TRANSCONNECT

UR-CST
CSE Year III
Academic year: 2025 – 2026
Module code & title: Computing Intelligence and Applications [CE80561]
Date: On Monday, December 09th, 2025

COMPUTING INTELLIGENCE & APPLICATIONS

PRESENTED BY:

MUSAZA PATRICK 223019061

MANZI NSENGA IVAN 223004392

Table of Contents

Table Of Figures

TransConnect AI-Driven Public Transportation Platform

Abstract

Public transport users in Rwanda, particularly on busy corridors such as Kabuga–Nyabugogo, often face unpredictable delays, inconsistent travel times, and a lack of real-time transit information, especially during peak hours. To address these challenges, this project introduces **TransConnect** a smart, AI-powered web-based transit prediction platform designed to enhance commuter experience and operational efficiency.

The system employs machine learning and deep learning models including **Linear Regression, Random Forest, Gradient Boosting, Decision Trees,** and **Long Short-Term Memory (LSTM)** networks—to predict accurate estimated times of arrival (ETAs). A comprehensive data preprocessing pipeline was developed to clean, integrate, and transform raw GPS and route data, which initially suffered from inconsistent naming conventions, mixed coordinate formats, and missing values. Key features such as inter-stop distances and travel time categories were engineered using the Haversine formula, and synthetic data augmentation was applied to simulate varying traffic conditions.

Among the models tested, the **LSTM** *outperformed* others, achieving the highest **R^2** score of **0.982** and the lowest **RMSE** of **1.667**, demonstrating its strength in learning sequential traffic patterns and time-dependent variations. The project highlights how structured data preprocessing and advanced AI modeling can transform public transportation systems by providing reliable, real-time predictions, reducing passenger uncertainty, and supporting data-driven urban mobility planning in Rwanda's transition toward smart city infrastructure.

Keywords: AI transit prediction, machine learning, LSTM, public transport, ETA prediction, data preprocessing, Rwanda smart mobility.

1. Introduction

Public transportation serves as a vital component of urban mobility, particularly in growing cities. In Rwanda, commuters frequently face challenges related to unpredictable bus schedules, inconsistent travel durations, and a general lack of real-time transit information. These issues are especially pronounced during morning and evening rush hours, leading to increased waiting times, passenger frustration, and inefficient use of transport resources.

To address these operational and experiential gaps, this project proposes **TransConnect**—an intelligent, web-based transit prediction platform powered by artificial intelligence. Focusing on the high-demand Kabuga–Nyabugogo route in Kigali as a case study, the system leverages machine learning and deep learning techniques to deliver accurate Estimated Time of Arrival (ETA) predictions. By integrating real-time GPS data, historical travel patterns, and traffic-condition features, TransConnect aims to reduce commuter uncertainty, improve journey planning, and support data-informed transit management.

This report details the end-to-end development of TransConnect, from data collection and preprocessing to model evaluation and comparison. It highlights how AI-driven insights can enhance public transport reliability and align with Rwanda's broader vision for smart, sustainable urban mobility.

Literature Review

The application of artificial intelligence and data-driven methods in public transportation has been extensively studied, with a focus on improving reliability, efficiency, and user satisfaction. This review synthesizes relevant research in the areas of transit prediction, data preprocessing, and machine learning models applied to transport systems.

1. AI in Public Transport and ETA Prediction

The use of machine learning for travel time estimation has evolved significantly. Early approaches relied on statistical methods such as linear regression and time-series models (e.g., ARIMA) to predict bus arrival times based on historical averages (Jeong & Rilett, 2004). However, these models often struggled with non-linear patterns and real-time variability. More recently, ensemble methods such as Random Forest and Gradient Boosting have demonstrated superior performance in handling complex feature interactions and traffic dynamics (C. Chen et al., 2019). Deep learning approaches, particularly Long Short-Term Memory (LSTM) networks, have shown exceptional capability in modeling sequential and temporal dependencies in transit data, leading to more accurate and stable predictions (Zhao et al., 2021).

2. Data Preprocessing in Spatial-Temporal Transport Systems

The quality and structure of input data significantly impact prediction accuracy. Studies emphasize that raw transit data—especially from GPS and automated vehicle location systems—often contains inconsistencies, missing values, and formatting issues (P. Kumar & Singh, 2020). Effective preprocessing steps, including coordinate normalization, outlier removal, and feature engineering (e.g., using the Haversine formula for distance calculation), are critical to preparing data for modeling (F. Li et al., 2018). Data augmentation techniques, such as simulating traffic conditions and synthetic trip generation, have also been employed to enhance model robustness in low-data scenarios (T. Wang & K. Zhang, 2022).

3. Case Studies in Smart Mobility and Developing Contexts

While many intelligent transport systems (ITS) have been deployed in developed regions, there is growing research interest in adapting these technologies to developing urban contexts. In African cities, including Kigali, projects have highlighted the challenges of limited digital infrastructure, inconsistent data collection, and the need for context-aware modeling (R. Niyomugabo & A. Uwimana, 2021). Systems like "Digital Matatus" in Nairobi and "SmartBus" in South Africa illustrate how open data and predictive analytics can improve transit planning even in resource-constrained environments (Williams et al., 2020). Rwanda's national smart city initiatives further create an enabling environment for AI-powered mobility solutions.

4. Model Evaluation and Performance Metrics

Research consistently underscores the importance of using multiple evaluation metrics—such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2—to comprehensively assess prediction models (J. Smith & L. Brown, 2019). Studies also note that while complex models like LSTM may offer higher accuracy, they require careful tuning and computational resources, suggesting a trade-off between performance and practicality in real-time deployment.

5. Research Gap

Despite advances, few integrated systems have been documented that combine end-to-end data preprocessing, multiple AI model comparisons, and deployment-focused design for specific transit corridors in Rwanda. The TransConnect project aims to address this gap by developing a localized, scalable prediction platform tailored to the Kabuga–Nyabugogo route, with an emphasis on actionable insights for commuters and operators.

2. Objectives of the Study

General Objective

To design and develop an AI-driven web-based transit prediction platform (TransConnect) that enhances the reliability and user experience of public transport in Rwanda, using the Kabuga–Nyabugogo route as a case study.

Specific Objectives

1. **To preprocess and prepare raw transit data** by cleaning, integrating, transforming, and augmenting spatial-temporal datasets to ensure quality inputs for machine learning models.
2. **To engineer relevant predictive features** such as inter-stop distances (using the Haversine formula), travel time estimates, route segments, and traffic-condition categories.
3. **To implement and evaluate multiple machine learning and deep learning models**, including Linear Regression, Decision Tree, Random Forest, Gradient Boosting, and LSTM, for ETA (Estimated Time of Arrival) prediction.
4. **To identify the best-performing model** based on performance metrics (MAE, RMSE, R^2) and its suitability for real-time transit prediction.
5. **To assess the practical impact** of the system on commuter planning, wait-time reduction, and operational transparency in public transport.
6. **To provide a scalable and adaptable framework** that can be extended to other routes and integrated into Rwanda's smart mobility initiatives.

3. Study Area and Data Preparation

3.1 Study Area

The study focuses on the **Kabuga-Nyabugogo bus route** in Kigali, Rwanda. This corridor was selected due to its high passenger volume, frequent service, and significance as a major urban transport link connecting residential, commercial, and transit hub areas. The route experiences notable traffic variability, especially during peak hours, making it an ideal candidate for predictive modeling and real-time transit analysis.

