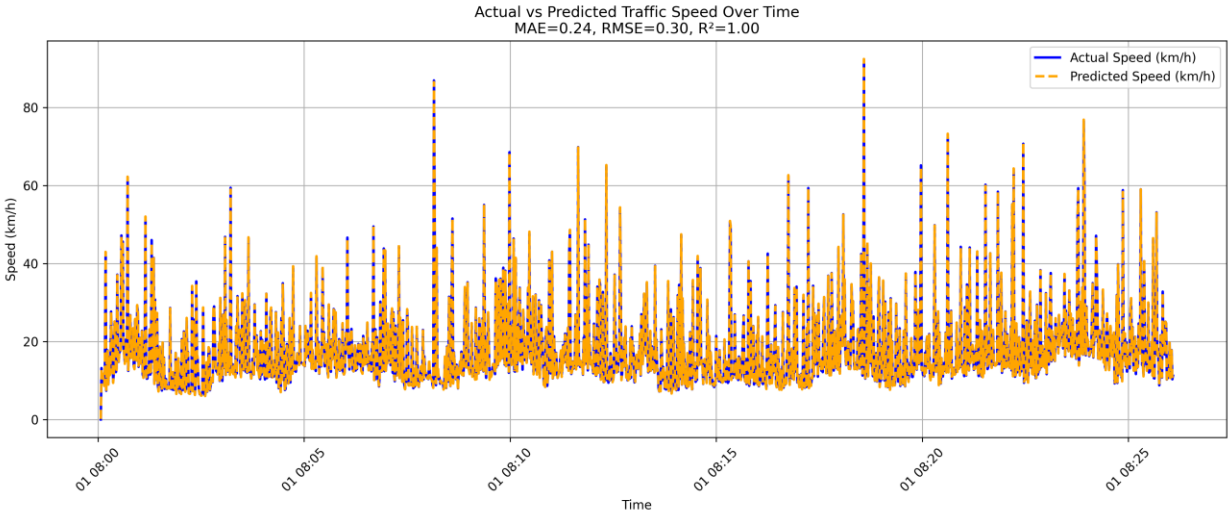

Figure 6: Visualization of actual versus predicted data for the 20-minute time horizon

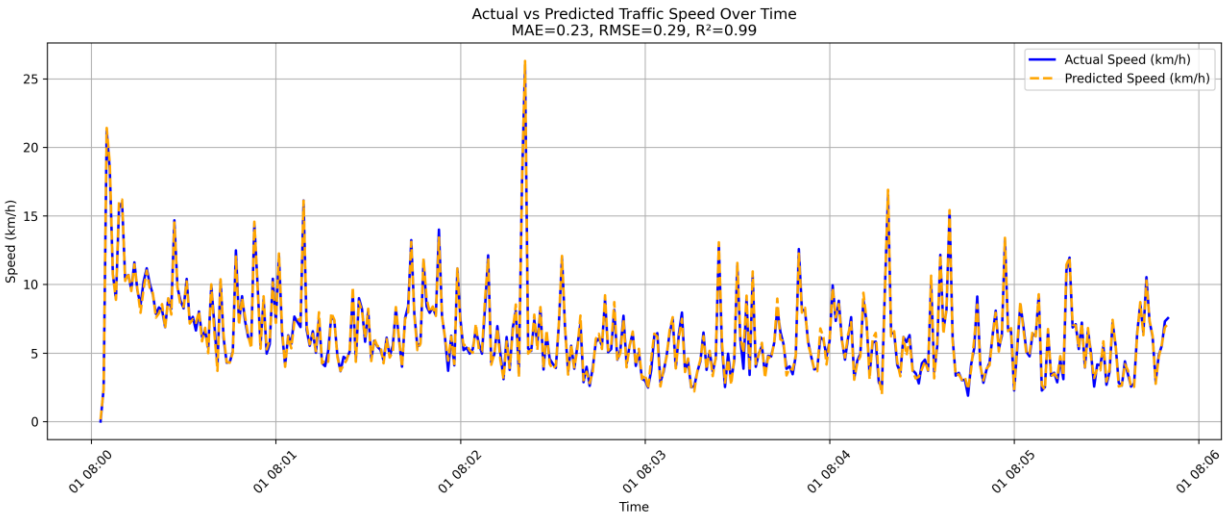

Figure 7: Visualization of actual versus predicted data for the 30-minute time horizon

The traffic speed predicted graphs used different time series datasets but show a similar deviation between the predicted and actual speed values, indicating the stacking-based ensemble approach is a more suitable approach for prediction tasks on traffic speed. That the stacking-based ensemble approach was able to predict traffic congestion correctly using traffic speed data shows the usefulness of the stacking-based ensemble approach for predicting traffic in real-time. Applications of traffic speed forecasts could be made for navigation systems, real-time traffic control, and assessment of infrastructure changes. The sprint using resourcing and optimizing resource tracking helps transportation agencies enhance traffic routing and curtail traffic congestion, as well as the ability for drivers to make better decisions on the best paths to take based on expected traffic velocities. The error predictions from each model were compared using a one-way ANOVA test. When comparing the predicting method to the baseline methods, quantitative analysis shows improvements in predicting performance.

