Report Content

Cover Page

Mumbai Traffic Congestion Prediction Using Artificial Intelligence

Student Name: [Your Name]
Roll No.: [Your Roll Number]
Guide: [Guide Name]
Department of Computer Science and Engineering
[Institute Name]
[Month Year]

Certificate

This is to certify that [Your Name], Roll Number [Your Roll Number], has satisfactorily completed the project entitled
“Mumbai Traffic Congestion Prediction Using Artificial Intelligence” under my supervision, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering at [Institute Name].

Date: _______________
Signature of Guide: _______________
Name of Guide: _______________

Acknowledgment

I would like to express my heartfelt thanks and sincere gratitude to my project guide, [Guide Name], for their invaluable guidance, encouragement, and support throughout this project. Their insightful suggestions and constant motivation were instrumental in successfully completing this work.

I am also grateful to the faculty members and staff of the Department of Computer Science and Engineering for providing the necessary resources and a conducive environment for research and learning.

Finally, I extend my appreciation to my family and friends for their unwavering support and patience during the course of this project.

Abstract

Mumbai, a city home to over 20 million inhabitants, faces enormous transportation challenges due to rapid urbanization and increased vehicular demand. This growth has led to frequent traffic congestion causing lost productivity, increased fuel consumption, air pollution, and diminished quality of life. Traditional traffic management approaches are reactive and often unable to prevent congestion beforehand.

This project employs advanced Artificial Intelligence (AI) and Machine Learning (ML) methods to predict traffic congestion levels across Mumbai using a diverse, multi-source dataset. The dataset integrates primary survey data capturing commuter demographics, travel patterns, and subjective congestion experiences, along with secondary data such as weather conditions, accident reports, and road infrastructure status to enrich model inputs.

To overcome the limitations of a relatively small primary dataset, synthetic data augmentation techniques were implemented by injecting controlled perturbations, improving model stability and generalization.

Three machine learning algorithms—Random Forest, CatBoost, and XGBoost—were trained and rigorously validated using metrics such as accuracy, precision, recall, and F1-score. Additional performance was achieved via stacking ensemble methods, which combine the strengths of individual base models.

The results demonstrate the promise of AI-based forecasting frameworks as vital tools for intelligent transportation systems, enabling dynamic route optimization and proactive urban traffic management. Moreover, feature importance analysis identified key traffic influencers such as peak travel times and adverse weather conditions, which can aid urban planners in targeted policy formulation.

This work lays a foundation for integrating real-time data sources and scalable AI solutions in metropolitan traffic control, ultimately facilitating smarter, greener, and more efficient urban mobility. The project also highlights opportunities for future enhancements including deployment in cloud platforms, incorporation of deep learning models, and the development of commuter-centric responsive applications.

Table of Contents

- To be created in MS Word after inserting all sections *

List of Tables

List of Figures

Abbreviations / Nomenclature

Abbreviation	**Full Form**
AI	Artificial Intelligence
ML	Machine Learning
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Oversampling Technique

Chapter 1: Introduction

Mumbai is India's commercial and financial epicenter, boasting a population exceeding 20 million residents. This megacity witnesses continuous growth in its residential density and vehicle ownership, engendering profound challenges in urban transportation management. With an intricate mosaic of private vehicles, public transport modes, pedestrians, and commercial traffic sharing limited road space, Mumbai frequently experiences severe traffic congestion, particularly during morning and evening peak hours.

The resultant traffic snarls contribute to extended travel durations, elevated fuel consumption, and increased vehicular emissions, all of which deteriorate air quality and contribute to climate change. In addition, passenger stress and reduced productivity detract from residents' well-being and economic contribution.

Several systemic factors amplify Mumbai's traffic woes:

- Limited and aged infrastructure: Roads designed for lower vehicle volumes struggle with modern traffic demands.
- Mixed traffic patterns: Interactions between private vehicles, buses, local trains, and pedestrians create complex flows.
- Inefficient traffic signal coordination: Despite efforts, signal timings lack adaptive real-time capabilities.