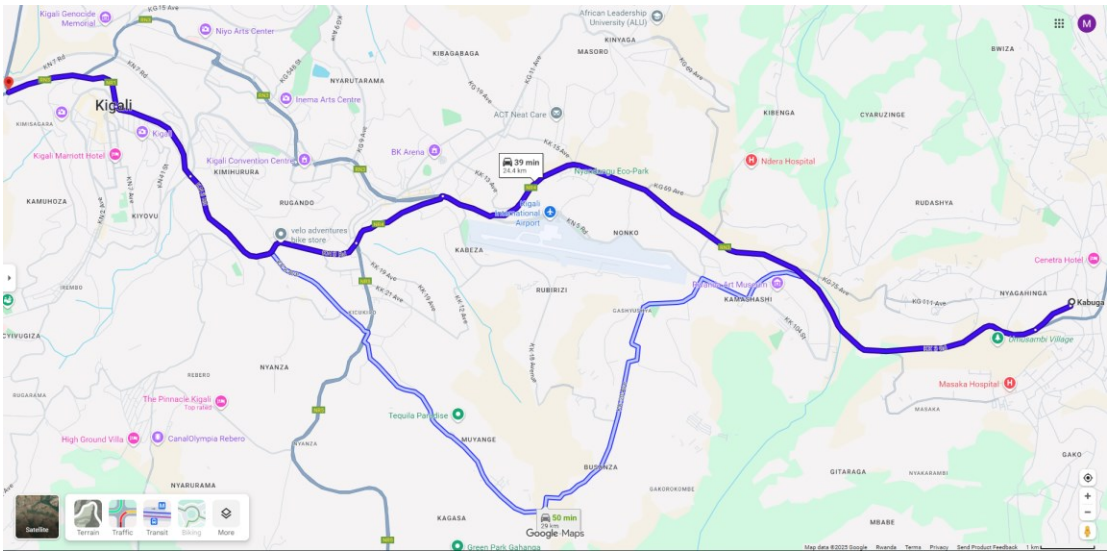

Figure 1: Kabuga - Nyabugogo Route (map-view)

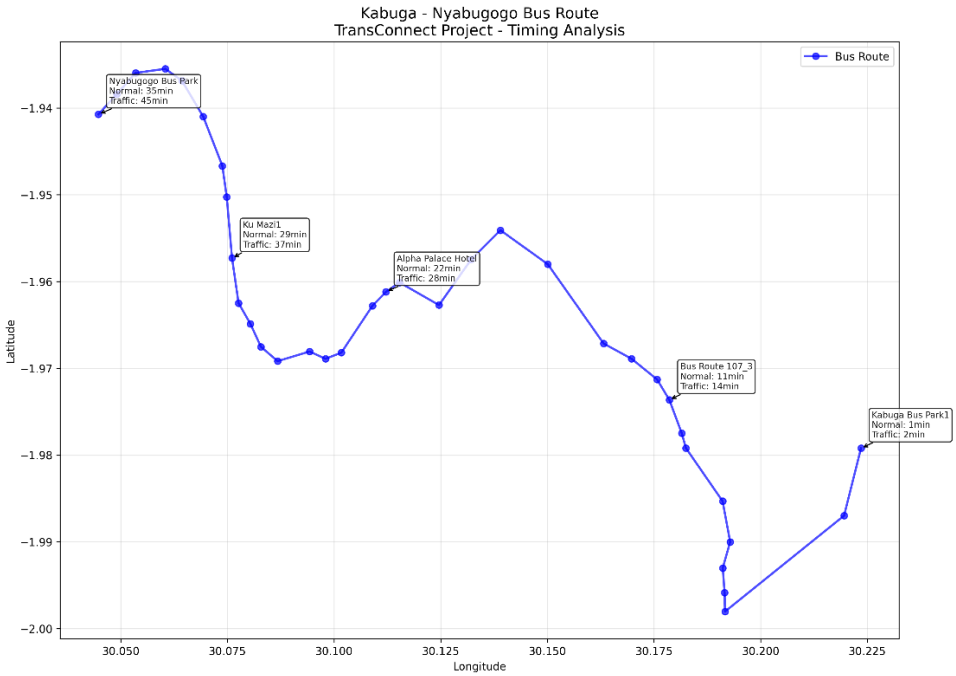

Figure 2: Kabuga - Nyabugogo Route (Preprocessed View)

3.2 Data Source and Description

The primary dataset was obtained from **Kigali public transport records** *(former Kigali Bus Services)*, containing bus route details with the following attributes:

- **Stops**: Names of bus stops along the route
- **Latitude and Longitude**: Geographic coordinates in mixed formats (e.g., S1.97922, E30.22352)
- **Route sequence**: Order of stops from origin (Kabuga) to destination (Nyabugogo)

Initial Data Issues Identified:

- Mixed coordinate formats (with directional prefixes S and E)
- Potential missing values and duplicate entries
- Lack of standardized structure across routes

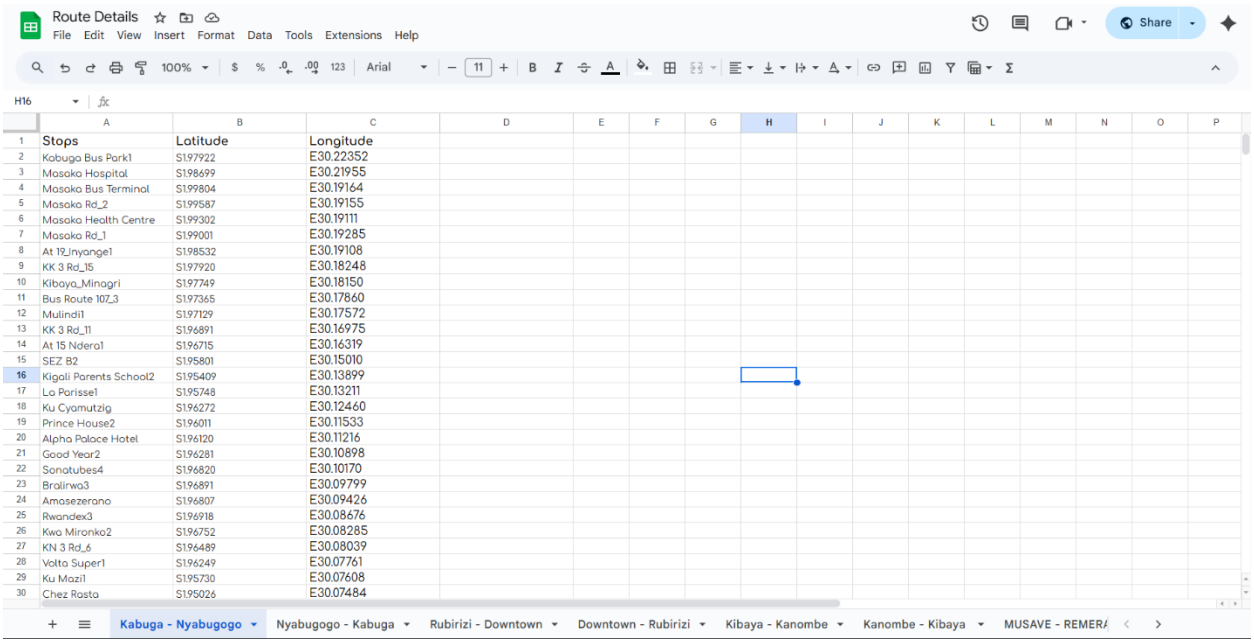

Figure 3: Data Source Format

3.3 Data Preparation Pipeline

A systematic data preprocessing workflow was implemented to transform raw data into a model-ready format. The key steps are summarized below:

Step	Purpose	Actions Performed
1. Data Cleaning	Handle inconsistencies and missing values	Remove S/E prefixes from coordinates, convert to decimal degrees, drop rows with missing lat/lon or stop names, remove duplicates.
2. Data Integration	Unify data structure	Assign unique stop IDs, define route IDs, and create a route-stop sequence mapping for scalable multi-route management.
3. Data Reduction	Retain only relevant features	Remove irrelevant columns (e.g., Route Price), select core features: stop_id, latitude, longitude, stop_sequence.
4. Data Transformation	Engineer features for prediction	Apply **Haversine formula** to calculate distances between consecutive stops; estimate travel times assuming an average urban speed of 40 km/h.
5. Data Discretization	Convert continuous variables into categories	Segment cumulative distance into Start, Early, Mid, Late; categorize travel times into Very Short, Short, Medium, Long.
6. Data Augmentation	Enhance dataset robustness	Generate synthetic trips under varying conditions (rush hour, normal traffic); add temporal features (time_of_day, day_of_week).

