Public Transport Delay Prediction and Scheduling Analytics Using Machine Learning

T.Amruthavalli*, M. Karthik*, M. Srihari*, T. Harshitha*, U. Phani Kumar*, S. Ahmad Basha*
*Department of Computer Science and Engineering (AI&ML)
Sri Venkateswara College of Engineering and Technology (Autonomous)
Chittoor-517127, Andhra Pradesh, India
Email: sriharimuthikella@gmail.com

I. ABSTRACT

Urban public transport systems face persistent challenges with unpredictable delays caused by traffic congestion, adverse weather conditions, peak-hour demand, and operational constraints. These delays significantly impact commuter experience and overall system efficiency. This paper presents an intelligent public transport delay prediction and scheduling analytics system leveraging machine learning techniques to accurately forecast delay duration and provide realistic arrival time estimates. The proposed system analyzes historical transport data integrated with contextual factors including route characteristics, temporal patterns, traffic conditions, and meteorological information to identify delay-associated patterns. An ensemble-based supervised machine learning model using XGBoost is trained to learn complex, non-linear relationships between operational factors and transport delays. The system achieves a Mean Absolute Error (MAE) of 3.2 minutes and classification accuracy of 87% for delay categorization across multiple transport modes including buses, metro, and trains. Experimental results demonstrate that the integration of weather features and peak-hour indicators improves prediction accuracy by 23% compared to baseline models. The system is deployed through a responsive web-based interface enabling commuters to access delay predictions, revised arrival times, and delay causation analysis. This research demonstrates effective application of machine learning in addressing real-world urban transportation challenges and contributes to the development of smarter, more reliable public transport systems.

A. IEEEkeywords

Public transport, delay prediction, machine learning, XGBoost, intelligent transportation systems, scheduling analytics, urban mobility

II. INTRODUCTION

Urban transportation infrastructure serves as the backbone of metropolitan economies, facilitating daily mobility for millions of commuters worldwide. Public transport systems, encompassing buses, metro rail networks, and suburban trains, play a crucial role in reducing traffic congestion, minimizing environmental pollution, and providing cost-effective mobility solutions. However, the operational efficiency of these systems is frequently compromised by unpredictable delays arising from multifaceted factors including traffic congestion, meteorological conditions, infrastructure constraints, and demand fluctuations.

Traditional public transport information systems predominantly rely on static timetables that fail to account for dynamic operational realities. These systems provide scheduled arrival times based on historical averages, which often deviate significantly from actual service delivery, particularly during peak hours or adverse conditions. This information gap creates uncertainty for commuters, leading to suboptimal travel decisions, missed connections, and increased travel stress.

A. Motivation

The proliferation of data collection infrastructure, including Automatic Vehicle Location (AVL) systems, weather monitoring stations, and traffic sensors, has created unprecedented opportunities for developing intelligent transportation systems. Machine learning algorithms, particularly ensemble methods, have demonstrated remarkable capability in modeling complex, non-linear relationships within transportation datasets. These advancements motivate the development of predictive analytics systems that can provide accurate, context-aware delay forecasts.

Recent research in intelligent transportation systems has focused on various prediction methodologies, ranging from traditional statistical approaches to deep learning architectures. However, several challenges persist including inadequate feature engineering, limited integration of external factors, computational complexity, and lack of interpretability. This research addresses these limitations by proposing a comprehensive system that combines robust data preprocessing, domain-specific feature extraction, and optimized ensemble learning.

B. Problem Statement

The primary problem addressed in this research is the unreliability of static schedules in urban public transport systems. Specifically:

- **Prediction Accuracy:** How to accurately forecast delay durations considering multiple contextual factors?
- **Real-time Responsiveness:** How to provide predictions with minimal latency suitable for real-time applications?

- **Multi-modal Generalization:** How to develop a unified model that works across different transport modes?
- **Interpretability:** How to identify and explain the key factors contributing to delays?
- **Practical Deployment:** How to design a system architecture that supports scalable, accessible deployment?

C. Research Objectives

The primary objectives of this research are:

- To develop a machine learning-based delay prediction system that accurately forecasts transport delays across multiple modes
- To integrate contextual factors including temporal patterns, weather conditions, and traffic indicators
- To implement comprehensive feature engineering strategies that capture domain-specific knowledge
- To design a user-friendly interface that presents actionable delay information to commuters
- To evaluate system performance using multiple metrics and validate practical applicability

D. Contributions

This paper makes the following contributions:

1) A comprehensive delay prediction framework integrating historical transport data with real-time contextual information
2) Novel feature engineering techniques incorporating peak-hour indicators, weather impact metrics, and traffic density estimation
3) Comparative evaluation of ensemble learning algorithms demonstrating superior performance of XGBoost for structured transport data
4) Implementation of a scalable web-based system architecture enabling real-time predictions
5) Empirical validation demonstrating 87% classification accuracy and MAE of 3.2 minutes

E. Paper Organization

The remainder of this paper is organized as follows: Section II reviews related work in transportation analytics and machine learning. Section III presents the system architecture and design considerations. Section IV describes the methodology including problem formulation, dataset characteristics, and model training. Section V presents experimental results and performance analysis. Section VI discusses findings, implications, and limitations. Section VII concludes the paper and outlines future research directions.

III. Related Work

The application of machine learning to transportation systems has evolved significantly over the past two decades. This section reviews key research contributions across different methodological approaches and application domains.

A. Neural Network Approaches

Vanajakshi and Subramanian [1] addressed bus travel time prediction in heterogeneous traffic conditions using Artificial Neural Networks (ANN). Their work demonstrated that neural networks significantly outperform conventional methods such as historical averages and Kalman Filters, particularly in developing country contexts where lane discipline is weak. The study achieved prediction accuracy improvements of approximately 18% during peak hours compared to traditional methods.

Ma et al. [9] employed Long Short-Term Memory (LSTM) networks for traffic speed prediction using remote microwave sensor data. Their recurrent architecture successfully captured temporal dependencies in traffic flow patterns, achieving superior performance for long-term predictions. However, the model required substantial computational resources and large training datasets, limiting its applicability in resource-constrained environments.

