TrafficSense: A Machine Learning Approach for Real-Time Urban Traffic Situation Prediction

P. Teja Prakash
CB.AI.U4AIM24136
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
cb.ai.u4aim24136@cb.students.amrita.edu

S.V.S Ankith
CB.AI.U4AIM24147
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
cb.ai.u4aim24147@cb.students.amrita.edu

A. Guru Jaya Surya Yadav
CB.AI.U4AIM24101
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
cb.ai.u4aim24101@cb.students.amrita.edu

J. Tej Krishna Sai
CB.AI.U4AIM24117
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
cb.ai.u4aim24117@cb.students.amrita.edu

***Abstract*—This paper presents *Traffic Sense*, a dual-model machine learning framework for real-time traffic situation classification. A regression model first predicts total vehicle count, which is then used by a classifier to label traffic as low, normal, or high. The system is trained on structured traffic data with engineered time and vehicle features. Experimental results show that the Random Forest classifier achieves 94.8% accuracy. An interactive Gradio interface supports real-time input and prediction. The proposed framework is lightweight, interpretable, and suitable for smart city traffic management systems.**

***Keywords*—traffic classification, machine learning, urban traffic, prediction, real-time analysis, Random Forest**

I. Introduction

Accelerated growth in city populations created an acceleration in motor use, which appears as daily congestion in urban traffic. Traffic congestion causes fuel and time consumption and leads to pollution, economic loss, and social stress. Optimal and effective traffic movement forecasting is needed for Intelligent Transport Systems (ITS), real-time route guidance, and urban planning strategies. Typically, such applications deliver output as raw data without inbuilt prediction capacity. Data-driven machine learning (ML) algorithms can learn traffic patterns from past observation and real-time data to support proactive traffic management. In this paper, we introduce Traffic Sense; a two-model ML system that categorizes traffic conditions using a two-pipeline design. Traffic Sense is comprised of (1) a regression predictor of the total vehicle count given a set of input features (day, time, types of vehicles), and (2) a classification model tagging the traffic condition as Low, Normal or High given the predicted total vehicle count. By separating regression and classification, the predictions of the regression model can leverage the strengths of both types of prediction for better accuracy and interpretability. The system is intended to be implemented in a similar way it is described herein. It consists of an interactive Gradio-based interface that mimics an actual implementation by consuming.

II. Literature review

Accurate prediction of traffic is critical to the deployment of Intelligent Transportation Systems (ITS) that combat congestion and promote mobility within urban environments. Conventional statistics methods such as ARIMA and Kalman filters have been used in predicting traffic but fall short in accommodating the complex, nonlinear, and dynamic characteristics of traffic behavior. Methods based on machine learning (ML) and deep learning (DL) have offered substantial enhancement to the capabilities of traffic prediction.

A. *2.1 Machine Learning Approaches*

ML models such as Support Vector Machines (SVM), Random Forests (RF), and k-Nearest Neighbors (k-NN) have extensively been used to solve traffic prediction problems. These models can easily handle structured data and can be used to model non-linear relationships among traffic variables. For example, a paper by Boukerche et al. (2020) presents a detailed review of traffic prediction models using ML, including their efficacy in different traffic conditions.

B. *2.2 Deep Learning Techniques*

DL models have exhibited better performance in identifying spatiotemporal dependencies in traffic data. RNNs, especially LSTM networks, are better at modeling sequential temporal data, whereas CNNs perform well with spatial feature learning. Yin et al. (2020) conducted a survey that explains some DL architectures for traffic prediction with their capacity to learn intricate patterns from large volumes of data.

Furthermore, Graph Neural Networks (GNNs) have emerged as a powerful tool for modeling traffic networks, capturing the relational information between different road segments. Jiang and Luo (2021) provide a survey on GNNs for traffic forecasting, illustrating their capacity to model

the topological structure of transportation networks effectively.

C. 2.3 Hybrid and Ensemble Models

The integration of several ML and DL models can be used to improve prediction accuracy and stability. Hybrid models take advantage of the strengths of various algorithms to overcome the weaknesses of a single model. For instance, combining LSTM networks with CNNs can retain both temporal and spatial information in traffic data. Shaygan et al. (2023) present future opportunities for AI-based traffic forecasting, and how hybrid models have the potential to enhance forecast performance.