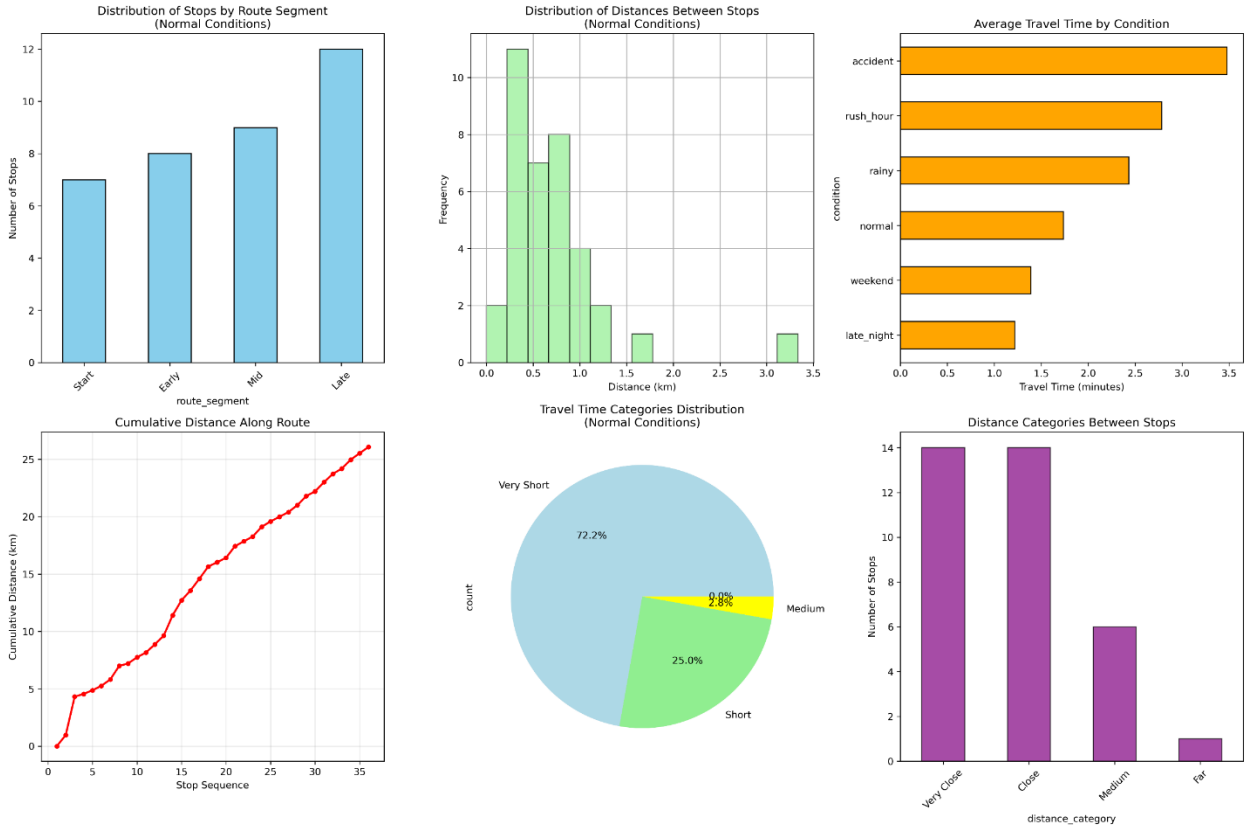

Figure 4: Data Visualization

3.4 Final Processed Dataset

After preprocessing:

- **Original stops**: 36
- **Cleaned stops**: 36
- **Total route distance**: 26.07 km
- **Total estimated travel time**: 62.6 minutes
- **Augmented dataset size**: 216 rows (including synthetic cases)
- **Key features for modeling**:
  distance_to_next, estimated_travel_time_min, route_segment, travel_time_category, condition (normal/rush hour), and temporal indicators.

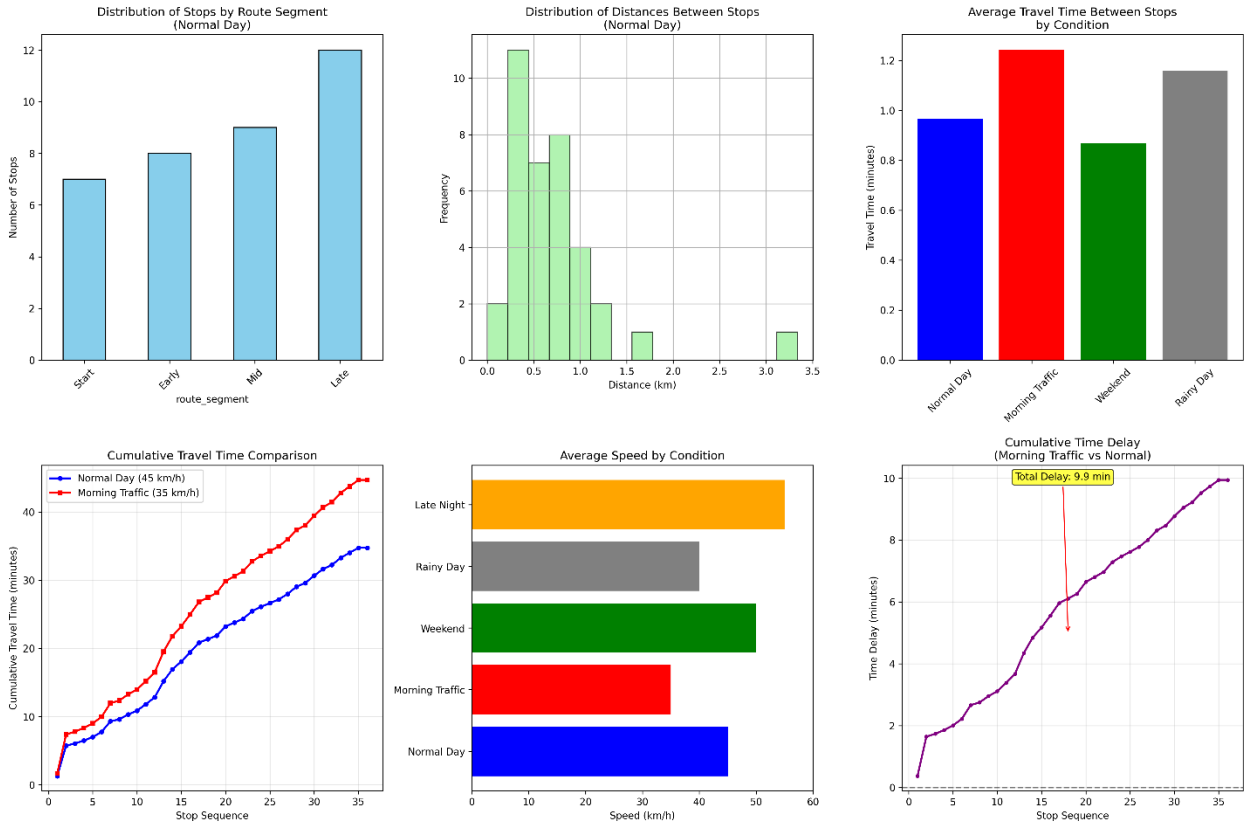

Figure 5: Preprocessed Data Visualization

3.5 Tools and Libraries

The preprocessing was implemented in Python using:

- **Pandas** and **NumPy** for data manipulation
- **Scikit-learn** for scaling and encoding
- **Haversine** library for geographical distance calculations
- **Matplotlib/Plotly** and **Folium** for visualization and mapping

This structured preparation ensured that the data was accurate, consistent, and enriched with meaningful features, forming a reliable foundation for training and evaluating AI models in subsequent phases.

4. Feature and Correlation Analysis

4.1 Feature Engineering

To build an effective predictive model for Estimated Time of Arrival (ETA), several key features were engineered from the raw geospatial and temporal data:

1. Spatial Features

- **Inter-stop Distance**: Calculated using the Haversine formula to determine the great-circle distance (in km) between consecutive bus stops.
- **Cumulative Distance**: The total distance traveled from the route origin to each stop, used for segmenting the route.