The city's traffic control agencies primarily rely on reactive strategies—responding post hoc to congestion through infrastructure expansion, signal adjustments, and occasional traffic restrictions. While these measures alleviate symptoms temporarily, they do not address the root cause or anticipate real-time fluctuations arising from weather disturbances, accidents, or maintenance activities.

Problem Statement

Mumbai's traffic congestion is a complex urban challenge marked by overwhelming vehicle population and limited infrastructure. Mixed traffic of private cars, buses, trains, and pedestrians interact in space-constrained roadways causing daily gridlocks. Current traffic management techniques focus on reactive measures and lack predictive model integration. This not only leads to inefficient congestion mitigation but also heightens environmental pollution, fuel wastage, and commuter stress. Therefore, there is a critical need for developing accurate, data-driven traffic congestion prediction models that can proactively aid urban planners and commuters in making informed decisions, ultimately enhancing the city’s mobility and quality of life.

Project Objectives

- To collect and synthesize detailed multi-source data including commuter surveys, weather conditions, accident reports, and synthetic data augmentation.
- To preprocess and engineer features that accurately represent traffic dynamics.
- To develop and fine-tune machine learning models (Random Forest, CatBoost, XGBoost) and ensemble methods for traffic congestion prediction.
- To evaluate models rigorously with multiple metrics and provide actionable insights for traffic authorities.

This project embarks on deploying Artificial Intelligence to predict Mumbai’s traffic conditions using comprehensive datasets, aiming to transform traffic management from a reactive to a foresighted discipline.

Chapter 2: Literature Review

Artificial Intelligence (AI) and Machine Learning (ML) techniques have revolutionized urban traffic prediction globally. Traditional forecasting methods based on historical averages and statistical models often fail to capture complex temporal and spatial dependencies in metropolitan traffic flows.

Recent research emphasizes the efficacy of ensemble learning methods such as Random Forests and gradient boosting for traffic congestion prediction due to their robustness and interpretability. CatBoost and XGBoost classifiers have gained popularity for handling heterogeneous and categorical urban datasets effectively.

Emerging deep learning models, including Long Short-Term Memory (LSTM) networks and Graph Neural Networks (GNNs), have shown superior performance by modeling sequential data and spatial correlations in traffic networks. However, these approaches require large volumes of high-quality real-time data, which is often lacking in developing urban regions like Mumbai.

In the Indian context, data limitations and heterogeneous traffic conditions pose challenges to direct adoption of global models. Hybrid approaches integrating primary survey data with secondary environmental datasets are less explored but critical for better model relevance.

Additional points include:

- Transfer learning techniques adapting pre-trained models to data-scarce cities offer promising prospects.
- Reinforcement learning for adaptive traffic signal control is gaining traction.
- Integrating crowd-sourced data such as GPS traces enhances prediction freshness and accuracy.
- Ethical and privacy considerations in urban data collection and AI use are crucial for sustainable adoption.
- Emphasis on interpretable AI models facilitates trust and city planners' acceptance.
- Multimodal traffic modeling combining road, rail, and pedestrian flows is essential yet underdeveloped.

This project addresses the gaps by using multi-source and synthetic data, combined with ensemble ML methods, contributing to advancing AI-driven traffic management tailored for Indian megacities.

Chapter 3: Methodology

This chapter delineates the comprehensive methodology implemented for the Mumbai Traffic Congestion Prediction project, encompassing data collection, preprocessing, feature engineering, and supervised machine learning model development.

Data Collection with Examples

A robust multi-source data acquisition strategy was employed to comprehensively capture traffic dynamics:

- **Primary Data: Commuter Survey**
  A structured survey was administered to 57 commuters across various demographics including age, gender, and travel behavior. The survey gathered information on daily travel duration, preferred modes of transport, frequency of travel, and perceived congestion severity.