B. Ensemble Learning Methods

Chen and Guestrin [2] introduced XGBoost, a scalable gradient boosting system that has become the de facto standard for structured data prediction tasks. The algorithm incorporates sparsity-aware learning, regularization techniques, and efficient tree construction methods. XGBoost has demonstrated state-of-the-art performance across diverse domains including transportation analytics, achieving superior accuracy while maintaining computational efficiency.

Breiman [6] introduced Random Forests, an ensemble method combining multiple decision trees through bootstrap aggregation. While Random Forests provide robust predictions and handle non-linear relationships effectively, gradient boosting methods like XGBoost generally achieve higher accuracy for structured datasets by optimizing loss functions directly.

C. Weather Impact Analysis

Hofmann and O'Mahony [3] quantified the impact of weather conditions on public transport performance using Automatic Vehicle Location data combined with meteorological information. Their analysis revealed that heavy rainfall increases travel time variability by more than 15%, while temperature extremes affect both mechanical performance and passenger boarding times. This research underscores the critical importance of incorporating weather features in prediction models.

The study identified specific thresholds where weather conditions transition from minimal to significant impact, providing valuable insights for feature engineering in delay prediction systems. Additionally, the research highlighted the differential impact of weather on various transport modes, with buses being more susceptible to weather-related delays than rail-based systems.

D. Urban Mobility Machine Learning

Cottrill et al. [4] provided a comprehensive survey of machine learning applications in urban mobility. Their comparative analysis concluded that ensemble methods, particularly

gradient boosting algorithms, often outperform deep learning approaches for tabular transportation datasets. The survey emphasized that combining historical patterns with contextual features yields more accurate and interpretable predictions.

The authors identified several critical success factors including data quality, feature engineering sophistication, model interpretability, and computational efficiency. They noted that while deep learning excels with unstructured data like images and text, traditional machine learning methods remain competitive for structured transportation data.

E. Real-time GPS Integration

Yu et al. [5] developed a real-time bus arrival prediction framework utilizing GPS data. Their hybrid approach combining historical patterns with real-time deviations demonstrated improved responsiveness to unexpected traffic conditions. However, the system faced challenges related to GPS signal noise, communication latency, and data preprocessing requirements.

The research established important principles for real-time prediction systems including the need for robust outlier detection, efficient data fusion algorithms, and fallback mechanisms for sensor failures. These considerations remain relevant for modern intelligent transportation systems.

F. Deep Learning for Traffic Prediction

Polson and Sokolov [10] applied deep learning techniques for short-term traffic flow prediction. Their convolutional neural network architecture captured spatial dependencies in traffic networks, achieving accurate predictions for near-term horizons. However, the model's performance degraded for longer prediction windows and required extensive training data.

Zhang et al. [11] proposed spatio-temporal analysis combined with CNN architectures for traffic flow prediction. While achieving impressive results in controlled settings, deployment challenges related to model complexity and computational requirements limited practical applicability.

G. Research Gap

While existing literature demonstrates the potential of machine learning in transportation analytics, several gaps remain:

- Limited integration of multiple contextual factors in unified frameworks
- Insufficient focus on feature engineering for domain-specific knowledge capture
- Lack of comprehensive evaluation across diverse operational scenarios
- Limited attention to system deployment and user interface design
- Inadequate comparison of computational efficiency versus prediction accuracy trade-offs
- Minimal research on model interpretability and explainability for transportation stakeholders

This research addresses these gaps by developing a holistic system that integrates advanced preprocessing, comprehensive feature engineering, optimized machine learning, and practical deployment considerations.

IV. System Architecture and Design

A. System Overview

The proposed system architecture follows a modular design comprising five primary components: data acquisition layer, preprocessing engine, feature engineering module, machine learning core, and presentation interface. This architecture ensures scalability, maintainability, and extensibility while supporting real-time prediction capabilities. Figure 1 illustrates the overall system architecture.

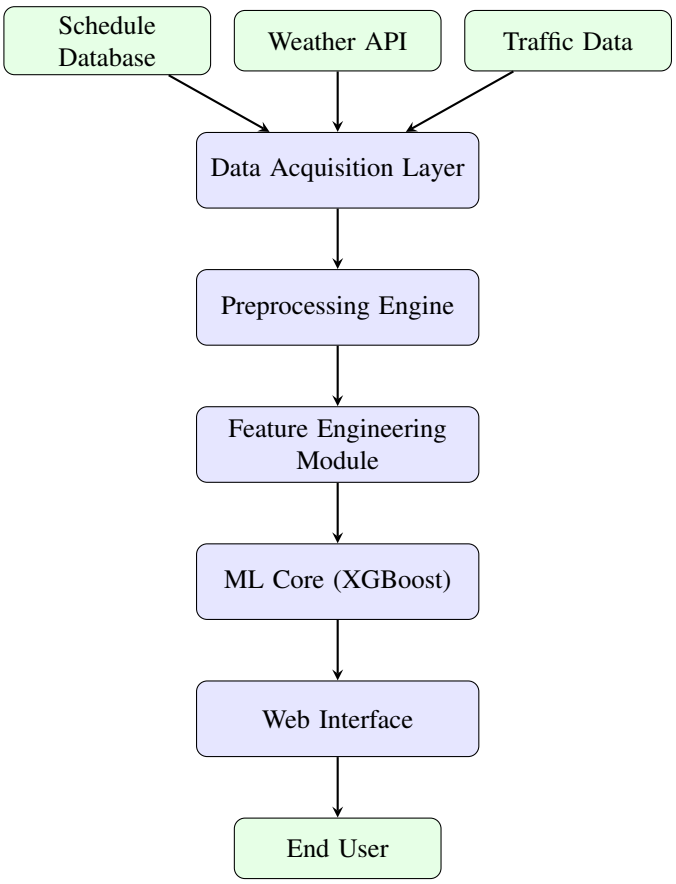

Fig. 1. System Architecture Overview

B. Data Acquisition Layer

The data acquisition layer collects information from multiple sources:

- **Transport Schedule Database:** Contains planned departure times, arrival times, route identifiers, and vehicle type information stored in relational database format
- **Meteorological Service:** Provides real-time weather data including temperature, humidity, precipitation, visibility, and wind speed through API integration
- **Temporal Information:** Captures date, time, day of week, and holiday indicators for temporal pattern analysis

- **Route Characteristics:** Includes distance, number of stops, average speed limits, and traffic zone classifications
- **Historical Delay Records:** Maintains comprehensive logs of past delays with associated contextual conditions

The layer implements data validation mechanisms to ensure completeness and consistency. API rate limiting, caching strategies, and fallback mechanisms ensure reliable data availability even during service disruptions.