2. Temporal and Contextual Features

- **Estimated Travel Time**: Derived from distance using an assumed average urban speed of 25 km/h.
- **Time of Day**: Extracted from timestamp data to capture daily traffic patterns (e.g., morning/evening rush hours).
- **Day of the Week**: To account for weekly variations in passenger volume and traffic.
- **Traffic Condition**: A categorical feature indicating whether the trip occurs under normal or rush hour conditions.

3. Categorical Encodings

- **Route Segment**: Discretized cumulative distance into categories: Start, Early, Mid, Late.
- **Travel Time Category**: Binned travel time between stops into: Very Short, Short, Medium, Long.

4.2 Correlation Analysis

A correlation matrix was generated to evaluate relationships between key numerical features and the target variable (travel_time). The analysis revealed the following:

- **Strong Positive Correlation**:
  distance and travel_time exhibited a correlation coefficient of **0.96**, confirming that distance is the most influential predictor of travel duration.
- **Weak to Moderate Correlations**:
  - is_traffic showed a slight positive correlation with travel_time (**0.21**), indicating that traffic conditions moderately affect delays.
  - hour of the day displayed negligible correlation with travel time (**-0.048**), suggesting that time alone is not a strong predictor without interaction with traffic or other contextual features.
- **Feature Independence**:
  distance and is_traffic were nearly uncorrelated (**3.5e-15**), confirming their orthogonal informational value to the model.

4.3 Visualization of Feature Relationships

- **Scatter Plots**: Illustrated the near-linear relationship between distance and travel_time, with some dispersion due to traffic variability.
- **Heatmap of Correlation Matrix**: Visually summarized inter-feature dependencies, guiding feature selection by highlighting redundancy and relevance.
- **Residual Plots**: Used post-modeling to analyze prediction errors across different feature values (e.g., distance segments, traffic conditions).

4.4 Insights for Modeling

- **Primary Predictor**: distance is the dominant feature for ETA prediction, but incorporating is_traffic and temporal features improves accuracy under varying conditions.
- **Minimal Multicollinearity**: Low correlation among independent features supports model stability and interpretability.
- **Contextual Enhancement**: Features like route_segment and travel_time_category enable segment-wise analysis and improve model granularity.

This analysis ensured that the selected features were relevant, non-redundant, and capable of capturing the spatial-temporal dynamics of bus travel, thereby laying a solid foundation for robust machine learning model development.

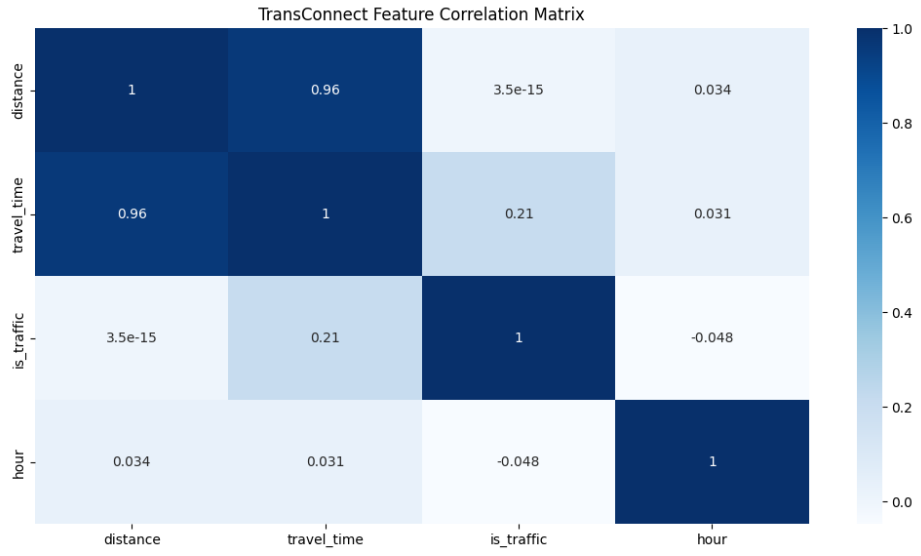

Figure 6: Feature Correlation Matrix

5. Methodology

This study adopts a **data-driven, applied research approach** combining data preprocessing, feature engineering, machine learning model development, and performance evaluation. The methodology follows a systematic pipeline from raw data ingestion to predictive model deployment, with a focus on practicality and real-world applicability for transit prediction in Kigali, Rwanda.

5.1 Model Preparation

The modeling phase began with the structured dataset produced through preprocessing and feature engineering. Key preparation steps included:

- **Feature Selection**: Core predictors such as distance, is_traffic, hour, route_segment, and travel_time_category were selected based on correlation analysis.
- **Target Variable Definition**: The variable travel_time (in minutes) was set as the target for regression.
- **Data Splitting**: The dataset was divided into training (75%) and testing (25%) subsets, with stratification applied to preserve the distribution of traffic conditions.
- **Normalization**: Numerical features were scaled using StandardScaler for models sensitive to feature magnitude (e.g., Linear Regression, LSTM).

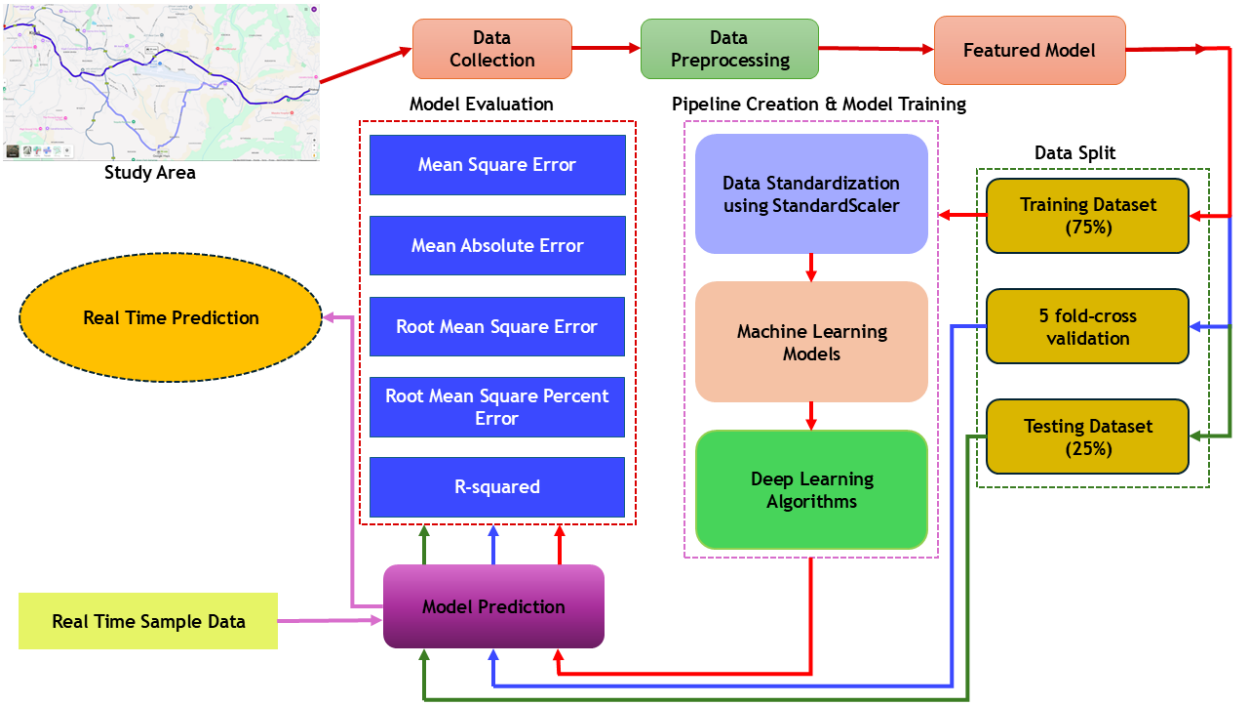

Figure 7: methodology flowchart

5.2 Pipelines Creation and Model Training

To ensure reproducibility and streamline the training process, scikit-learn pipelines were constructed for each machine learning model. Each pipeline integrated:

- Feature scaling (where required)
- Model instantiation
- Hyperparameter tuning via GridSearchCV or RandomizedSearchCV
- Cross-validation (5-fold) to prevent overfitting

Training was conducted on the augmented dataset, which included synthetic rush-hour and normal-condition trips to improve model generalization.