Age Group	Gender	Travel Frequency	Avg Daily Traffic Time	Mode of Transport	Congestion Severity Rating
18-25	**Male**	**Daily**	**< 30 mins**	**Local Train**	**High**
26-35	**Female**	**Weekly**	**45 mins**	**Bus**	**Moderate**
36-45	**Male**	**Monthly**	**> 60 mins**	**Private Vehicle**	**Low**

This survey data formed the essential understanding of commuter experiences and behaviors affecting congestion patterns, providing a human-centric perspective.

- **Secondary Data: Environmental and Traffic Reports**
  Complementing primary data, secondary datasets were retrieved from governmental and meteorological sources encompassing weather metrics (rainfall, temperature, fog presence), detailed accident logs with locations and timings, and municipal information on road maintenance and traffic enforcement.

Example snapshot of weather data:

Date	Rainfall (mm)	Temperature (°C)	Fog (Yes/No)
2025-08-01	**15**	**30**	**Yes**
2025-08-02	**0**	**33**	**No**

Date	Rainfall (mm)	Temperature (°C)	Fog (Yes/No)
2025-08-03	**5**	**29**	**No**

Including such data captures fluctuations in traffic flow influenced by environmental conditions and external disruptions, enabling the model to factor in real-world complexities.

- ### Synthetic Data Augmentation
  Due to the limited size of primary survey data, synthetic augmentation was crucial to enhance model training. Using statistical modeling, Gaussian noise was added to continuous features like average traffic time and weather metrics, creating realistic variability while preserving inter-feature relationships.

Example:

Original point: Avg Traffic Time = 45 minutes
Augmented point: Avg Traffic Time = 47 minutes (noise added)

This approach helps reduce model overfitting and improves generalization across diverse plausible traffic scenarios.

Data Preprocessing

Preprocessing ensured raw data was transformed into a clean, balanced, and machine-readable format:

- **Data Cleaning:** Privacy-sensitive fields were removed. Missing values were imputed with median or mode values depending on variable type. Duplicate records were identified and removed. Outliers beyond acceptable thresholds were carefully evaluated and either corrected or excluded.
- **Categorical Encoding:** Categorical variables such as gender and mode of transport were label-encoded or one-hot encoded as appropriate. For example, gender was binary-encoded as Male=0 and Female=1, while modes of transport were expanded into separate boolean columns.
- **Feature Engineering:** Traffic data was segmented temporally into rush hours and off-peak periods. Weather features were converted into binary indicators (e.g., Rain=True/False). Historical traffic averages informed trend-based features. Interaction terms (e.g., rain during peak time) were also formulated.
- **Balancing Dataset:** Traffic congestion labels were imbalanced, with fewer high congestion instances. SMOTE oversampling technique generated synthetic minority class examples to achieve balanced label distribution. Additionally, class weights were tuned during model training to counter bias.

Chapter 4: Implementation / Data Analysis

This chapter details the machine learning model development and exploratory data analysis (EDA) carried out for Mumbai Traffic Congestion Prediction.

Model Selection

Selecting algorithms suitable for heterogeneous urban traffic data with mixed categorical and numerical features was critical.

- **Random Forest:** Utilizes multiple decision trees with bootstrapped datasets combined by majority voting. Random Forest is robust to noise and well suited for tabular data. It also provides feature importance metrics useful in understanding predictor influence.
- **CatBoost:** Specialized gradient boosting approach that handles categorical features natively and mitigates target leakage through ordered boosting. Well suited to traffic data involving categorical survey responses.
- **XGBoost:** Optimized gradient boosting framework implementing regularization to reduce overfitting, offering excellent speed and efficiency with high accuracy on structured data.

Training and Hyperparameter Tuning

The dataset was split into 70% training and 30% testing subsets. Hyperparameters such as tree depth, number of estimators, and learning rate were fine-tuned using grid search and cross-validation.

