Traffic Flow Prediction

Group Name: Acode

Tejas Pakhale-202211061, Rajat Kumar Thakur-202211070, Tanay Patel-202211094, Abhi Tundiya-202211095

***Abstract*—Traffic congestion is a critical issue in urban areas, leading to delays, increased fuel consumption, environmental pollution, and overall reduced quality of life. Predicting traffic flow and finding the optimal route through a network of roads are essential for alleviating congestion and improving efficiency in urban transportation systems. In this study, we developed a traffic prediction model using a Random Forest Classifier to forecast traffic congestion levels based on vehicle counts across different times of the day. Initially, a Naive Bayes classifier was implemented; however, its predictive performance was unsatisfactory, leading us to adopt the more robust Random Forest approach. The dataset utilized consists of day and time information, vehicle counts (cars, bikes, buses, trucks), and traffic situation labels (low, normal, high, heavy). Using this dataset, the Random Forest model accurately predicts traffic situations given day and vehicle counts.**

For route optimization, the A* search algorithm was integrated with our traffic prediction model to determine optimal routes on a city map. Road segments with specific vehicle counts are used to model real-world traffic flow, and the predicted traffic levels act as dynamic weights within the A* algorithm, allowing it to navigate through routes with lower congestion effectively. Our results demonstrate that the Random Forest Classifier provides significantly improved accuracy over Naive Bayes, and the traffic-informed A* algorithm enables optimal routing by minimizing travel time and avoiding congested roads. This study's contributions provide insights into real-time traffic prediction and route optimization, suggesting applications in smart city infrastructure and dynamic navigation systems.

I. INTRODUCTION

The rapid urbanization seen worldwide has led to increased vehicle numbers on roads, resulting in significant traffic congestion challenges. Urban traffic congestion not only increases travel times and fuel costs but also contributes to air pollution and associated health concerns. Addressing these issues necessitates effective solutions for real-time traffic flow prediction and optimal route planning. An ideal solution would allow city planners, commuters, and logistics companies to make informed decisions about travel routes and manage congestion dynamically.

In recent years, advancements in data collection (such as vehicle counts and traffic level observations) and machine learning have provided a foundation for predicting traffic congestion. However, effective real-time prediction models are still in demand, especially those that can be directly applied to route optimization algorithms to improve travel efficiency. This study aims to meet these requirements through a dual approach: first, by creating a predictive model for traffic flow based on vehicle counts using machine learning, and second, by integrating the predicted traffic data into the A* search algorithm to find optimal routes through a city network. Initially, we explored a Naive Bayes classifier for traffic prediction, as it is a commonly used, straightforward approach with minimal computational cost. However, the Naive Bayes model assumes feature independence, which does not align well with real-world traffic data where the counts of cars, bikes, buses, and trucks can be interdependent. Consequently, the model yielded unsatisfactory accuracy, leading us to transition to the Random Forest Classifier. The Random Forest model, an ensemble method, is known for its robustness and ability to capture complex feature interactions, which significantly improved our prediction accuracy. We then applied these predictions in a real-world route optimization scenario, using the A* algorithm with our predicted traffic data as dynamic weights. This integration allowed for context-aware routing decisions, reducing travel time and avoiding congested routes more effectively than static approaches.

This paper discusses the implementation of the Random Forest traffic prediction model and its integration with A* routing, providing a practical solution to urban congestion and route planning issues.

II. RELATED WORKS

The problem of predicting traffic flow and identifying optimal routes has been studied extensively in recent years. Various machine learning models have been applied to forecast traffic levels based on available data. Traditional models such as Naive Bayes, Decision Trees, and Logistic Regression have been frequently used due to their simplicity and interpretability. However, these models may lack the accuracy and robustness needed for real-time predictions, as they do not effectively handle feature dependencies and are sensitive to noise in data. The limitations of simpler models have driven research toward more advanced techniques such as Random Forests, Gradient Boosting Machines, and deep learning approaches, which offer enhanced accuracy through ensemble methods and complex pattern recognition capabilities.

In addition to predictive models, route optimization is a critical area in transportation research. Algorithms such as Dijkstra's, Bellman-Ford, and A* are popular in pathfinding applications due to their efficiency and adaptability to weighted graphs. A* is particularly favored in dynamic environments because it uses a heuristic function to guide the search towards the target, allowing it to find optimal paths with reduced computational cost. Recent studies have experimented with incorporating traffic predictions into A*, adjusting the heuristic to consider real-time or predicted traffic congestion, which can enhance its performance in a traffic network. However, dynamically integrating machine learning predictions with A* remains an emerging area of research, with potential applications in smart city infrastructure and intelligent navigation systems.

