Metropolitano Plus: A Machine Learning-Based Mobile Application for Predicting Bus Arrival Times in the Corredor Metropolitano of Lima

Deneb Traverso
Faculty of Information Systems Engineering, Universidad Peruana de Ciencias Aplicadas, San Isidro, Lima, Peru
u202011093@upc.edu.pe

Gonzalo Pacheco
Faculty of Information Systems Engineering, Universidad Peruana de Ciencias Aplicadas, San Isidro, Lima, Peru
u202014801@upc.edu.pe

Sandra Wong-Durand
Faculty of Information Systems Engineering, Universidad Peruana de Ciencias Aplicadas, San Isidro, Lima, Peru
pcsiswon@upc.edu.pe (corresponding author)

Pedro Castaneda
Faculty of Systems Engineering and Electrical Mechanics, Universidad Nacional Toribio Rodriguez de Mendoza, Amazonas, Peru
pedro.castaneda@untrm.edu.pe

Alejandra Onate-Andino
Escuela Superior Politecnica de Chimborazo (ESPOCH), Riobamba, Ecuador
monate@espoch.edu.ec

Received: 17 September 2025 | Revised: 10 November 2025, 4 December 2025, 15 December 2025, 13 January 2026, and 18 January 2026 | Accepted: 21 January 2026

DOI: https://doi.org/10.48084/etasr.14855

ABSTRACT

This study aimed to enhance the efficiency and reliability of Lima's Metropolitan Bus system by applying machine learning to predict bus arrival times and support data-driven operational management. T-RAPPI is a predictive model based on the Random Forest algorithm, trained with historical operational data from the Corredor Metropolitano. The model achieved high predictive accuracy ($R^2 = 0.9998$, MAE = 0.0062 min), demonstrating its ability to reproduce real operational patterns. These predictions were integrated into the Metropolitano Plus mobile application, developed with Flutter and Firebase, which provides real-time bus arrival forecasts, station occupancy visualization, and trip evaluation features. By improving information reliability and reducing passenger waiting times, the proposed solution enhances both user experience and operational efficiency. A user validation survey based on the ISO/IEC 25010 quality standard reported satisfaction levels above 88% across all quality dimensions. Future work will focus on incorporating real-time traffic data and expanding the system to other public transport networks in Lima and similar urban contexts in Latin America.

Keywords-machine learning; random forest; bus arrival prediction; mobile application; intelligent transportation systems; public transit in Latin America

I. INTRODUCTION

Traffic congestion in Metropolitan Lima is among the most severe in Latin America, generating an average delay of 24 minutes for every 10 kilometers traveled [1]. This situation worsens during peak hours, reaching an average of 33 minutes per kilometer. The public transport system, particularly the Corredor Metropolitano, faces structural and operational challenges such as insufficient buses, long queues, and disorganization at stations [2]. Similar inefficiencies are observed in other Latin American capitals, including Bogotá, Mexico City, and Rio de Janeiro, where inadequate infrastructure and limited management tools constrain the efficiency of public transport [3]. The absence of advanced technological mechanisms within the Metropolitano system restricts its ability to effectively manage passenger flow, thereby increasing waiting times, reducing service quality, and limiting operational productivity [4].

In recent years, several Latin American cities—such as Bogotá and Mexico City—have implemented technological solutions, including mobile applications and real-time monitoring systems, to improve the reliability and efficiency of public transport [5]. Applications such as TransMilenio (Colombia) and Transantiago (Chile) have emerged as regional references, while global platforms such as Moovit operate in more than 3,400 cities across 112 countries [6]. However, despite their success, many of these systems lack advanced fleet management and user personalization capabilities. Transantiago, for example, does not integrate adaptive control mechanisms, and Moovit provides only estimated data rather than real-time precision, often resulting in user frustration and reduced adoption levels.

Beyond the regional context, recent studies have demonstrated that Machine Learning (ML) techniques can significantly enhance the predictive and operational capabilities of public transportation systems. In [7], millions of subway smart-card transactions were analyzed, finding that Random Forest outperformed SVM and regression methods in predicting passenger demand. In [8], a deep multi-graph learning model was developed to predict ride-hailing demand in New York, integrating spatial and functional dependencies for greater accuracy. In [9], Random Forest was applied to rural Switzerland, explaining 25% of the variability using demographic and spatial factors. Similarly, in Lisbon [10], a 92.84% estimation accuracy of passenger drop-offs was achieved using smart-card data, while in [11] it was shown that combining trip-planner and boarding-record data reduced MAE by 50% compared to trend-based models. Other works have focused on improving operational efficiency and route planning through ML. The study in [12] evaluated the robustness of public transport schedules using ML methods, while in [13] incremental learning was applied to detect congestion, reducing waiting times. In [14], ensemble models, such as Random Forest, were validated for metro ridership prediction. Likewise, in [15, 16], frequency adjustments and deep learning, respectively, were employed to enhance passenger-flow forecasting. In [17], ML and IoT were integrated to optimize traffic control in real time, demonstrating the adaptability of intelligent systems for smart cities.