Sample training code snippet (Random Forest):

python

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100, max_depth=8, random_state=42, class_weight='balanced')

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)
```

Exploratory Data Analysis (EDA)

EDA was instrumental in revealing data characteristics and feature relationships:

- Distribution plots of the “Average Daily Traffic Time” showed a peak between 30-60 minutes, representing rush hour congestion.
- Box plots for “Travel Frequency” revealed that daily commuters experienced higher congestion levels than occasional travelers.

- Correlation matrix heatmaps showed positive correlation (>0.6) between "Rainfall Presence" and congestion severity, illustrating weather impact.

Feature Importance Visualization

Random Forest and XGBoost models provided feature importance scores. The top contributors to congestion prediction included:

- Average Daily Traffic Time: highest importance (~30%)
- Weather Conditions: rainfall and fog indicators (~20%)
- Mode of Transport: e.g., local train, bus, or private vehicle (~15%)
- Travel Frequency: daily vs weekly commuters (~10%)

These insights affirm domain expectations and guide feature engineering priorities.

Model Performance and Analysis

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	**62%**	**60%**	**65%**	**62%**
CatBoost	**71%**	**70%**	**68%**	**69%**
XGBoost	**85%**	**83%**	**81%**	**82%**

XGBoost delivered superior predictive accuracy, with marked improvements over traditional Random Forests. CatBoost also performed admirably, particularly benefiting from native categorical encoding.

Confusion Matrix Example: XGBoost

Actual \ Predicted	Low Traffic	Moderate Traffic	High Traffic
Low Traffic	**14**	**3**	**0**
Moderate Traffic	**4**	**12**	**5**
High Traffic	**1**	**2**	**10**

The model showed occasional difficulty distinguishing adjacent classes (Moderate vs High congestion), a known challenge in nuanced urban traffic states.

Stacking Ensemble Model

To leverage the strengths of individual models, a stacking ensemble was implemented combining Random Forest and CatBoost base learners, with Logistic Regression as the meta-classifier.

Sample stacking code snippet:

python

```
from sklearn.ensemble import StackingClassifier

from sklearn.linear_model import LogisticRegression

estimators = [('rf', rf), ('cat', cat_model)]

stacking_clf = StackingClassifier(estimators=estimators, final_estimator=LogisticRegression())

stacking_clf.fit(X_train, y_train)