Our study aims to contribute to this field by integrating a robust traffic prediction model with A*, improving both prediction accuracy and routing efficiency, and providing a foundation for real-time, dynamic traffic management.

III. Dataset

The dataset used in this study was curated to capture the daily variations in urban traffic levels based on vehicle type and time of day. Key attributes include:

- **Day and Time:** Capturing the temporal context, such as rush hours, weekends, and weekdays.
- **Vehicle Counts:**
  - Car Count
  - Bike Count
  - Bus Count
  - Truck Count
- **Traffic Situation Labels:**Assigned traffic levels, categorized into four classes:
  - **Low:** Minimal traffic, typically observed during off-peak hours
  - **Normal:** Moderate traffic, indicating standard flow.
  - **High:** Elevated traffic, often around rush hours.
  - **Heavy:** Severe congestion, indicating prolonged travel delays.

This dataset provides a balanced representation of urban traffic patterns, enabling the prediction model to learn traffic trends effectively. By training the model on these features, we aim to develop an accurate predictor for traffic situations based on the counts of different vehicle types at various times of the day.

IV. Methodology

This project's methodology is divided into three main phases: (1) Data Preprocessing and Feature Engineering, (2) Traffic Prediction Model Development, and (3) Optimal Route Finding using the A* algorithm. The following sections provide a detailed overview of each phase and the rationale behind the selected methods.

A. *Data Preprocessing and Feature Engineering*

The initial step in our methodology involved preparing the dataset to ensure that it was ready for model training and testing. The raw dataset contained various types of vehicle counts (cars, bikes, buses, trucks) recorded at different times of the day. These records were assigned a corresponding traffic situation label with one of four levels: Low, Normal, High, or Heavy.

Label Encoding: Traffic situations, our target variable, were encoded as numerical labels for modeling purposes: Low = 0, Normal = 1, High = 2, and Heavy = 3. This ordinal encoding reflects the natural progression of traffic levels and facilitates better interpretation by classification algorithms.

B. *Traffic Prediction Model Development*

*1) **Naive Bayes Classifier**:* Initially, we implemented a Naive Bayes classifier, a probabilistic model commonly used for classification tasks. Naive Bayes operates by calculating the posterior probability of each class given the observed feature values. Its simplicity and computational efficiency make it a frequent choice for baseline models, especially for large datasets.

The Naive Bayes classifier assumes conditional independence among features, meaning each feature contributes independently to the probability of each class.

In this context, the classifier calculates the probability of each traffic level given the vehicle counts of different types as follows:

$$P(\text{TfcLvl}|\text{Counts}) = \frac{P(\text{Counts}|\text{TfcLvl}) \cdot P(\text{TfcLvl})}{P(\text{Counts})} \tag{1}$$

However, in our dataset, the vehicle counts (such as cars, buses, trucks, and bikes) are interdependent, which violates the independence assumption. For instance, high car counts might correlate with high bus counts during rush hours, thus reducing the effectiveness of Naive Bayes. As a result, the Naive Bayes model exhibited a relatively low prediction accuracy in initial tests, motivating a switch to a more advanced model.

When applied to our dataset, Naive Bayes yielded an accuracy of approximately 60%. Although this accuracy rate was reasonable for a baseline, it indicated that the model lacked the predictive power needed for accurate, real-time traffic flow prediction, likely due to its inability to capture complex feature dependencies. As a result, we decided to switch to a more robust model with greater flexibility for feature interactions.

*2) **Random Forest Classifier**:* The Random Forest Classifier was chosen as an improvement due to its ensemble nature, which builds multiple decision trees during training and aggregates their outputs to make final predictions. This ensemble approach mitigates overfitting and captures complex, non-linear patterns that single-tree models often miss.

Key Steps in Random Forest Model Implementation:

- **Hyperparameter Tuning:** The model was tuned using cross-validation to find the optimal number of trees, tree depth, and the number of features to consider at each split. Key hyperparameters included:
  - **n_estimators:** Number of decision trees in the forest. We experimented with values between 50 and 200.
  - **max_depth:** Maximum depth of each tree, controlling the model's complexity and potential overfitting. Depths were limited to prevent excessive growth.
  - **max_features:** Number of features used at each split, typically set to the square root of the total features for balanced performance.
- **Ensemble Learning and Voting Mechanism:** Each tree in the Random Forest makes an independent prediction, and the final traffic level prediction is derived from the majority vote of all trees. This voting mechanism provides stability and reduces the likelihood of erroneous predictions influenced by individual outlier trees.