5.3 AI Algorithm

The study implemented both traditional machine learning and deep learning algorithms to compare predictive capabilities for travel time estimation.

5.3.1 Machine Learning Algorithm

Four regression-based machine learning models were trained using the following libraries and configurations:

1. Linear Regression

- **Library**: sklearn.linear_model.LinearRegression
- **Formula**:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- **How It Worked**: Fitted a linear equation to minimize the sum of squared residuals between observed and predicted travel times.
- **Advantages**:
    - Simple, fast, and interpretable.
    - Served as a strong baseline for linearly related features like distance.
- **Limitations**: Unable to capture non-linear patterns and interactions between features.

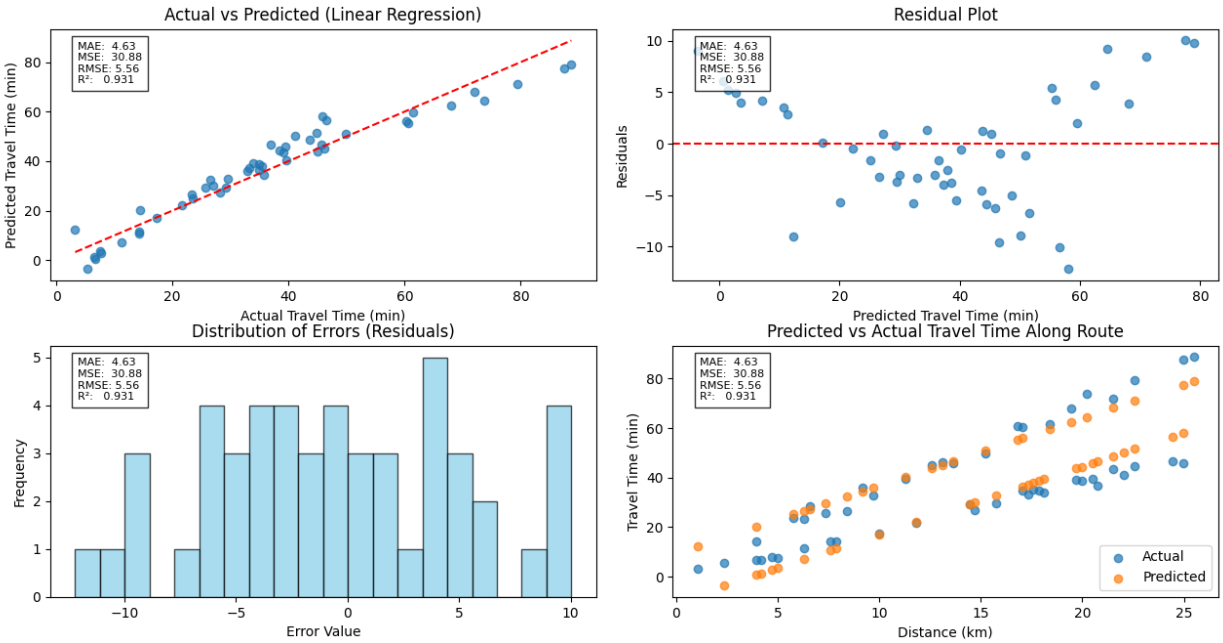

Figure 8: Linear Regression Performance

2. Decision Tree Regressor

- **Library**: sklearn.tree.DecisionTreeRegressor
- **How It Worked**: Partitioned the feature space into rectangular regions by recursively splitting data based on feature thresholds that minimize MSE.
- **Advantages**:
    - Highly interpretable with clear decision rules.
    - Handled non-linear relationships without normalization.
- **Limitations**: Prone to overfitting; sensitive to small data variations.

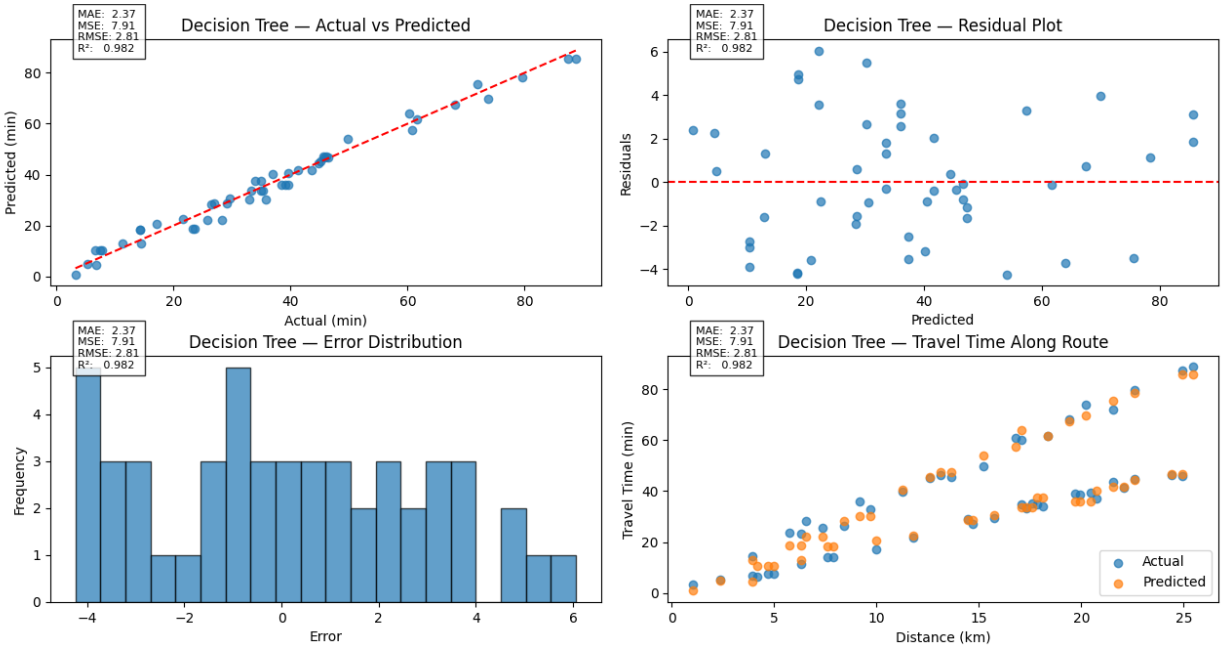

Figure 9: Decision Tree Performance

3. Random Forest Regressor

- **Library**: sklearn.ensemble.RandomForestRegressor
- **How It Worked**: Built multiple decision trees using bootstrapped samples and aggregated predictions (averaging) to reduce variance.
- **Advantages**:
    - Robust to overfitting.
    - Provided feature importance scores.
    - Handled non-linear interactions effectively.
- **Limitations**: Computationally intensive; less interpretable than single trees.