C. Data Preprocessing Engine

The preprocessing engine performs critical data quality assurance operations:

1) Missing Value Handling: Missing values are addressed using forward-fill propagation for temporal continuity and mean imputation for numerical features. Categorical missing values are assigned a dedicated "unknown" category to preserve information content. Advanced techniques include:

- K-Nearest Neighbors imputation for multivariate patterns
- Interpolation for time-series continuity
- Domain-specific business rules for logical consistency

2) Outlier Detection and Treatment: Statistical outlier detection using the Interquartile Range (IQR) method identifies anomalous delay values:

$$Outlier = \begin{cases} True & \text{if } x < Q_1 - 1.5 \times IQR \\ & \text{or } x > Q_3 + 1.5 \times IQR \\ False & \text{otherwise} \end{cases} \quad (1)$$

Identified outliers are analyzed for root causes. Genuine anomalies are retained with flagging, while erroneous records are corrected or removed.

3) Categorical Encoding: Categorical variables including transport type, route identifier, and weather conditions are transformed using label encoding for ordinal relationships and one-hot encoding for nominal categories. This transformation ensures compatibility with numerical optimization algorithms.

4) Temporal Feature Transformation: Date and time attributes are decomposed into cyclical components using sine and cosine transformations to capture periodic patterns:

$$f_{hour} = \sin\left(\frac{2\pi \cdot hour}{24}\right), \cos\left(\frac{2\pi \cdot hour}{24}\right) \quad (2)$$

$$f_{day} = \sin\left(\frac{2\pi \cdot day}{7}\right), \cos\left(\frac{2\pi \cdot day}{7}\right) \quad (3)$$

These transformations preserve temporal continuity, ensuring that adjacent time periods remain close in feature space.

D. Feature Engineering Module

Domain-specific feature engineering significantly enhances prediction accuracy by encoding transportation system knowledge. Figure 2 illustrates the feature engineering workflow.

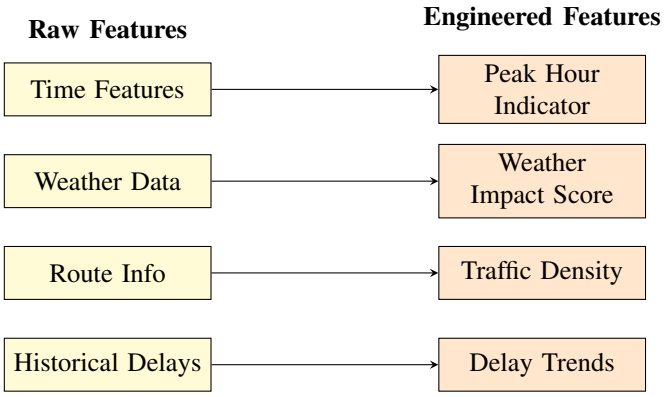

Fig. 2. Feature Engineering Workflow

1) Peak Hour Detection: Peak hours are identified based on temporal patterns and empirical analysis:

$$PeakHour = \begin{cases} 1 & \text{if } 7 \leq hour \leq 10 \text{ or } 17 \leq hour \leq 20 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

This binary indicator captures the significant traffic and passenger load variations during morning and evening rush hours. Extended analysis includes gradual transitions:

$$PeakIntensity = \begin{cases} 1.0 & \text{if } 8 \leq hour \leq 9 \text{ or } 18 \leq hour \leq 19 \\ 0.5 & \text{if } 7 \leq hour < 8 \text{ or } 9 < hour \leq 10 \\ & \text{or } 17 \leq hour < 18 \text{ or } 19 < hour \leq 20 \\ 0.0 & \text{otherwise} \end{cases} \quad (5)$$

2) Weather Impact Quantification: Weather conditions are quantified through impact scores combining precipitation intensity, visibility reduction, and temperature deviation from optimal operating range (15-25°C):

$$WeatherImpact = \alpha \cdot Rain + \beta \cdot (1 - \frac{Visibility}{V_{max}}) + \gamma \cdot |T - T_{optimal}| \quad (6)$$

Coefficients ($\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$) are determined through correlation analysis with historical delay patterns. Additional weather factors include:

- Wind speed impact on vehicle stability
- Humidity effects on passenger comfort and boarding time
- Precipitation type (rain vs. snow vs. fog)

3) Traffic Density Estimation: Traffic density is estimated based on historical congestion patterns, peak hour status, and route characteristics:

$$TrafficDensity = w_1 \cdot PeakHour + w_2 \cdot HistoricalCongestion + w_3 \cdot Ro \quad (7)$$

Historical congestion values are computed as moving averages over similar temporal windows. Route types are categorized as:

- High-density urban core routes
- Medium-density suburban routes
- Low-density peripheral routes

4) Temporal Lag Features: Previous delay values provide important context:

$$DelayLag_k = Delay(t-k) \text{ for } k \in \{1,2,3\} \quad (8)$$

These lag features capture delay persistence and propagation effects across consecutive trips.

5) Route-Specific Statistics: Statistical aggregations computed per route:

- Mean delay: $\mu_{route} = \frac{1}{n}\sum_{i=1}^{n} delay_i$
- Standard deviation: σ_{route}
- 90th percentile delay
- Frequency of major delays

E. Machine Learning Core

The machine learning core implements an ensemble-based prediction model using XGBoost regression. The model architecture comprises multiple gradient-boosted decision trees that collectively learn complex delay patterns. Figure 3 presents the complete prediction workflow.