From a user-centered perspective, ML has also been applied to measure and predict quality perception. In [18], passenger satisfaction was modeled using GPS and mobile app data, while Du-Bus [19] is a multi-source system that predicts bus waiting times with a mean absolute error of 0.78 minutes. In [20], ML applications in rail systems were reviewed, highlighting advances in automation but identifying a lack of empirical validation studies. In [21], more than two million social-media posts in Turkey were analyzed, extracting service quality indicators such as reliability and safety based on passenger feedback. Furthermore, in [22], predictive accuracy was improved by incorporating drivers' behavioral similarities, in [23], ML methods used in transport prediction were categorized, and in [24], stochastic decision-making strategies were developed for transit networks.

Despite these advances, most prior studies rely on continuous sensor data (GPS, AFC, or smart cards) from large-scale metropolitan systems and rarely address contexts with limited real-time infrastructure, such as Lima's Metropolitano. In addition, few works integrate ML models directly into mobile applications to support both passengers and system operators.

To address these limitations, this study introduces T-RAPPI, a predictive model based on the Random Forest algorithm trained with historical operational data on bus arrivals and departures. The model is integrated into the Metropolitano Plus mobile application—developed with Flutter and Firebase—which provides functionalities such as real-time bus arrival prediction, station occupancy visualization, and trip evaluation. This dual focus on predictive analytics and user-centered design contributes to improving both operational efficiency and passenger experience. In addition, the application was validated with users according to the ISO/IEC 25010 software quality standard, ensuring reliability and satisfaction. In more detail, this study sought to answer the following research questions:

- Can an ML model based on Random Forest accurately predict bus arrival times in Lima's Metropolitano system using historical operational data?
- Can the integration of such a predictive model into a mobile application improve user satisfaction and operational efficiency?

Based on these questions, the following hypotheses were proposed:

- H_1: The T-RAPPI model, trained on historical data, can predict bus arrival times with high accuracy (R^2>0.95).
- H_2: Integrating the predictive model into the Metropolitano Plus mobile app enhances user satisfaction (≥85%) and perceived system reliability.

This paper is organized as follows: Section II presents the design and implementation of the system architecture, dataset, and interfaces; Section III describes the evaluation and the results; Section IV discusses the findings and implications; and Section V concludes with recommendations and future research directions.

II. SYSTEM DESIGN

A. Architecture

The architecture of the Metropolitano Plus application integrates the T-RAPPI model, based on Random Forest, in a mobile application that allows real-time orientation of the service and user interaction with the Metropolitano system (see Figure 1). The app was developed with Flutter and the Dart language for Android devices. This integrated design allows for real-time estimates of bus arrivals, as well as end-user-oriented functionalities, all on a scalable infrastructure.

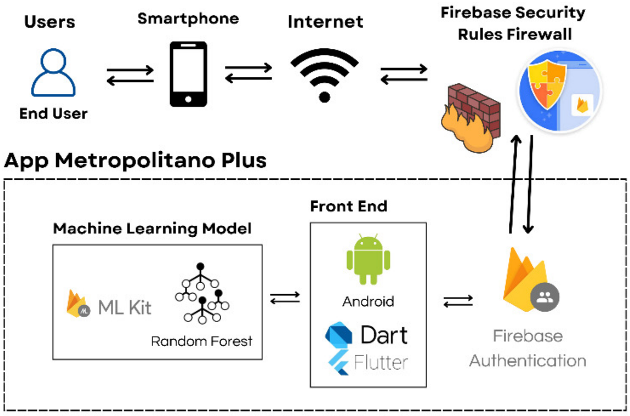


Fig. 1. Architecture of the Metropolitano Plus app [25].

1) Main Components

- Presentation Layer: Flutter/Dart to build the adaptable graphical interface according to the type of user (role).
- Business Logic Layer: Flow control, functionality management, model integration, and operational rules.
- Data Layer: Firebase (Authentication, Firestore, ML Kit, Hosting) as a scalable backend.