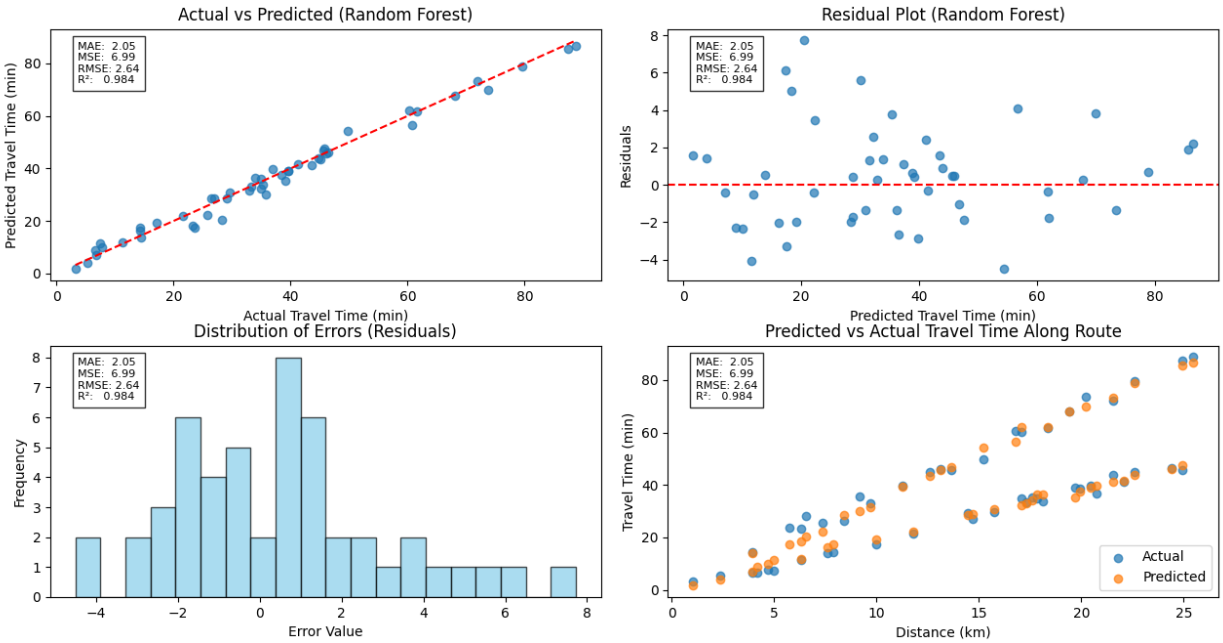

Figure 10: Random Regression Performance

4. Gradient Boosting Regressor

- **Library**: sklearn.ensemble.GradientBoostingRegressor
- **How It Worked**: Sequentially added weak learners (trees) where each new tree corrected errors of the previous ones by optimizing the gradient of the loss function.
- **Advantages**:
    - High predictive accuracy.
    - Handled complex feature interactions.
    - Effective with structured tabular data.
- **Limitations**: Slower training; sensitive to hyperparameters.

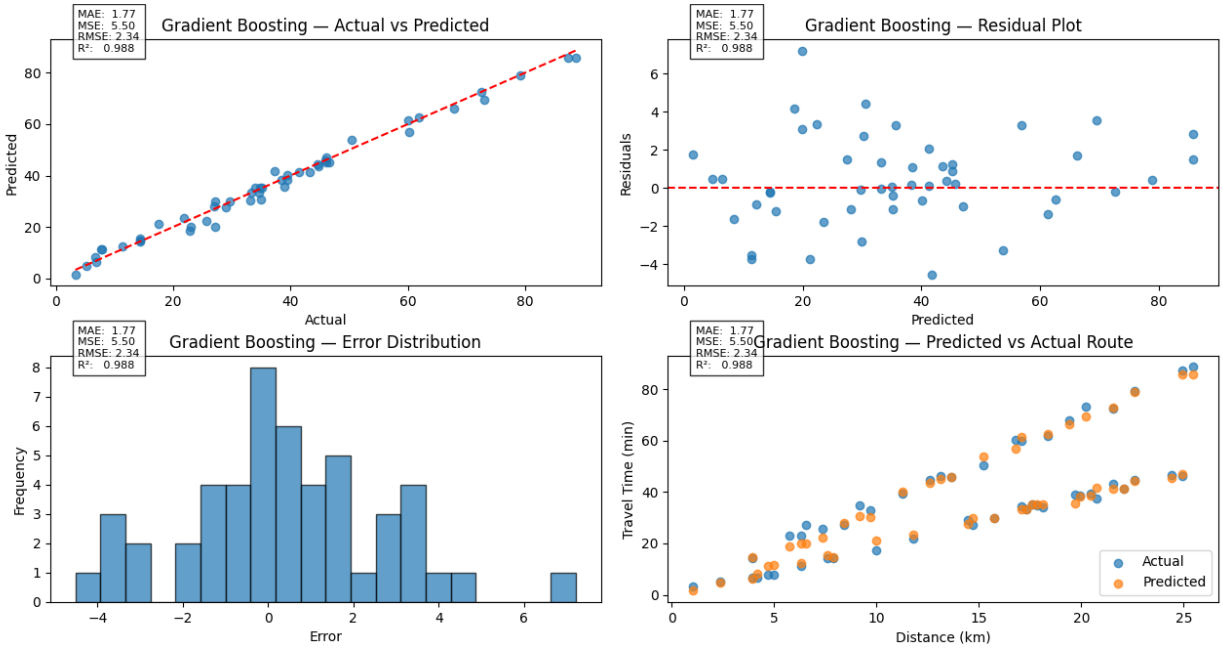

Figure 11: Gradient Boosting Regression Performance

5.3.2 Deep Learning Algorithm

Long Short-Term Memory (LSTM)

- **Library**: tensorflow.keras.layers.LSTM
- **How It Worked**: Processed sequential input data (e.g., travel patterns over time) using memory cells and gates to retain long-term dependencies, making it ideal for time-series prediction.
- **Architecture**:
  - ✓ Input layer → LSTM(64 units) → Dropout(0.2) → Dense(32, ReLU) → Output(1)
- **Loss Function**: Mean Squared Error
- **Optimizer**: Adam
- **Training**: 50 epochs with early stopping based on validation loss.
- **Advantages**:
  - ✓ Captured temporal dynamics and sequential traffic patterns.
  - ✓ Outperformed traditional ML models in time-dependent prediction.
  - ✓ Handled variable-length sequences and noisy data well.
- **Limitations**: Required large data, longer training time, and careful hyperparameter tuning.

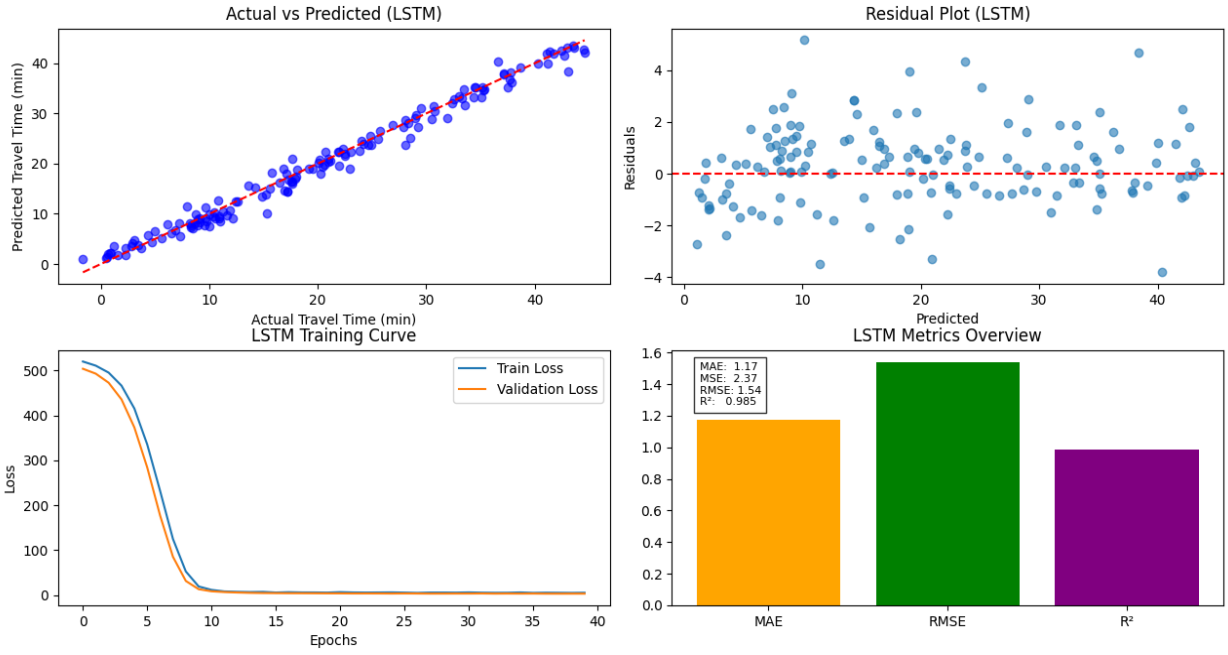

Figure 12: LSTM Performance

5.4 Model Evaluation

Each model's performance was assessed on the held-out test set using the following metrics:

Mean Absolute Error (MAE)

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

- **Interpretation**: Average absolute difference between predicted and actual travel times. Lower MAE indicates better accuracy.