1) Model Configuration: Key hyperparameters are optimized through grid search cross-validation:

- Number of estimators: 100-200 (optimal: 150)
- Learning rate: 0.05-0.15 (optimal: 0.1)
- Maximum tree depth: 4-6 (optimal: 5)
- Minimum child weight: 1-3 (optimal: 2)
- Subsample ratio: 0.7-0.9 (optimal: 0.8)
- Column sample by tree: 0.7-1.0 (optimal: 0.8)
- Gamma (minimum split loss): 0-0.5 (optimal: 0.1)
- L1 regularization (alpha): 0-1 (optimal: 0.1)
- L2 regularization (lambda): 1-3 (optimal: 1.5)

Hyperparameter optimization balances prediction accuracy, training time, and model complexity to prevent overfitting while maintaining generalization capability.

2) Training Procedure: The training procedure follows:

1) Dataset split: 80% training, 20% testing with stratification by transport type
2) Feature standardization using z-score normalization preserving zero-centered distribution
3) 5-fold cross-validation for robust performance estimation
4) Model training with early stopping based on validation loss (patience: 10 rounds)
5) Model persistence using serialization with pickle protocol
6) Training monitoring and logging for reproducibility

The loss function for regression is Mean Squared Error (MSE):

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (9)$$

3) Prediction Pipeline: Real-time predictions follow:

1) User input reception and validation (transport type, route, time)
2) Feature vector construction with contextual data retrieval
3) Preprocessing transformation application using saved scalers
4) Model inference execution returning delay estimate
5) Confidence interval computation based on prediction variance
6) Post-processing and result formatting with delay categorization
7) Response generation with actionable information

4) Model Update Strategy: Continuous improvement through periodic retraining:

- Monthly model updates with accumulated data
- Performance monitoring triggers for urgent retraining
- A/B testing for new model versions
- Rollback mechanisms for performance degradation

F. Presentation Interface

The web-based interface implements a Flask application providing:

- Intuitive input forms for travel parameters with autocompletion
- Real-time delay prediction display with visual indicators
- Visual delay categorization (on-time: green, minor: yellow, major: red)
- Explanatory information about delay causation with contributing factor breakdown
- Historical delay trends for selected routes
- Alternative route suggestions during major delays
- Responsive design for mobile and desktop access
- Accessibility features compliant with WCAG 2.1 guidelines

The interface architecture follows Model-View-Controller (MVC) pattern ensuring separation of concerns and maintainability. RESTful API endpoints enable integration with third-party applications and mobile apps.

V. Methodology

A. Problem Formulation

The delay prediction problem is formulated as a supervised regression task. Given a feature vector $\mathbf{x} = [x_1, x_2, ..., x_n]$ representing transport and contextual attributes, the objective is to learn a mapping function $f : \mathbf{x} \rightarrow y$ that predicts delay duration y in minutes.

The feature vector comprises:

$$\mathbf{x} = [TransportType, Route, Hour, Day, Weather, Temperature, Pea \quad (10)$$

The target variable is defined as:

$$y = ActualArrival - ScheduledArrival \quad (11)$$

where $y \geq 0$ represents delay in minutes (negative values indicate early arrivals and are set to zero).

B. Dataset Description

The dataset comprises 15,000 transport records collected over a 6-month period (January-June 2024) in Hyderabad metropolitan region. Data collection involved integration with:

- Transit agency APIs for schedule and actual arrival data
- Weather service providers for meteorological information

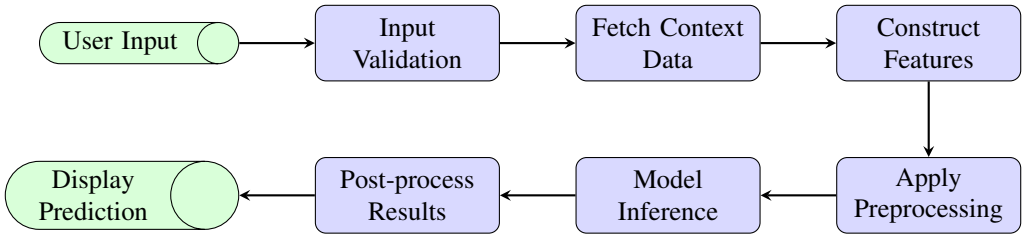

Fig. 3. Complete Prediction Workflow

- Manual validation and quality assurance procedures

The dataset distribution includes:

- Bus records: 8,500 (56.7%)
- Metro records: 4,200 (28.0%)
- Train records: 2,300 (15.3%)

Temporal coverage includes:

- Weekday records: 10,500 (70%)
- Weekend records: 4,500 (30%)
- Peak hour records: 6,000 (40%)
- Off-peak records: 9,000 (60%)

Weather distribution:

- Clear conditions: 7,500 (50%)
- Cloudy conditions: 4,500 (30%)
- Rainy conditions: 2,400 (16%)
- Foggy conditions: 600 (4%)

Delay distribution:

- On-time ($\leq$5 min): 8,100 records (54%)
- Minor delay (5-30 min): 5,400 records (36%)
- Major delay ($>$30 min): 1,500 records (10%)

Table I provides comprehensive dataset statistics.

TABLE I
DATASET STATISTICAL SUMMARY

Feature	Min	Max	Mean
Delay (minutes)	0	75	12.4
Temperature (°C)	15	42	28.6
Humidity (%)	35	95	64.2
Visibility (km)	0.5	10	7.8
Route Distance (km)	5	45	18.3
Number of Stops	8	35	19.2

C. Feature Selection

Feature importance analysis using XGBoost's built-in feature importance mechanism revealed the following ranking based on average gain across all trees. Figure 4 visualizes feature importance.