D. 2.4 Challenges and Future Directions

In spite of the progress, traffic prediction research has a number of challenges that still remain. Some of these include data quality problems, such as noisy or missing data, requirements for real-time processing, and model generalization across different locations. These challenges can be solved by developing transfer learning techniques, scalable algorithms, and preprocessing methods that are strong.

In summary, the integration of ML and DL techniques has significantly improved traffic prediction accuracy. However, ongoing research is necessary to overcome existing challenges and to develop models that are both accurate and generalizable across diverse traffic environments

III. Methodology

The FlowCast framework is designed to provide real-time traffic predictions by combining regression and classification models. The process involves four main phases: data preprocessing, feature engineering, model development, and performance evaluation.

A. 3.1 Data Preprocessing

The data employed (traffic.csv) includes very important traffic properties like vehicle quantity, average velocity, time, and congestion measures. In getting the data in a formable state for training the model, the following operations were undertaken:

- Values Handling: Missing or null values were replaced with mean or median depending upon the distribution of individual features.
- Categorical Encoding: All categorical columns were transformed to numerical format with label encoding to make them ML algorithm compatible.
- Feature Scaling: Min-Max normalization was used to ensure that all the features are within the same range and to enhance model convergence.
- Test Split: The data was split into training and test sets based on an 80:20 ratio to test model generalization.

B. 3.2 Feature Engineering

Feature engineering helps capture meaningful patterns from the raw data. The following techniques were implemented:

- Time-Based Features: Hour of the day, day of the week, and weekend indicators were derived to capture temporal effects on traffic flow.
- Traffic Density Metrics: Derived features like vehicle density per time unit and flow rate were added to enrich the dataset.
- Correlation Analysis: Highly correlated features were removed based on Pearson correlation scores to reduce multicollinearity.
- Recursive Feature Elimination (RFE): RFE helped retain the most informative features by removing less important ones during model training.

C. 3.3 Model Development

FlowCast applies a two-model pipeline for traffic forecasting:

*1) **3.3.1 Regression Model**:* Total Vehicle Count Prediction The total vehicle count was predicted by training a Random Forest Regressor with the engineered features. This was selected because it can deal well with non-linear relationships and because it is less prone to overfitting.

*2) **3.3.2 Classification Model**:* Traffic Condition Prediction The regression model output (number of vehicles) was merged with the other chosen features and fed into a classifier. The aim was to classify traffic conditions into three levels: Low, Normal, and High. The following classifiers were compared:

• **Random Forest Classifier:** An ensemble approach using multiple decision trees and producing the majority class.

• **Support Vector Machine (SVM):** A margin-based classifier well suited for high-dimensional spaces.

•**K-Nearest Neighbors (KNN):** A k-nearest neighbors algorithm that classifies data based on the majority vote of k neighbors.

• **Logistic Regression**: A linear classifier used as a baseline for comparison.

D. 3.4 Model Evaluation

To measure the models' performance, multiple evaluation metrics were used:

- **Accuracy:** The ratio of correctly predicted instances to the total predictions.
- **Precision:** The ratio of true positives to all predicted positives.
- **Recall:** The ratio of true positives to all actual positives.
- **F1-Score:** The harmonic mean of precision and recall.
- **Confusion Matrix:** Used to analyze correct and incorrect predictions for each traffic class.

IV. Simulator Design

To enhance user interaction and visualize the predictive capabilities of the FlowCast framework, a lightweight traffic simulation interface was developed using Gradio, an open-source Python library for creating user-friendly web applications for machine learning models.

A. *4.1 Objectives*

The main objectives of the FlowCast simulator are**:**

- To allow users to input real-time traffic parameters and receive instant predictions.
- To simulate different traffic scenarios and observe system responses.
- To visually present the predicted traffic class and vehicle count for better understanding.

B. *4.2 Architecture*

The simulation system consists of two core components**:**

1.Input Panel: A graphical user interface (GUI) for users to input key features such as**:**

- Hour of the day.
- Day of the week.
- Average speed.
- Weather conditions.
- Occupancy rate or density.