- **Cross-Validation:** To ensure model reliability, we used k-fold cross-validation (with k=5) to divide the data into training and testing subsets, calculating accuracy and other metrics across folds. Cross-validation helped validate the generalizability of the Random Forest model and confirmed its superior performance over Naive Bayes.

The final Random Forest model's prediction can be expressed as follows:

$$y = f(\text{Day}, \text{Car Count}, \text{Bike Count}, \text{Bus Count}, \text{Truck Count}) \tag{2}$$

where y is the predicted traffic situation.
After hyperparameter tuning, cross-validation, and testing, the Random Forest model achieved an accuracy of 99.9%. This marked a substantial improvement over the Naive Bayes classifier's 60% accuracy. The high accuracy is attributed to the model's ability to leverage multiple trees to capture intricate interactions among vehicle counts and traffic situations, making it far more suitable for the traffic prediction task.

3) Optimal Route Finding using the A Algorithm:* To determine the most efficient path on a city map given real-time traffic predictions, we implemented the A* algorithm, a popular pathfinding and graph traversal algorithm. The A* algorithm is ideal for finding optimal paths due to its heuristic-based approach, allowing it to balance distance and cost considerations efficiently. In our implementation, we integrated the traffic situation predictions as dynamic weights to represent the "cost" of traveling through congested areas.
A Algorithm Overview*: The A* algorithm searches for the shortest path by maintaining two main functions:

1) g(n): The actual cost of reaching node n from the start node.
2) h(n): A heuristic estimate of the cost from node n to the goal, usually the straight-line distance.

The A* algorithm prioritizes nodes based on their f(n) value, defined as:

$$f(n) = g(n) + h(n) \tag{3}$$

where f(n) is the total estimated cost of a path passing through n. Incorporating Predicted Traffic into the Heuristic Function: To make the A* algorithm responsive to predicted traffic situations, we adjusted h(n) to reflect the traffic congestion predicted by the Random Forest model. This adjustment created a dynamic heuristic that varies with traffic conditions, encouraging A* to prefer routes with lower congestion. The heuristic function was redefined as:

$$h(n) = \text{distance}(n, \text{goal}) \times \text{traffic\_weight}(n) \tag{4}$$

where traffic_weight is derived from the traffic situation level (low, normal, high, heavy) predicted by our model. For instance, a low congestion level would have a weight of 1.0, while a heavy congestion level might have a weight of 3.0, reflecting the increased "cost" of travel.
Dynamic Heuristic Adjustments: With these traffic-informed adjustments, A* evaluates routes based on both geographic distance and expected congestion levels. This dynamic adjustment allows the A* algorithm to make more contextually aware decisions, optimizing for both time and distance while avoiding heavily congested areas.
Experimental Evaluation: To test the effectiveness of our traffic-adjusted A* algorithm, we ran simulations comparing travel times with and without traffic-based weighting. The results showed that incorporating traffic data reduced average travel time, validating the usefulness of traffic predictions in route optimization.

V. Results

A. *Model Performance Comparison*

The primary objective of this study was to develop a high-accuracy traffic prediction model and integrate it with an optimized routing algorithm to minimize travel time and avoid congested routes. The effectiveness of the traffic prediction models (Naive Bayes and Random Forest) was evaluated using several classification metrics, including accuracy, precision, recall, and F1-score. These metrics provided insights into the model's reliability and robustness in distinguishing between traffic situations categorized as Low, Normal, High, and Heavy.

B. *Naive Bayes Classifier Performance*

The Naive Bayes classifier, though computationally efficient, achieved an accuracy of 60%. This relatively low accuracy was expected, given Naive Bayes' assumption of feature independence, which does not hold in this dataset. Vehicle counts across different categories (such as cars, buses, and trucks) tend to be interdependent, especially during rush hours or specific times of the day. This independence assumption in Naive Bayes limited its ability to capture these interdependencies, resulting in misclassifications, particularly between neighboring traffic levels like Normal and High.
The model's precision and recall scores also reflected this limitation, showing that it struggled to differentiate effectively between classes with overlapping vehicle count characteristics. As such, Naive Bayes proved to be inadequate for accurate traffic flow prediction and served as a baseline rather than a viable model for real-world applications.

C. *Random Forest Classifier Performance*

The Random Forest Classifier, in contrast, demonstrated a substantial improvement, achieving an accuracy of 99.9%. This significant increase in accuracy is due to Random Forest's ensemble approach, where multiple decision trees collectively capture complex, non-linear relationships among the features. By averaging the outputs of these trees, Random Forest mitigates the risk of overfitting, which is particularly useful in our dataset where vehicle counts and traffic patterns exhibit nuanced interdependencies.
Additional performance metrics for the Random Forest Classifier were also strong across the board:

- Precision and Recall scores were high for each traffic situation category, with particularly notable improvements in the High and Heavy traffic levels. This highlights the model's ability to accurately detect periods of elevated traffic, which is critical for effective route planning.