y_stack_pred = stacking_clf.predict(X_test)
```

The stacking approach improved overall balanced accuracy to approximately 88%, outperforming all standalone models and providing a more reliable prediction across congestion categories.

Chapter 5: Results & Discussion

This chapter presents a comprehensive analysis of the machine learning model results, evaluating their predictive performance and discussing key insights and limitations identified.

Evaluation Metrics Overview

Model evaluation utilized multiple performance metrics commonly applied in classification problems:

- **Accuracy:** Percentage of correct predictions overall.
- **Precision:** Reliability of positive congestion predictions.
- **Recall:** Ability to capture true congestion events, minimizing false negatives.
- **F1-Score:** Harmonic mean of precision and recall, balancing false positives and negatives.

Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	**62%**	**60%**	**65%**	**62%**
CatBoost	**71%**	**70%**	**68%**	**69%**
XGBoost	**85%**	**83%**	**81%**	**82%**
Stacking Ensemble	**88%**	**85%**	**85%**	**85%**

The stacking ensemble model outperformed individual classifiers, exhibiting balanced improvements across all metrics.

Confusion Matrix Analysis

The confusion matrices revealed that most misclassifications occurred between adjacent congestion categories, particularly between moderate and high traffic levels. This reflects the inherent challenge in distinguishing marginal congestion variations that often occur in real-world scenarios.

Example Confusion Matrix (Stacking Ensemble):

Actual \ Predicted	**Low Traffic**	**Moderate Traffic**	**High Traffic**
Low Traffic	**15**	**2**	**0**
Moderate Traffic	**3**	**13**	**4**
High Traffic	**1**	**1**	**11**

Feature Importance

Analysis of feature importance unveiled the critical predictors influencing congestion:

- Average Daily Traffic Time: The strongest influencer, highlighting rush hour impact.
- Weather Parameters: Rainfall and fog negatively affect traffic flow, increasing congestion probabilities.
- Mode of Transport: Commuting by local trains was associated with lower congestion ratings compared to private vehicles.
- Travel Frequency: Daily commuters were more exposed to high congestion than weekly travelers.

Discussion

The results affirm the efficacy of ensemble learning to model complex, heterogeneous urban traffic patterns. However, challenges persist in accurately classifying borderline congestion categories due to overlapping feature spaces.

The integration of multi-source data including synthetic augmentation enhanced prediction robustness. Greater accuracy would benefit from larger, real-time datasets and deeper incorporation of spatial-temporal dynamics.

Overall, the project demonstrates the promise of AI-driven predictive models as practical components within urban transportation systems facilitating improved commuter experience and planning.

Chapter 6: Conclusion & Future Work

This research project successfully demonstrated the effective application of Artificial Intelligence (AI) and Machine Learning (ML) methodologies to predict traffic congestion in the densely populated metropolitan city of Mumbai. By integrating and analyzing multiple heterogeneous data sources — including primary commuter surveys, detailed weather statistics, comprehensive accident and road condition datasets, and innovative synthetic data augmentation — the project established a robust and multi-dimensional dataset essential for modeling such a complex urban traffic phenomenon.

The exploration and comparative evaluation of several advanced machine learning classifiers—Random Forest, CatBoost, and XGBoost—highlighted each model's unique strengths and weaknesses in addressing challenges related to heterogeneity, data imbalance, and feature complexity. Among these, XGBoost achieved the highest standalone predictive accuracy, while ensemble stacking techniques that combine multiple base models further enhanced overall predictive balance, stability, and reliability.

Contextual insights gained from feature importance analyses illuminated the dominant influence of temporal traffic patterns—such as peak hour effects—and a range of external conditions including adverse weather (rainfall, fog) and accident occurrences on congestion dynamics. These findings underscore the capacity and promise of data-driven, AI-powered prediction systems to transform urban transport management from reactive to proactive paradigms.

This project contributes a foundational AI modeling framework designed for scalability and integration within emerging smart city platforms. The framework holds strong potential for enabling real-time, adaptive traffic forecasting and intelligent management tools that can substantially improve commuter experiences, reduce environmental impacts, and foster sustainable urban mobility.

Future Work

To build upon the current achievements and address recognized limitations, several avenues for further research and development are proposed:

- Integration of Real-Time and Heterogeneous Data: Incorporate live traffic sensor feeds, GPS trajectory data, crowdsourced inputs, and mobile app data to provide dynamic, up-to-the-minute updates to congestion forecasts, enhancing model responsiveness.
- Scaling Dataset Size and Diversity: Expand survey participation and incorporate additional secondary data sources such as public event schedules, construction zones, and social media feeds to enrich model inputs and improve generalization.
- Advanced Deep Learning Techniques: Explore sequential learning models including Long Short-Term Memory (LSTM) networks, transformers, and graph neural

networks specially designed to capture temporal and spatial correlations in traffic data.

- Cloud-Based Real-Time Deployment: Develop scalable cloud infrastructure and APIs to operationalize traffic prediction models, coupled with user-friendly mobile and web applications delivering real-time alerts and route optimization suggestions.
- Enhanced Feature Engineering: Integrate location-specific features such as known choke points, road quality indices, and multi-modal transport interactions (road, rail, pedestrian) to build a holistic traffic ecosystem model.
- User-Centric Feedback and Personalization: Incorporate mechanisms for continual user feedback through mobile interfaces to iteratively refine prediction accuracy and tailor congestion information to individual commuter preferences.
- Robustness, Security, and Ethical Governance: Implement rigorous measures to safeguard data privacy and security, comply with legal frameworks, and maintain ethical standards in AI deployment to foster public trust and long-term sustainability.
- Policy Integration and Decision Support: Bridge AI forecasting outputs with urban policy-making and transport planning processes, enabling data-informed decision-making to alleviate congestion and promote smarter infrastructure development.

This project lays a significant foundation toward intelligent traffic management solutions not only for Mumbai but also as a scalable and adaptable template for other rapidly urbanizing megacities facing similar transportation challenges globally**.**

Appendix A: Code Listings

This appendix provides key Python code snippets utilized in the Mumbai Traffic Congestion Prediction project, covering data preprocessing, model training, and evaluation phases.

Data Preprocessing

Efficient data preprocessing is crucial to transform raw, multi-source datasets into a clean and model-ready format. The following example demonstrates how categorical features are encoded, missing data handled, and dataset imbalance addressed via SMOTE.

python

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from imblearn.over_sampling import SMOTE

# Load data
df = pd.read_csv('traffic_data.csv')

# Label encode binary variable
le = LabelEncoder()
df['Gender_encoded'] = le.fit_transform(df['Gender'])

# One-hot encode multi-category variable
ohe = OneHotEncoder()
mode_encoded = ohe.fit_transform(df[['Mode_of_Transport']])
mode_df = pd.DataFrame(mode_encoded.toarray(), columns=ohe.categories_[0])
df = pd.concat([df, mode_df], axis=1)

# Drop original categorical columns
df.drop(['Gender', 'Mode_of_Transport'], axis=1, inplace=True)

# Handle missing values by median/mode imputation (code not shown for brevity)
# Balance dataset using SMOTE to synthesize minority class samples
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(df.drop('Congestion_Level', axis=1), df['Congestion_Level'])
```