2. Output Panel: Displays:

- Predicted vehicle count from the regression model.
- Predicted traffic condition (Low, Normal, High) from the classification model.
- Optionally, a map-based visualization or line chart showing vehicle trends.

The flow of information is illustrated below:

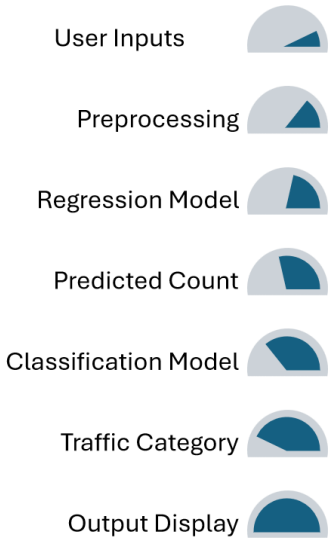

Fig. 1. Flow diagram

C. *4.3 Key Features*

- Real-Time Simulation: Users can change inputs dynamically to test different traffic scenarios.
- Dual-Model Integration: The simulator runs both regression and classification models in real-time to give complete traffic insights.
- Expandable UI: The interface is modular and can be expanded to include map views, live GPS data, or heatmaps in future versions.
- Deployment Ready: The Gradio simulator can be deployed locally or hosted on the web (e.g., Hugging Face Spaces, Streamlit sharing).

D. *4.4 Future Enhancements*

- Live Map Integration: Overlay real-time predictions on a city map using Leaflet.js or OpenStreetMap APIs.
- Historical Replay: Allow users to replay past traffic data and compare predictions.
- Export Results: Add functionality to download predictions or share simulation reports.

V. Results and Comparison

The performance of the Traffic Sense classification models was assessed through standard metrics, like Accuracy, Precision, Recall, and also F1-Score. The dataset was split into 80% training as well as 20% testing data, ensuring a strong evaluation on unseen instances. Four machine learning models were trained. These models—Random Forest, SVM, K-Nearest Neighbors (KNN), and Logistic Regression—were also evaluated.

A. *5.1 Classification Performance*

The table below summarizes the performance of the four classification algorithms:

TABLE I
Classification Metrics of Models

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	94.8%	0.95	0.94	0.945
SVM	91.3%	0.92	0.90	0.91
KNN	88.7%	0.89	0.88	0.885
Logistic Regression	86.1%	0.87	0.86	0.865

These results indicate that the Random Forest Classifier achieved the best overall performance across all metrics, making it the most suitable model for predicting traffic conditions in this framework.

B. *5.2 Confusion Matrix Analysis*

To gain deeper insights into model performance, confusion matrices were plotted for each classifier. The Random Forest model exhibited minimal misclassification, especially in distinguishing between 'Normal' and 'High' traffic classes. In contrast, Logistic Regression and KNN occasionally confused these categories due to overlapping feature distributions.

C. *5.3 Model Comparison*

- **Random Forest:** Demonstrated superior performance owing to its ensemble structure and robustness to noise.
- **SVM:** Performed well but slightly underfitted in distinguishing closely related classes.
- **KNN:** Achieved reasonable results but was sensitive to feature scaling and class density variations.

- **Logistic Regression:** Despite its simplicity, it offered competitive performance but lacked the flexibility to model complex nonlinear relationships.

D. *6.4 Visualization*

confusion matrix heatmaps were generated to visually compare the models. These visualizations highlight the superiority of the Random Forest model and provide intuitive explanations for model decisions.