2) Structure of the Integrated Architecture

- User Roles: The application supports two differentiated roles, Regular Users and Service Guides. Each role has access to specific functionalities. Users can register and sign in using email/password or Google Sign-In, and their roles are managed in Firebase Firestore.
- Metropolitano Plus App: Includes features such as My Routes, Station Status, Trip Rating, and Bus Arrival Prediction. It is the main point of interaction, presenting information processed by the model and cloud services.
- Flutter/Dart: The framework used for the development of the application's user interface, which stands out for its fluid and efficient navigation through the different sections of the app.
- T-RAPPI model: The ML model deployed thanks to the services of Firestore allows analyzing the records of departures and arrivals of Metropolitano buses to make predictions of upcoming arrivals.
- Firebase Cloud Firestore: Stores all application information in the cloud.
- Firebase Authentication: Allows the management of user authentication for the app.
- Firebase ML Kit: Enables integration of the trained model within the mobile app, facilitating real-time inference.
- Firebase Hosting (optional for web): Although the app is mobile, Firebase Hosting is contemplated to host an administration panel or a future web dashboard.

B. Methodology

The design of the Metropolitano Plus system was carried out following a structured data-driven methodology, which included data collection, preprocessing, model training, and integration with the mobile interface.

1) Dataset

In order to obtain the information used for the dataset, a request was made through the ATU's Transparency Portal. The dataset covers the period from January 1 to December 31, 2023, selected because it represents the first complete year of stable post-pandemic operation of the Corredor Metropolitano, with standardized routes, frequencies, and service schedules. Using this period ensures consistency in operational behavior and allows the model to learn from representative data that reflect regular transport demand and performance conditions.

The dataset includes details of scheduled and actual arrival and departure times of buses at each station, frequencies and schedules per line, and service patterns differentiated by day of the week and time slot. This database was essential for training the model with variables representative of the system's operational behavior.

The preprocessing stage involved several key steps. First, missing or inconsistent records were removed, and outliers generated by registry errors or exceptional events were filtered out. Next, categorical variables, such as line, type of day, and season, were transformed using one-hot encoding to enable numerical processing. Continuous variables, including arrival intervals and frequencies, were normalized to ensure consistent learning. In addition, a temporal indexing variable was created to help the model capture behavioral patterns based on the time of day and whether the period corresponded to peak or off-peak hours. 70% of the dataset was used for model training, while the remaining 30% was reserved for testing and evaluation using Accuracy and Mean Squared Error (MSE). Finally, a 5-fold cross-validation scheme was applied to obtain robust estimates and minimize dependency on specific partitions.

It is worth noting that the training environment used for the model consisted of a Google Colab free-tier instance, operating on a CPU-only configuration (no GPU). Although this setup limits computational speed and parallel processing capacity, it demonstrates that the T-RAPPI model can be efficiently implemented in low-resource environments, which is advantageous for scalability and replication in similar public institutions or research settings.

The dataset used in this study was obtained from public institutional sources (ATU Transparency Portal) and does not contain personal or sensitive information. In compliance with ethical research practices, the dataset is available from the corresponding author upon reasonable request. The use of public institutional data without personal identifiers follows established ethical and reproducibility practices in applied ML research [26, 27].

2) Model

The core component of this proposal is the T-RAPPI model, specifically designed to predict the arrival times of Metropolitano buses at different stations. This model was developed using the Random Forest algorithm. Ensemble-based algorithms, such as Random Forest, have demonstrated high predictive reliability in diverse engineering and transport applications [28].

The development process followed a structured sequence of steps: data preprocessing, feature engineering, model training, and deployment. The cleaned dataset was enriched with engineered features to better capture operational dynamics and temporal dependencies, such as historical delays and station load patterns. Categorical encoding, normalization, and outlier handling were performed according to standard ML preprocessing pipelines designed to minimize bias and variance during training [29-31].

a) Input Features

The T-RAPPI model was trained with the input variables shown in Table I, which represent the main operational and contextual attributes influencing bus arrival times. The target variable (y) corresponds to the actual arrival time (in minutes), predicted based on the above operational parameters.