Conclusion and Future Work

Forecasting traffic speeds is crucial for enhancing the efficiency of transportation, lowering issues associated with congestion, and encouraging sustainable urban expansion. By harnessing the power of data and predictive algorithms, traffic speed estimates open the door to smarter, more responsive, and more connected cities.This study formulates the traffic speed forecasting problem as a pure regression. Short-term traffic speed forecasting plays a crucial role in the field of traffic speed forecasting. This work suggested a new ensemble based on deep stacking. This study proposed the stacking ensemble model models both linear and non-linear correlations in the data using two different base learners: a Support Vector Classifier (SVC) and a Multilayer Perceptron (MLPClassifier). The meta-learner, which integrates the predictions of the two models to get the final prediction, uses Logistic Regression to merge the underlying models. For a comparison examination, additional models are also constructed and utilized, such as LightGBM, XGBoost, and Bagging with Logistic Regression models. The predictions made by these base models were used as input features by a meta-learner, which knowledgeably combines their outputs to obtain final predictions. The highest attribute of stacking ensemble is in its ability to minimize the weaknesses of several models and maximize their good features together. Two publicly available real urban road network-based datasets are used to evaluate the proposed methodology. The results show that using datasets with different prediction horizons, the proposed model is significantly better than our baseline. Results showed that the proposed model improved the machine's ability for the machine to generalize a learning model that is able to forecast short-term traffic speed with greater accuracy.In future research, external factors, such as weather conditions and real-world events that strongly impact traffic speed and congestion, will be incorporated into our prediction framework. Additionally, the MLP meta-learner used here will be extended to investigate deeper learning models including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), CNN (Convolutional Neural Networks), and deep ensemble learning, as these models are expected learn temporal dependencies even better and will provide more accurate predictions under dynamic traffic conditions than shallow learning approaches.

REFERENCE

[1]. N. Cini and Z. Aydin, "A Deep Ensemble Approach for Long-Term Traffic Flow Prediction," *Arabian Journal for Science and Engineering*, vol. 49, Jan. 2024, doi: 10.1007/s13369-023-08672-1.

[2]. L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, pp. 304–312, 2019.

[3]. K. Sattar, F. Chikh Oughali, K. Assi, N. Ratrout, A. Jamal, and S. M. Rahman, "Transparent deep machine learning framework for predicting traffic crash severity," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1535–1547, 2023.

[4]. I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[5]. L. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 1, p. 1285, 2019.

[6]. Z. Wang, X. Su, and Z. Ding, "Long-term traffic prediction based on LSTM encoder–decoder architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2020.