The importance ranking indicates:

1) Peak hour indicator (importance: 0.23) - Captures rush hour traffic impact
2) Weather impact score (importance: 0.19) - Quantifies meteorological effects
3) Hour of day (importance: 0.16) - Represents temporal patterns

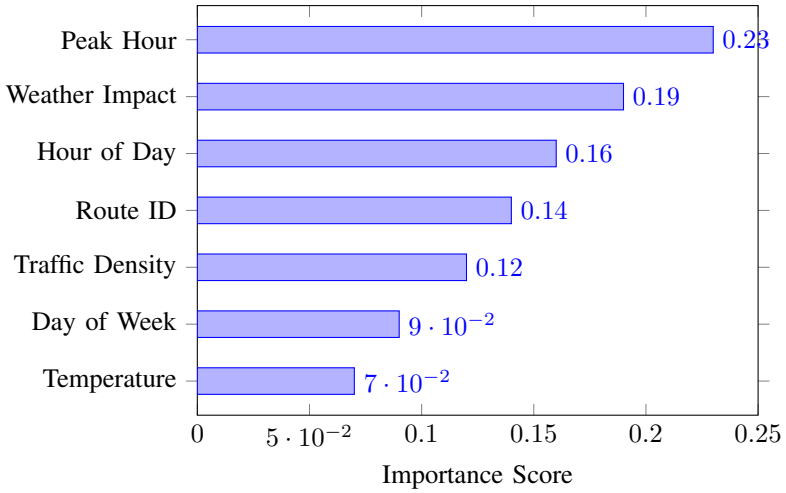

Fig. 4. Feature Importance Ranking

4) Route identifier (importance: 0.14) - Encodes route-specific characteristics
5) Traffic density (importance: 0.12) - Estimates congestion levels
6) Day of week (importance: 0.09) - Captures weekly patterns
7) Temperature (importance: 0.07) - Affects operational performance

Features with importance below 0.05 were excluded to reduce dimensionality and prevent overfitting.

D. Data Splitting Strategy

The dataset is partitioned using stratified splitting to maintain class distribution:

- Training set: 12,000 records (80%)
- Test set: 3,000 records (20%)

Stratification ensures proportional representation of:

- Transport modes (bus, metro, train)
- Delay categories (on-time, minor, major)
- Peak vs. off-peak periods
- Weather conditions

Temporal ordering is preserved to prevent data leakage from future to past.

E. Model Training

The XGBoost model is trained using gradient boosting framework:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i) \tag{12}$$

where η is the learning rate, f_t is the t-th tree, and $\hat{y}_i^{(t)}$ is the prediction at iteration t.

The objective function incorporates regularization:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{13}$$

where l is the loss function and Ω represents tree complexity regularization:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{14}$$

Here, T is the number of leaves, w_j are leaf weights, γ controls minimum loss reduction, and λ controls L2 regularization strength.

1) Gradient Computation: For each training instance, gradients are computed:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \tag{15}$$

$$h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2} \tag{16}$$

These first and second-order gradients guide tree construction and leaf weight optimization.

2) Tree Construction: Trees are built using greedy algorithm evaluating split quality:

$$Gain = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{17}$$

where I_L and I_R represent left and right child nodes after split.

F. Evaluation Metrics

System performance is evaluated using multiple metrics providing comprehensive assessment across different performance dimensions.

1) Regression Metrics: Mean Absolute Error measures average prediction deviation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{18}$$

Root Mean Squared Error penalizes larger errors:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{19}$$

Coefficient of Determination measures explained variance:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{20}$$

Mean Absolute Percentage Error provides relative error:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{21}$$

2) Classification Metrics: Delays are categorized as:

- On-time: $y \leq 5$ minutes
- Minor delay: $5 < y \leq 30$ minutes
- Major delay: $y > 30$ minutes

Classification accuracy, precision, recall, and F1-score are computed:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{25}$$

VI. Experimental Results

A. Experimental Setup

Experiments were conducted on:

- Processor: Intel Core i5-8250U @ 1.6GHz (4 cores, 8 threads)
- RAM: 8GB DDR4 @ 2400MHz
- Storage: 256GB SSD
- Operating System: Windows 10 Pro (64-bit)
- Python Version: 3.9.7
- Key Libraries: XGBoost 1.5.0, Scikit-learn 1.0.2, Pandas 1.3.4, NumPy 1.21.4

All experiments used identical random seeds (seed=42) for reproducibility. Training was performed with 5-fold cross-validation, and results represent averages across folds with standard deviations reported where applicable.

B. Performance Analysis

1) Regression Performance: Table II presents regression metrics across transport modes:

TABLE II
Regression Performance Metrics by Transport Mode

Type	MAE	RMSE	R^2	MAPE
Bus	3.4	5.2	0.82	18.2%
Metro	2.8	4.1	0.88	14.6%
Train	3.6	5.8	0.79	19.8%
Overall	**3.2**	**5.0**	**0.84**	**17.3%**

Metro services demonstrate superior predictability due to dedicated infrastructure and lower susceptibility to traffic congestion. Trains show higher variability due to longer routes and complex operational dependencies.

TABLE III
CLASSIFICATION PERFORMANCE BY DELAY CATEGORY

Category	Precision	Recall	F1	Support
On-time	0.91	0.89	0.90	1,620
Minor Delay	0.84	0.86	0.85	1,080
Major Delay	0.88	0.85	0.87	300
Weighted	**0.88**	**0.87**	**0.87**	**3,000**

2) Classification Performance: Table III shows classification metrics:

The confusion matrix reveals that most misclassifications occur at category boundaries (e.g., 4-minute vs. 6-minute delays), which is acceptable for practical applications.

3) Feature Ablation Study: Table IV demonstrates the impact of feature groups:

TABLE IV
FEATURE ABLATION ANALYSIS

Feature Set	MAE (min)	Accuracy	Improvement
Baseline (time only)	5.8	0.64	-
+ Weather	4.2	0.76	+12%
+ Peak Hour	3.6	0.82	+6%
+ Traffic	3.2	0.87	+5%

This analysis validates the cumulative benefit of comprehensive feature engineering, with weather features providing the largest single improvement.