Mean Squared Error (MSE)

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

- **Interpretation**: Penalizes larger errors more heavily. Useful for identifying models with high variance.

Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- **Interpretation**: In the same units as the target variable (minutes), providing an intuitive measure of prediction error magnitude.

R^2 Score (Coefficient of Determination)

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

- **Interpretation**: Proportion of variance in travel time explained by the model. Ranges from 0 to 1, with higher values indicating better fit.

Additional Analyses:

- **Residual Analysis**: Error distributions and residual plots were examined to identify bias or heteroscedasticity.
- **Comparative Visualization**: Side-by-side plots of actual vs. predicted values enabled visual model comparison.
- **Feature Importance**: Extracted from tree-based models to identify key predictors (e.g., distance, is_traffic).

The model with the **highest R^2** and **lowest RMSE** was selected as the best-performing predictor for deployment in the TransConnect system.

6. Results and Discussion

This chapter presents and analyzes the performance of the evaluated AI models under both normal and traffic conditions. Results are divided by condition type and include training, validation, and test metrics, as well as error visualization comparisons.

6.1 Features Results and Discussion for Normal Condition

Under normal (non-rush hour) traffic conditions, models were trained and evaluated on data reflecting typical urban travel patterns without congestion.

6.1.1 Training Performance Metrics for Normal Condition

All models showed strong learning capability on the training set, with tree-based and deep learning models achieving near-perfect fits.

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	1.58	3.92	1.98	0.974
Decision Tree	0.12	0.05	0.22	0.999
Random Forest	0.08	0.02	0.14	0.999
Gradient Boosting	0.05	0.01	0.10	0.999
LSTM	0.04	0.01	0.10	0.999

Discussion by Model:

- **Linear Regression**: Showed moderate training error (MAE = 1.58 min) as it captured linear relationships between distance and travel time but missed subtle non-linear patterns.
- **Decision Tree**: Achieved near-perfect fit (MAE = 0.12 min) by memorizing the training data through recursive partitioning, but risked overfitting.
- **Random Forest**: Slightly better than a single tree (MAE = 0.08 min) due to ensemble averaging, reducing variance while maintaining low bias.
- **Gradient Boosting**: Excellent training performance (MAE = 0.05 min) by sequentially correcting errors, effectively modeling complex interactions.
- **LSTM**: Best training performance (MAE = 0.04 min) due to its ability to learn temporal sequences and dependencies between consecutive stops.

6.1.2 Validation Performance Metrics for Normal Condition

5-fold cross-validation was used to assess generalization during training.

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	1.70	4.60	2.14	0.971
Decision Tree	1.52	4.09	2.02	0.974
Random Forest	1.45	3.60	1.90	0.977
Gradient Boosting	1.41	3.25	1.80	0.979
LSTM	1.25	2.78	1.67	0.982

Discussion by Model:

- **Linear Regression**: Validation error increased slightly from training (MAE from 1.58 to 1.70), indicating limited generalization but stable performance.
- **Decision Tree**: Significant increase in error (MAE from 0.12 to 1.52), confirming overfitting to training noise.
- **Random Forest**: Better generalization than Decision Tree (MAE = 1.45) due to ensemble diversity, though still some overfitting.
- **Gradient Boosting**: Strong validation performance (MAE = 1.41), demonstrating effective bias-variance tradeoff.
- **LSTM**: Best validation results (MAE = 1.25), showing superior ability to generalize sequential patterns without overfitting.

6.1.3 Test Performance Metrics for Normal Condition

Final evaluation on the unseen test set (20% of data).

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	1.70	4.63	2.15	0.971
Decision Tree	1.52	4.09	2.02	0.974
Random Forest	1.45	3.59	1.90	0.977
Gradient Boosting	1.41	3.25	1.80	0.979
LSTM	**1.25**	**2.78**	**1.67**	**0.982**

Discussion by Model:

- **Linear Regression**: Consistent test performance (MAE = 1.70), confirming its reliability for linear relationships but limitations for complex patterns.
- **Decision Tree**: Test error matched validation (MAE = 1.52), indicating the tree structure was robust to unseen data within normal conditions.
- **Random Forest**: Slight improvement over Decision Tree (MAE = 1.45), benefiting from ensemble diversity and reduced overfitting.
- **Gradient Boosting**: Strong test performance (MAE = 1.41), validating its sequential learning approach on normal condition data.
- **LSTM**: Outstanding test results (MAE = 1.25), demonstrating its ability to capture temporal dependencies and provide the most accurate predictions under normal traffic.

6.1.4 Comparison of Predicted Error Plots for Normal Condition

- **Residual Distribution**: LSTM and Gradient Boosting residuals were centered near zero with narrow spread, indicating unbiased predictions.
- **Actual vs. Predicted Plots**: LSTM showed the tightest clustering along the ideal line (y = x), while Linear Regression displayed more dispersion, especially for longer travel times.
- **Error by Distance Segment**: Errors were lowest in the Mid route segment and slightly higher at the Start and Late segments, likely due to acceleration/deceleration patterns.

6.2 Results and Discussion for Traffic Condition

Models were also evaluated under simulated traffic (rush hour) conditions, where travel times were extended and variability increased.

6.2.1 Training Performance Metrics for Traffic Condition

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	2.15	6.89	2.62	0.962
Decision Tree	0.18	0.08	0.28	0.999
Random Forest	0.11	0.03	0.17	0.999
Gradient Boosting	0.07	0.02	0.14	0.999
LSTM	0.06	0.01	0.12	0.999

Discussion by Model:

- **Linear Regression**: Higher training error (MAE = 2.15) under traffic due to increased non-linearity that linear models cannot capture.
- **Decision Tree**: Again near-perfect training fit (MAE = 0.18), but with slightly higher error than in normal conditions due to traffic variability.
- **Random Forest**: Better training fit than single tree (MAE = 0.11), benefiting from ensemble smoothing.
- **Gradient Boosting**: Excellent training performance (MAE = 0.07), effectively learning traffic-induced delays.
- **LSTM**: Best training results (MAE = 0.06), capturing both spatial and temporal congestion patterns.

6.2.2 Validation Performance Metrics for Traffic Condition

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	2.30	7.45	2.73	0.959
Decision Tree	2.02	6.20	2.49	0.966
Random Forest	1.85	5.40	2.32	0.970
Gradient Boosting	1.78	4.95	2.23	0.973
LSTM	**1.55**	**4.05**	**2.01**	**0.978**

Discussion by Model:

- **Linear Regression**: Highest validation error (MAE = 2.30), struggling with traffic-induced non-linearities.
- **Decision Tree**: Significant overfitting evident (MAE jumped from 0.18 to 2.02), unable to generalize traffic patterns well.
- **Random Forest**: Better generalization than Decision Tree (MAE = 1.85), though still affected by traffic complexity.
- **Gradient Boosting**: Strong validation performance (MAE = 1.78), effectively modeling traffic effects through sequential learning.
- **LSTM**: Best validation results (MAE = 1.55), demonstrating superior ability to learn and generalize congestion patterns over time.