Model Training Example (XGBoost)

This snippet shows training with the XGBoost classifier, a powerful gradient boosting algorithm capable of modeling complex relationships with regularization to prevent overfitting.

python

```
import xgboost as xgb
from sklearn.model_selection import train_test_split

# Feature-target split
X = df.drop('Congestion_Level', axis=1)
y = df['Congestion_Level']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize and train XGBoost model
xgb_model = xgb.XGBClassifier(n_estimators=100, max_depth=5, learning_rate=0.1, random_state=42)
xgb_model.fit(X_train, y_train)

# Predictions on test data
y_pred = xgb_model.predict(X_test)
```

Model Evaluation and Visualization

Evaluation metrics and graphical analysis illustrate model effectiveness and aid in interpreting predictive strengths and weaknesses.

python

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns

# Classification report
print(classification_report(y_test, y_pred))

# Confusion matrix heatmap for visual error analysis
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# ROC AUC can be used for binary or one-vs-rest multi-class evaluation
# Example usage depends on data format and label binarization
```

Appendix B: Dataset Summary

- **Primary Survey Data:** Contains 57 records with features such as Age Group, Gender, Travel Frequency, Average Traffic Time, Mode of Transport, and labeled congestion severity. Provides insight into commuter perception and travel habits.
- **Secondary Data:** Sourced from government and weather agencies, detailing environmental factors (rainfall, fog, temperature), accident occurrences, traffic challans, and scheduled road maintenance affecting congestion.
- **Synthetic Data Generation:** Applied Gaussian noise perturbations to existing numeric features, creating over 2000 augmented samples. This process mitigates overfitting risks and enhances model generalization on unseen data.

Appendix C: Visualizations and Charts

Suggestions for key graphical inclusions that summarize data and analytic outcomes:

- **Project Workflow Diagram:** Illustrates end-to-end process flow from data collection, preprocessing, modeling to evaluation.
- **Confusion Matrices:** Heatmaps for each classifier to identify error patterns across congestion categories.
- **ROC Curves:** Multi-class ROC curves evaluating true positive rates versus false positives to assess classifier discrimination.
- **Feature Importance Bar Plots:** Highlight top variables influencing model decisions, assisting domain experts and planners.

Charts can be produced using Matplotlib or Seaborn libraries in Python, leveraging outputs from model metrics.

This detailed appendix suite enhances the transparency and reproducibility of the Mumbai Traffic Congestion Prediction project by providing essential code, data, and visual references.