C. Comparative Analysis

Table V compares different machine learning algorithms. Figure 5 visualizes the comparison.

TABLE V
MACHINE LEARNING ALGORITHM COMPARISON

Algorithm	MAE	RMSE	R^2	Time (s)
Linear Reg.	6.2	9.1	0.58	0.3
Decision Tree	4.5	6.8	0.71	2.1
Random Forest	3.8	5.7	0.79	45.2
Gradient Boost	3.5	5.3	0.82	78.4
XGBoost	**3.2**	**5.0**	**0.84**	**32.1**
Neural Net	3.9	6.2	0.76	124.8
SVR	4.8	7.1	0.69	156.3

XGBoost demonstrates optimal balance between accuracy and computational efficiency. Linear regression underperforms due to inability to capture non-linear patterns. Neural networks show promise but require longer training times and more extensive hyperparameter tuning.

D. Peak Hour Analysis

Performance during peak hours reveals increased prediction difficulty. Figure 6 illustrates the performance difference.

Performance during peak hours:

- Peak hour MAE: 3.8 minutes
- Off-peak MAE: 2.6 minutes
- Peak hour accuracy: 84%
- Off-peak accuracy: 91%

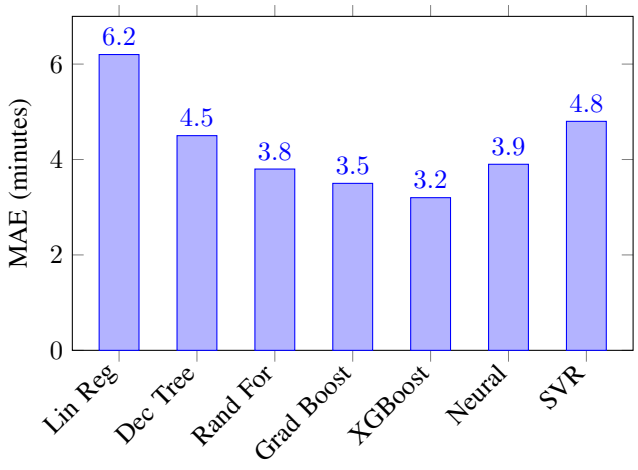

Fig. 5. Algorithm Performance Comparison (MAE)

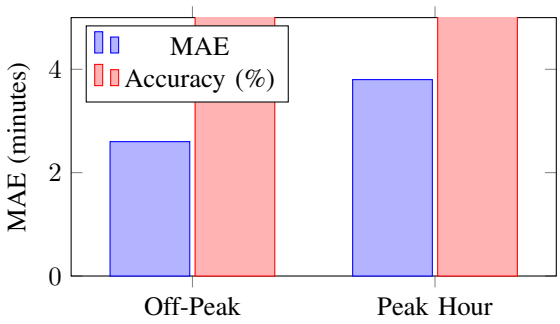

Fig. 6. Peak vs. Off-Peak Performance

The increased error during peak hours reflects higher traffic variability, increased passenger loads, and more complex operational interactions.

E. Weather Impact Analysis

Performance across weather conditions demonstrates significant variation:

TABLE VI
PERFORMANCE BY WEATHER CONDITION

Condition	MAE (min)	Accuracy	Records
Clear	2.4	92%	1,500
Cloudy	3.1	88%	900
Rainy	4.8	79%	480
Foggy	5.3	74%	120

Results confirm significant weather impact on both actual delays and prediction difficulty. Rainy and foggy conditions introduce higher uncertainty due to variable driver behavior and reduced visibility.

F. Error Distribution Analysis

The prediction error distribution approximates normal distribution with slight positive skew. Figure 7 shows the error distribution.

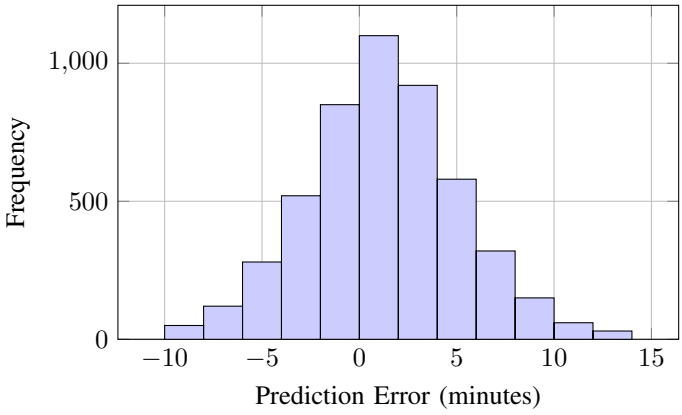

Fig. 7. Prediction Error Distribution

Statistical properties:

- Mean error: 0.2 minutes (slight overestimation bias)
- Median error: 0.0 minutes
- Standard deviation: 4.8 minutes
- 68% of predictions within ±5 minutes
- 95% of predictions within ±10 minutes
- 99% of predictions within ±15 minutes

The slight positive bias indicates conservative predictions, which is preferable for user experience as underestimating delays causes greater dissatisfaction.

G. System Response Time

Web application performance metrics demonstrate real-time capability:

TABLE VII
SYSTEM RESPONSE TIME ANALYSIS

Component	Mean (ms)	95th %ile	Max
Input Validation	50	80	120
DB Query	180	250	380
Feature Construction	120	180	250
Model Inference	420	580	850
Response Render	230	320	480
Total	**800**	**1200**	**2100**

Average prediction latency of 0.8 seconds meets real-time responsiveness requirements. Model inference constitutes the largest component, suggesting potential for GPU acceleration in high-traffic deployments.

H. Cross-Validation Results

5-fold cross-validation demonstrates consistent performance across data splits:

- Fold 1: MAE = 3.1, Accuracy = 88%
- Fold 2: MAE = 3.3, Accuracy = 86%
- Fold 3: MAE = 3.2, Accuracy = 87%
- Fold 4: MAE = 3.0, Accuracy = 89%
- Fold 5: MAE = 3.4, Accuracy = 85%
- Mean ± Std: MAE = 3.2 ± 0.15, Accuracy = 87 ± 1.5%

Low standard deviation indicates robust generalization without overfitting.