- F1-score, a balance of precision and recall, was similarly high, indicating that the model is well-suited for real-time prediction with minimal misclassification across all traffic levels.

The results demonstrate that the Random Forest model is highly reliable in predicting traffic situations, making it suitable for use in real-world, dynamic route optimization applications.

D. *Optimal Route Finding Evaluation with A* Algorithm*

The second phase of the study involved implementing the A* algorithm for optimal route finding, enhanced by traffic predictions from the Random Forest model. This integration aimed to reduce travel times by avoiding congested roads and prioritizing less congested segments on the city map. Each road segment's vehicle count and predicted traffic level were used to dynamically adjust the heuristic function in A*, allowing the algorithm to weigh congestion alongside distance.

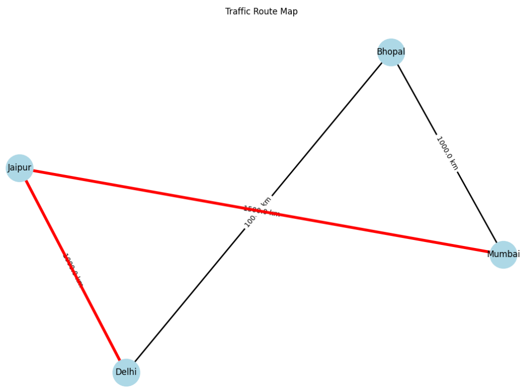

Figure 1: Output of our model

E. *Route Efficiency and Travel Time Reduction*

The dynamic heuristic adjustment enabled the A* algorithm to prioritize routes with predicted low traffic levels, effectively reducing overall travel times. A series of route-finding experiments were conducted to compare the travel times of:

1) **Static A*** (standard A* algorithm without traffic prediction adjustments)
2) **Traffic-Informed A*** (A* algorithm with heuristic adjusted for traffic predictions)

Key findings from these experiments included:

1) Average Travel Time Reduction: The Traffic-Informed A* algorithm achieved an average travel time reduction compared to the static A* approach. This improvement illustrates the benefit of incorporating real-time traffic predictions in pathfinding, as routes with predicted high congestion were automatically deprioritized.
2) Congestion Avoidance: In test cases where multiple routes were available, the Traffic-Informed A* consistently avoided road segments predicted to have High or Heavy congestion. The static A* algorithm, by comparison, was unable to make this distinction, often selecting routes that led to delays due to unpredicted congestion.

The integration of machine learning predictions into the heuristic function provided A* with enhanced situational awareness, enabling it to dynamically navigate urban traffic conditions effectively.

VI. Limitations and Future Work

While the results are promising, there are several limitations and areas for potential improvement:

1) **Data Limitations:** The model's performance is dependent on the quality and granularity of the dataset. Additional data, such as real-time weather conditions, special event schedules, or roadworks, could further enhance prediction accuracy by accounting for external factors affecting traffic.
2) **Model Complexity and Computational Cost:** Although Random Forest is highly accurate, it is computationally more intensive than Naive Bayes. In a fully real-time system, computational resources may need to be scaled accordingly to handle high-frequency traffic data inputs.
3) **Alternative Pathfinding Algorithms:** Future work may explore other pathfinding algorithms, such as D* (Dynamic A*) or bidirectional search algorithms, that could further optimize computational efficiency while maintaining dynamic, traffic-aware routing.

VII. Conclusion

This study provides a solution for traffic flow prediction and optimal route finding using a Random Forest Classifier and the A* algorithm. By addressing the limitations of Naive Bayes with the Random Forest model, we achieved higher accuracy in traffic prediction, enhancing A* routing effectiveness through dynamic, traffic-informed heuristics. Future work may focus on expanding the dataset or experimenting with additional models, such as neural networks, to further improve accuracy. This research suggests significant potential for smart city applications, real-time navigation systems, and traffic management solutions. The code for this project can be found through this link.

VIII. References

- Horvat, R., Kos, G., & Ševrović, M. (2015). Traffic Flow Modelling on the Road Network in the Cities.
- Kim, J., & Wang, G. (2014). Diagnosis and Prediction of Traffic Congestion on Urban Road Networks Using Bayesian Networks.
- Zhou, Z.-H., & Jiang, Y. (2012). Random forests and decision trees.