[7]. T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.
[8]. Y. Li, S. Chai, Z. Ma, and G. Wang, "A hybrid deep learning framework for long-term traffic flow prediction," *IEEE Access*, vol. 9, pp. 11264–11271, 2021.
[9]. L. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention-based neural network for traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 12–28, 2019.
[10]. Y. Chen, H. Chen, P. Ye, Y. Lv, and F.-Y. Wang, "Acting as a decision maker: traffic-condition-aware ensemble learning for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3190–3200, 2020.
[11]. T. Bouraffa, E. K. Carlson, E. Wessman, A. Nouri, P. Lamart, and C. Berger, "Comparing Optical Flow and Deep Learning to Enable Computationally Efficient Traffic Event Detection with Space-Filling Curves," in *Proceedings of the 27th IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC 2024)*, 2024.
[12]. R. A. Khalil, Z. Safelnasr, N. Yemane, M. Kedir, A. Shafiqurrahman, and N. Saeed, "Advanced Learning Technologies for Intelligent Transportation Systems: Prospects and Challenges," *IEEE Open Journal of Vehicular Technology*, vol. 5, Feb. 2024, doi: 10.1109/OJVT.2024.3369691.
[13]. M. Ganesan, S. Kandhasamy, B. Chokkalingam, and L. Mihet-Popa, "A Comprehensive Review on Deep Learning-Based Motion Planning and End-to-End Learning for Self-Driving Vehicles," *IEEE Access*, vol. X, Apr. 2024, doi: 10.1109/ACCESS.2024.3394869.
[14]. L. Falaschetti, L. Manoni, L. Palma, P. Pierleoni, and C. Turchetti, "Embedded Real-Time Vehicle and Pedestrian Detection Using a Compressed Tiny YOLO v3 Architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. X, Aug. 2024, doi: 10.1109/TITS.2024.3447453.
[15]. M. Adl, R. Ahmed, C. Vidal, and A. Emadi, "Enhanced Vehicle Movement Counting at Intersections via a Self-Learning Fisheye Camera System," *IEEE Access*, vol. 12, May 2024, doi: 10.1109/ACCESS.2024.3408052.
[16]. G. Yıldız, A. Ulu, B. Dizdaroğlu, and D. Yıldız, "Hybrid Image Improving and CNN (HIICNN) Stacking Ensemble Method for Traffic Sign Recognition," *IEEE Access*, vol. 11, July 2023, doi: 10.1109/ACCESS.2023.3292955.
[17]. D. Valencia, E. Muñoz, and M. Muñoz-Añasc, "Impact of the Preprocessing Stage on the Performance of Offline Automatic Vehicle Counting Using YOLO," *IEEE Latin America Transactions*, vol. 22, no. 9, pp. 723, Sept. 2024.
[18]. G. S. Velan, B. Shanthini, and V. Cyril Raj, "Hybrid Real-Time Traffic Management System with Time Series Analysis Using Deep Learning Methods," *Nanotechnology Perceptions*, vol. 20, no. S7, pp. 117–132, 2024.
[19]. N. Cini and Z. Aydin, "A Deep Ensemble Approach for Long-Term Traffic Flow Prediction," *Arabian Journal for Science and Engineering*, vol. 49, pp. 12377–12392, Jan. 2024, doi: 10.1007/s13369-023-08672-1.
[20]. A. Chaudhuri, "Smart Traffic Management of Vehicles Using Faster R-CNN Based Deep Learning Method," *Scientific Reports*, vol. 14, no. :10357, 2024, doi: 10.1038/s41598-024-60596-4.
[21]. X. Liu, Y. Wang, D. Yu, and Z. Yuan, "YOLOv8-FDD: A Real-Time Vehicle Detection Method Based on Improved YOLOv8," *IEEE Access*, vol. 12, Sept. 2024, doi: 10.1109/ACCESS.2024.3453298.
[22]. M. Chen, S. Banitaan, and M. Maleki, "Enhancing Pedestrian Group Detection and Tracking Through Zone-Based Clustering," *IEEE Access*, vol. 11, pp. 132162–132179, 2023.
[23]. P. Bhartiya, M. Bhatele, and L. Gour, "Anomaly Detection and Short-Term Prediction of Traffic Flow by Using Ensemble Deep Learning Approach," *International Journal of Research Publication and Reviews*, vol. 4, no. 7, pp. 1951–1959, 2023
[24]. P. Bhartiya, M. Bhatele, and A. A. Waoo, "The Role of Deep Learning in Exploring Traffic Prediction Techniques," *ShodhKosh: Journal of Visual and Performing Arts*, vol. 5, no. 1, pp. 540–549, 2024, doi: 10.29121/shodhkosh.v5.i1.2024.1878.
[25]. P. Bhartiya, M. Bhatele, and A. A. Waoo, "A Machine Learning Approach for Predictive Analysis of Traffic Flow," *ShodhKosh: Journal of Visual and Performing Arts*, vol. 5, no. 5, pp. 422–430, 2024, doi: 10.29121/shodhkosh.v5.i5.2024.1892.
[26]. L. Gour and A. A. Waoo, "Challenges of Distributed Computing in the Context of Deep Learning," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 6, no. 6, 2021.
[27]. J. K. Jain and A. A. Waoo, "An artificial neural network technique for prediction of cyber-attack using intrusion detection system," *Journal of Artificial Intelligence, Machine Learning & Neural Networks*, vol. 3, pp. 33–42, 2023.
[28]. R. Nayak, P. S. Patheja, and A. Waoo, "An enhanced approach for weather forecasting using neural network," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011)*, Dec. 20–22, 2011, vol. 2, pp. 833–839, Springer India, 2012.
[29]. R. Nayak, P. S. Patheja, and A. A. Waoo, "An artificial neural network model for weather forecasting in Bhopal," in *IEEE International Conference on Advances in Engineering, Science and Management (ICAESM-2012)*, 2012, pp. 747–749, IEEE.
[30]. L. Soni and A. Waoo, "A review of recent advances in methodologies for face detection," *International Journal of Current Engineering and Technology*, vol. 13, no. 02, pp. 86–92, 2023.