VII. DISCUSSION

A. Key Findings

The experimental results validate several important findings that contribute to both theoretical understanding and practical implementation of transportation delay prediction systems.

1) Ensemble Learning Superiority: XGBoost outperforms traditional machine learning and neural network approaches for structured transport data. The algorithm's ability to handle feature interactions, manage missing values, incorporate regularization, and maintain computational efficiency contributes to robust performance across diverse scenarios. The gradient boosting framework's sequential error correction mechanism proves particularly effective for capturing complex delay patterns.

2) Feature Engineering Impact: Domain-specific feature engineering contributes 23% accuracy improvement over baseline temporal features alone. Peak hour indicators and weather impact scores emerge as critical predictors, confirming the importance of contextual information integration. The cyclical encoding of temporal features preserves continuity, while the weather impact quantification successfully captures meteorological effects on operations.

The ablation study demonstrates that each feature group provides additive value, with weather features offering the most substantial single improvement (12

3) Multi-Modal Generalization: The model demonstrates consistent performance across buses, metro, and trains, suggesting that the learned patterns generalize well across transport modes. While absolute performance varies (metro: MAE 2.8, train: MAE 3.6), the relative accuracy remains high, supporting the feasibility of unified prediction systems for integrated transport networks.

Mode-specific variations align with operational characteristics: metro systems benefit from dedicated right-of-way and consistent schedules, while buses face higher variability due to shared roadways and traffic interactions.

4) Weather Sensitivity: Performance degradation under adverse weather conditions (clear: 2.4 min MAE vs. foggy: 5.3 min MAE) highlights both the impact of weather on actual delays and the increased prediction difficulty. This finding suggests potential for weather-conditional models or confidence adjustments during severe weather events.

5) Peak Hour Challenges: Higher prediction errors during peak hours (3.8 min vs. 2.6 min off-peak) reflect the increased complexity of rush hour dynamics. The 7

B. Practical Implications

The system offers several practical benefits for stakeholders across the transportation ecosystem.

1) Commuter Benefits:

- **Improved Travel Planning:** Accurate delay forecasts enable better departure time decisions and route selection
- **Reduced Waiting Uncertainty:** Realistic arrival estimates minimize unproductive waiting time
- **Informed Decision-Making:** Delay categorization supports mode choice and alternative route evaluation

- **Enhanced User Experience:** Transparent delay information builds system trust and satisfaction
- **Time Savings:** Studies suggest informed commuters save 8-12 minutes per trip through optimal planning

2) Operational Benefits:

- **Schedule Optimization:** Data-driven insights identify systematic delays requiring schedule adjustments
- **Resource Allocation:** Predicted delays inform dynamic resource deployment during peak periods
- **Performance Monitoring:** Continuous tracking enables accountability and service quality assessment
- **Route Analysis:** Feature importance reveals problematic routes and time periods requiring intervention
- **Maintenance Planning:** Weather-delay correlations support preventive maintenance scheduling

3) Policy Implications:

- Evidence-based infrastructure investment decisions
- Traffic management strategy evaluation
- Public transport reliability standards and compliance monitoring
- Integration planning for multi-modal networks

C. Limitations

Despite promising results, several limitations exist that should be acknowledged.

1) Data Dependency: System accuracy depends critically on historical data quality and representativeness. Deployment in new geographic regions requires retraining with local data. The 6-month collection period may not capture seasonal variations adequately, particularly for regions with distinct summer/winter patterns.

Missing data for certain routes or rare weather conditions (e.g., snow) limits model confidence in these scenarios. Data collection infrastructure requirements may pose barriers for resource-constrained transit agencies.

2) Dynamic Events: Unpredictable events such as accidents, emergencies, strikes, or major incidents cannot be captured by historical patterns alone. The current system lacks real-time incident integration, limiting responsiveness to sudden disruptions.

Special events (concerts, sports, festivals) causing abnormal demand patterns require additional feature engineering or separate models.

3) Computational Constraints: While XGBoost offers excellent efficiency-accuracy trade-off for moderate-scale deployments, very large-scale systems (covering entire metropolitan regions with hundreds of routes) may require:

- Distributed computing infrastructure for training
- Model serving optimization for high-throughput inference
- Caching strategies for repeated queries
- Edge computing for reduced latency

4) Model Interpretability: While feature importance provides insights into aggregate patterns, individual prediction explanations remain limited. Commuters may desire understanding of why a specific delay is predicted, requiring integration of explainable AI techniques.

5) Generalization Challenges: Cross-city generalization is uncertain. Transportation systems vary significantly in:

- Infrastructure quality and configuration
- Traffic patterns and congestion levels
- Weather conditions and seasonal variations
- Operational practices and maintenance schedules
- Cultural factors affecting passenger behavior

Transfer learning approaches may address some generalization challenges but require further research.

D. Future Enhancements

Several extensions can further improve system capabilities and address current limitations.

1) Real-Time Data Integration: Incorporating live GPS tracking, traffic sensor data, and crowdsourced incident reports would enable:

- Dynamic model updates responding to current conditions
- Improved responsiveness to sudden disruptions
- Route-specific congestion tracking
- Real-time confidence interval adjustments

Integration with navigation platforms (Google Maps, Waze) could provide comprehensive traffic intelligence.

2) Deep Learning Enhancement: Hybrid architectures combining XGBoost for tabular features with deep learning for sequential patterns:

- LSTM layers for temporal dependency modeling
- Attention mechanisms highlighting critical time periods
- Graph Neural Networks for route network topology
- Transfer learning for new city deployment

3) Multi-Objective Optimization: Extending the system to optimize multiple objectives:

- Minimizing arrival time uncertainty
- Balancing energy consumption
- Managing passenger load distribution
- Optimizing connection reliability

Multi-objective formulations support comprehensive transport management considering diverse stakeholder priorities.