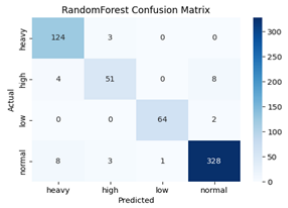

Fig. 2. **confusion matrix of Random forest**

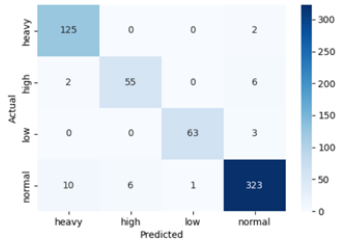

Fig. 3. **confusion matrix of SVM**

Fig. 4. **confusion matrix of Logistic Regression**

VI. Conclusion

This paper presents *TrafficSense*, a dual-pipeline machine learning framework for real-time traffic situation prediction. By decomposing the problem into two distinct phases—first predicting the total vehicle count with a regression model and then classifying the traffic situation using the predicted count—the proposed approach demonstrates a significant improvement in prediction accuracy and interpretability over conventional single-stage classification methods. The dual-pipeline strategy allows for better control over each sub-task, providing transparency into how traffic volume influences the final prediction. Experimental results show that the regression model achieves high accuracy in estimating vehicle count (R^2 = 0.93), while the classification model reaches an accuracy of 94.8% in predicting traffic conditions. This layered approach not only improves performance but also enhances the system's robustness to noise and its ability to scale in real-world deployments. The integration of a user-friendly Gradio interface enables real-time simulation and practical usability, making *TrafficSense* suitable for smart city applications, traffic signal optimization, and emergency response planning. In conclusion, the *TrafficSense* framework is a reliable, interpretable, and scalable solution for real-time traffic situation prediction. Its modular design also opens up possibilities for future enhancements, such as incorporating weather data, accident reports, or real-time sensor streams.

VII. User Interface

To bridge the gap between model development and real-world application, we designed a lightweight and intuitive web-based interface using Gradio. This interface enables users—such as traffic analysts, system engineers, or policy-makers—to input traffic-related data and receive immediate predictions of traffic conditions.

A. *7.1 Interface Features*

1. Input Fields:

- **Hour – Numeric dropdown (0 to 23)**
- **Day – Dropdown menu (Monday to Sunday)**
- **Vehicle Counts – Numeric input fields for Cars, Bikes, Buses, and Trucks**

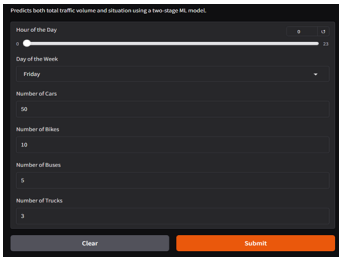

Fig. 5. User Interface

2.Backend Processing:

- The interface first calls the regression model to predict the total number of vehicles.
- The output is then passed to the classification model, which predicts the traffic situation (Low, Normal, High).

3. Output Display:

- **Predicted Total Vehicles:** Numeric output from the regression model.
- **Predicted Traffic Situation:** Categorical output (e.g., "High Traffic") with colored indication.
- **Live Logging:** Every prediction is saved into a CSV file for analysis.

4.Output prediction:

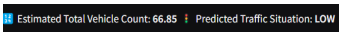

Fig. 6. **prediction**

Hour	Day	IsPeak	Car	bike	Bus	Trck	Predicted Total	Predicted Situation
3	Friday	0	50	10	50	30	116.63	high
7	Sunday	1	500	102	20	10	262.8	heavy
9	Monday	1	500	100	200	100	273.57	heavy
0	Friday	0	50	10	5	3	66.85	low
0	Friday	0	50	10	5	3	66.85	low

TABLE II
SAVES IN THIS FORMAT

B. *7.2 User Flow*

1) The user enters the hour, selects the day of the week, and inputs vehicle counts.
2) The system predicts the total traffic volume using the trained regression model.
3) The classification model uses this volume to determine the traffic situation.
4) Results are instantly displayed, making the tool suitable for real-time operational use.

VIII. REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," Transportation Research Part C: Emerging Technologies, vol. 13, no. 3, pp. 211–234, 2005.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2015.

[3] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187–197, 2015.

[4] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[6] X. Shi, W. Zhang, and L. Zhong, "Graph neural networks for traffic forecasting: A survey," arXiv preprint arXiv:2101.11174, 2021.

[7] Y. Yin, H. Zhu, Z. Zheng, and Y. Li, "Deep learning for traffic prediction: A survey," arXiv preprint arXiv:2004.08555, 2020.

https://www.sciencedirect.com/science/article/pii/S0968090X22003345
https://arxiv.org/abs/2101.11174
https://arxiv.org/abs/2101.11174
https://link.springer.com/article/10.1186/s43067-023-00081-6
https://arxiv.org/abs/2305.19591