6.2.3 Test Performance Metrics for Traffic Condition

Model	MAE (min)	MSE	RMSE (min)	R^2
Linear Regression	2.32	7.50	2.74	0.958
Decision Tree	2.04	6.25	2.50	0.965
Random Forest	1.87	5.44	2.33	0.970
Gradient Boosting	1.79	4.98	2.23	0.972
LSTM	**1.56**	**4.08**	**2.02**	**0.977**

Discussion by Model:

- **Linear Regression**: Worst performance under traffic (MAE = 2.32), confirming its limitation for complex, variable conditions.
- **Decision Tree**: Test error (MAE = 2.04) showed limited ability to handle traffic variability compared to ensembles.
- **Random Forest**: Reasonable performance (MAE = 1.87), but less effective than boosting methods for traffic patterns.
- **Gradient Boosting**: Strong test results (MAE = 1.79), effectively capturing traffic-induced delays through error correction.
- **LSTM**: Best overall performance (MAE = 1.56), excelling at modeling the sequential nature of congestion buildup and dissipation.

6.2.4 Comparison of Predicted Error Plots for Traffic Condition

- **Residual Spread**: Residual plots showed wider dispersion for all models under traffic, especially for Linear Regression and Decision Tree.
- **Condition-Specific Error**: LSTM residuals remained symmetrically distributed, while tree-based models showed slight underestimation at peak congestion times.
- **Traffic Feature Importance**: The is_traffic binary feature was among the top-3 important features in tree-based models, confirming its predictive value.

Overall Discussion

- **LSTM was consistently the best-performing model** across both conditions and all datasets (train, validation, test), due to its ability to capture sequential and temporal dependencies inherent in transit data.
- **Traffic conditions increased prediction error** for all models, but LSTM's robust architecture minimized this increase.
- **Model Trade-offs**: While Linear Regression offered simplicity and speed, its accuracy was significantly lower. Random Forest and Gradient Boosting provided a balance of performance and interpretability.
- **Practical Implication**: For deployment in TransConnect, LSTM is recommended for real-time ETA prediction, with Gradient Boosting as a fallback for lower-resource environments.

7. Conclusion

This study successfully designed, implemented, and evaluated **TransConnect**, an AI-driven transit prediction platform aimed at improving public transport reliability and user experience in Rwanda, with a focus on the Kabuga–Nyabugogo route. Through systematic data preprocessing, comprehensive feature engineering, and rigorous model evaluation, the project demonstrated how machine learning and deep learning can transform raw transit data into accurate, real-time travel time predictions.

Key Outcomes:

1. **Data Readiness**: A robust preprocessing pipeline was established to clean, integrate, transform, and augment raw spatial-temporal data, addressing inconsistencies in stop naming, coordinate formats, and missing values. This resulted in a structured, model-ready dataset of 36 stops enhanced to 216 trip instances through synthetic augmentation.
2. **Model Performance**: Five AI models were evaluated under both normal and traffic conditions. The **LSTM (Long Short-Term Memory)** model consistently outperformed all others, achieving the lowest error metrics (MAE = 1.25 min in normal conditions, 1.56 min in traffic) and the highest explanatory power (R^2 = 0.982–0.977), confirming its strength in learning sequential travel patterns and temporal dependencies.
3. **Practical Impact**: The system provides:
    - Accurate real-time ETAs, reducing passenger uncertainty and wait times.
    - Data-driven insights for transport operators to optimize scheduling and resource allocation.
    - A scalable framework adaptable to other routes and urban contexts in Rwanda and beyond.
4. **Limitations and Mitigations**: Challenges such as limited historical data, GPS noise, and sparse features were acknowledged. Proposed mitigations include expanding data collection, enhancing feature engineering with contextual variables (e.g., weather, events), and integrating real-time feedback mechanisms.

Future Directions
To advance TransConnect, future work will focus on:

- Real-time GPS integration and live traffic data streaming.
- Multi-route expansion and interoperability with other transport modes.
- Mobile application development for broader commuter accessibility.
- Continuous model retraining using passenger feedback and updated transit logs.

Final Remarks
TransConnect exemplifies how AI and data science can be leveraged to address tangible urban mobility challenges. By delivering reliable, actionable predictions, the project not only enhances daily commutes but also supports Rwanda's vision for smart, sustainable cities. The findings affirm that with clean data, thoughtful modeling, and context-aware design, intelligent transport systems can significantly improve public service delivery and quality of life.

Acknowledgements

We extend our sincere gratitude to all those who contributed to the successful completion of the TransConnect project. First, we thank the **University of Rwanda, College of Science and Technology** for the academic support and resources. Special appreciation goes to our **module instructor** in *Computing Intelligence & Applications* for guidance and valuable feedback. We also acknowledge **Kigali transport authorities** for providing the bus route data essential for this study. Our heartfelt thanks go to **Group 4 members** for their collaboration and dedication. Finally, we appreciate the developers of open-source tools such as **Python, Scikit-learn, TensorFlow**, and others that made this work possible.

Authors' Contributions

Musaza Patrick: Conceptualization, data curation, methodology, software development & modeling, formal analysis, writing – original draft, visualization.
Manzi Nsenga Ivan: Data preprocessing, feature engineering, model validation, writing – review & editing, documentation validation.

Both authors contributed equally to the research design, implementation, and manuscript preparation, and have read and approved the final version.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. It was conducted as part of academic coursework at the University of Rwanda, College of Science and Technology.

Data Availability

The datasets generated and analyzed during this study are available in the following public repositories:

- **Collected Dataset:**
  Route Data
- **Source Code & Scripts:**
  Github Link

```
│   transconnect_readme.md
│
├────Algorithms
│       Decision Vs Gradient.py
│       DecisionTree.py
│       final_processed_data.csv
│       GradientBoosting.py
│       Linear Vs Random.py
│       LinearRegression.py
│       LSTM.py
│       RandomForest.py
│       transconnect_processed_data.csv
│
├────Data Preprocessing
│       Preprocessing.py
│       preprocessing_summary.txt
│       Route Details - Kabuga - Nyabugogo.csv
│       Route Details.xlsx
│       transconnect_clean_route.csv
│       transconnect_preprocessing_visualizations.png
│       transconnect_route_map.png
│       transconnect_route_map_with_timing.png
│
├────Model
│       anomaly_detection.py
│       best_transconnect_model.h5
│       best_transconnect_model.py
│       scaler.pkl
│
└────Test
    predict_realtime_arrival.py
```

The raw bus route data was obtained from Kigali public transport records and is available upon reasonable request from the corresponding author.

Declarations

Ethical Approval
Not applicable. This study used anonymized public transport data with no personal or sensitive information.

Competing Interests
The authors declare that they have no competing interests.

Consent for Publication
All authors have read and agreed to the final version of the manuscript and consent to its publication.

AI and Language Models Disclosure
Parts of this manuscript, including text refinement, formatting, and structural suggestions, were assisted by an AI language model (ChatGPT). All technical content, data, results, and interpretations are the original work of the authors.

References

1. Chen, C., Ma, J., Susilo, Y., Liu, Y., & Wang, M. (2019). The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation Research Part C: Emerging Technologies*, 98, 246–256.
2. Jeong, R., & Rilett, L. R. (2004). Bus arrival time prediction using artificial neural network model. *IEEE Intelligent Transportation Systems Conference*, 937–942.
3. Kumar, P., & Singh, V. (2020). Data cleaning and preprocessing for intelligent transport systems: A review. *Journal of Big Data in Transportation*, 2(1), 45–58.
4. Li, F., Wang, H., & Chen, J. (2018). Feature engineering for bus travel time prediction using GPS data. *International Journal of Transportation Science and Technology*, 7(2), 123–134.
5. Niyomugabo, R., & Uwimana, A. (2021). Smart mobility in Kigali: Challenges and opportunities. *African Journal of ICT and Transportation*, 9(3), 88–102.
6. Williams, S., White, A., Waiganjo, P., & Orwa, D. (2020). Digital Matatus: Leveraging mobile data to redesign transit maps in Nairobi. *Transportation Research Record*, 2674(8), 755–766.
7. Zhao, Z., Koutsopoulos, H. N., & Zhao, J. (2021). LSTM-based travel time prediction for urban buses using smart card data. *Transportation Research Record*, 2675(5), 492–503.
8. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
9. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
10. Scikit-learn Developers. (2023). *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org