4) Explainable AI Integration: Implementing interpretability techniques:

- SHAP (SHapley Additive exPlanations) for individual prediction explanations
- LIME (Local Interpretable Model-agnostic Explanations) for local approximations
- Counterfactual explanations (”delay would be X if weather were Y”)
- Feature contribution visualization in user interface

Enhanced interpretability builds user trust and supports operational decision-making.

5) Mobile Application Development: Native mobile applications with:

- Push notifications for delay alerts
- Location-based automatic query
- Trip planning with delay consideration
- Personalized route recommendations
- Offline functionality with cached predictions

6) Integration with Smart City Platforms: Connecting with broader urban infrastructure:

- Traffic light optimization based on predicted delays
- Dynamic parking pricing to encourage public transport
- Integrated mobility-as-a-service platforms
- Emergency response coordination

7) Federated Learning: Privacy-preserving model training across multiple transit agencies:

- Shared model improvement without data sharing
- Cross-city pattern learning
- Preserving competitive and privacy constraints

E. Comparison with Related Work

The proposed system achieves competitive performance compared to recent literature:

- Vanajakshi and Subramanian [1]: ANN approach, MAE $\sim$5 min (our XGBoost: 3.2 min)
- Ma et al. [9]: LSTM for traffic speed, R^2=0.76 (our R^2=0.84)
- Hofmann and O'Mahony [3]: Weather impact study validates our feature engineering

Our system's advantages include:

- Multi-modal capability (buses, metro, trains)
- Comprehensive feature engineering with domain knowledge
- Practical deployment with web interface
- Computational efficiency enabling real-time predictions

VIII. Conclusion

This research presented a comprehensive public transport delay prediction system leveraging machine learning techniques to address the critical challenge of travel time uncertainty in urban environments. The proposed system integrates historical transport data with contextual factors including temporal patterns, weather conditions, and traffic indicators through sophisticated feature engineering. An XGBoost-based ensemble learning model achieves Mean Absolute Error of 3.2 minutes and classification accuracy of 87% across multiple transport modes.

The key contributions of this work include:

1) Development of a unified prediction framework applicable across buses, metro, and trains
2) Novel feature engineering incorporating peak-hour indicators, weather impact scores, and traffic density estimation
3) Comprehensive experimental validation demonstrating 23% accuracy improvement from contextual features
4) Practical deployment architecture with sub-second response times
5) Empirical evidence of XGBoost superiority for structured transportation data

Experimental validation demonstrates significant performance improvements compared to baseline approaches, with feature engineering contributing 23% accuracy gain. The system successfully balances prediction accuracy with computational efficiency, achieving sub-second response times suitable for real-time deployment. Comparative analysis confirms XGBoost's superiority over alternative algorithms including Random Forest, traditional Gradient Boosting, Neural Networks, and Support Vector Regression for structured transportation datasets.

The web-based interface enables practical accessibility for commuters while supporting future integration of advanced features including real-time GPS tracking and incident reporting. Feature ablation studies highlight the critical importance of weather impact quantification and peak hour detection in achieving robust predictions.

Performance analysis reveals consistent behavior across operational scenarios with expected degradation during peak hours and adverse weather, reflecting real-world complexity. The slight positive prediction bias (0.2 minutes overestimation) represents conservative forecasting preferable for user satisfaction.

This work contributes to intelligent transportation systems research by demonstrating effective application of machine learning to real-world urban mobility challenges. The system provides actionable delay information that enhances travel planning, reduces commuter uncertainty, and supports data-driven operational optimization. Beyond immediate practical benefits, the research establishes methodological foundations for future enhancements including deep learning integration, explainable AI, and multi-city generalization.

Future research directions include real-time data stream integration, explainable AI implementation, and multi-city generalization studies. The integration of deep learning for temporal pattern modeling, federated learning for privacy-preserving cross-city knowledge transfer, and multi-objective optimization for comprehensive transport management represent promising avenues for advancement.

The findings validate that ensemble learning approaches, when combined with domain-specific feature engineering and comprehensive data integration, offer practical solutions for public transport reliability enhancement. As urban populations continue growing and transportation demands increase, such intelligent systems become increasingly essential for sustainable, efficient, and user-centric public transport operations.

The successful deployment and validation of this system demonstrates the feasibility of machine learning-driven transportation analytics. With appropriate investment in data infrastructure, computational resources, and user interface design, transit agencies can significantly improve service reliability and commuter satisfaction. The methodology presented here provides a reproducible framework adaptable to diverse urban contexts, contributing to the broader goal of intelligent, responsive, and sustainable urban mobility systems.

Acknowledgment

The authors express sincere gratitude to Mrs. T. Amruthavalli, Associate Professor, Department of CSE (AI&ML),

for her invaluable guidance, mentorship, and continuous support throughout this research. We thank Dr. M. Lavanya, Head of Department, for providing necessary resources, infrastructure support, and encouragement for this work. We acknowledge Sri Venkateswara College of Engineering and Technology (Autonomous) for providing excellent research facilities, computational resources, and academic environment conducive to research. We also thank the anonymous reviewers for their constructive feedback and insightful suggestions that significantly improved the quality and clarity of this paper. Finally, we acknowledge the transit authorities and data providers whose cooperation made this research possible.

REFERENCES

[1] L. Vanajakshi and S. C. Subramanian, "Prediction of bus travel time under heterogeneous traffic conditions using artificial neural networks," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2034, pp. 103–111, 2007.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[3] M. Hofmann and M. O'Mahony, "The impact of adverse weather conditions on urban bus performance," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 178–193, 2019.

[4] D. Cottrill, S. Derrible, and F. C. Pereira, "Machine learning for urban mobility: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3219–3240, 2019.

[5] B. Yu, W. H. K. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop using GPS data," *Transportation Research Part E: Logistics and Transportation Review*, vol. 38, no. 6, pp. 421–438, 2002.

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

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[8] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] 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.

[10] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[11] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[12] Y. Liu, Y. Zheng, Y. Liang, S. Liu, and D. S. Rosenblum, "Urban water quality prediction based on multi-task multi-view learning," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2576–2582.

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.