TABLE I. INPUT FEATURES

Resource	Specifications Used
Direction	Direction of travel (North or South).
Day Type	Type of day (weekday, Saturday, or Sunday).
Service Code	Bus service or route identifier (e.g., Regular, Express, Super Express, Night).
Visual Occupancy	Estimated station occupancy level at the time of prediction.
Day of Date	Day of the month extracted from the operational record.
Month of Date	Month of the year extracted from the operational record.
Trip Duration	Scheduled duration of the bus trip according to the operational timetable.
Service Category	Type or category of service (e.g., Regular A, B, C, Express 1–8).
Scheduled Hour	Scheduled departure or arrival time at the selected station.
Final Station Scheduled Hour	Scheduled arrival time at the final station of the route.

b) Model Structure and Mathematical Representation

The Random Forest algorithm constructs multiple independent decision trees ($T_1, T_2, \dots, T_n$) trained on random subsets of the dataset and features. For each input vector $x = (x_1, x_2, \dots, x_{10})$, the final prediction is obtained as the average of the outputs of all trees:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \qquad (1)$$

where $T_i(x)$ represents the prediction of the i^{th} three and $\hat{y}$ is the estimated bus arrival time.

Model tuning was carried out through a grid search and 5-fold cross-validation strategy to ensure optimal generalization. The selected hyperparameters were:

- *n_estimators*: 200 (number of trees)
- *max_depth*: 10 (depth limit for overfitting control)
- *min_samples_split*: 4 (minimum number of samples per split)
- *max_features*: $\sqrt{M}$, where M is the total number of features.

c) Integration and Real-Time Inference

After training and validation, the final model was exported in a Firebase ML Kit-compatible format for mobile deployment. The model was integrated within the Metropolitano Plus app, developed in Flutter, enabling real-time inference directly from the user interface. The integration of ML prediction into a mobile platform aligns with current trends in intelligent transportation and smart-city infrastructures [32-34].

Each time a user requests an arrival-time estimate, the app gathers the operational inputs (e.g., Direction, Service Code, Scheduled Hour, Visual Occupancy, etc.) and sends them to Firebase, where the model computes the predicted arrival time as follows:

$$Estimated\ Arrival\ Time = f(Features) \qquad (2)$$

The output is immediately returned to the application, allowing users to visualize the expected waiting time and enabling operators to monitor route performance. This architecture ensures both real-time feedback and reproducibility, even under limited computational resources.

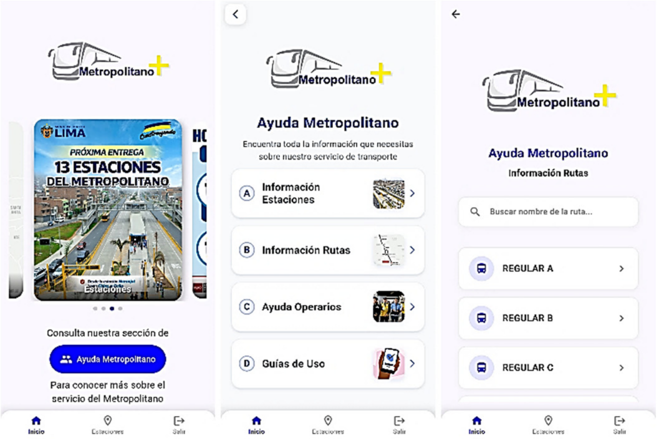
3) Interfaces

Interfaces are the user interaction layer and are key to connecting the T-RAPPI model with the app's functionalities (see Figure 2). Three main screens were prioritized:

1. Login and onboarding module: Shows the start and authentication screens, allowing users to access the system either through registered credentials or Google Sign-In. This module ensures secure access to personalized functionalities such as route prediction, station selection, and travel history.
2. Prediction and station-monitoring module: Displays the list of available stations and the selected station's detailed information. For each station, users can view bus routes, estimated arrival times generated by the T-RAPPI model, and the current occupancy level. These real-time indicators help passengers plan their trips efficiently and reduce waiting times.
3. Assistance and route-information module: Presents the in-app help center and route visualization screens. Users

can access notifications, tutorials, and information about the different service types (regular, express, and special routes), improving usability and engagement.

Together, these components illustrate how Metropolitano Plus integrates predictive analytics and user-centered design to provide reliable, real-time public-transport information for the Corredor Metropolitano system.



Consulta nuestra sección de Ayuda Metropolitano Para conocer más sobre el servicio del Metropolitano

Encuentra toda la información que necesitas sobre nuestro servicio de transporte

Fig. 2. Graphical interface of the Metropolitano Plus app: Station information screen.

4) *Training*

The T-RAPPI model was trained in the Google Colab environment, using the free plan that offers limited support in terms of computational resources, but sufficient for medium-scale models (see Table II). The Python language was used in conjunction with the scikit-learn library, which provides a robust implementation of the RF algorithm.

TABLE II. ENVIRONMENT CONFIGURATION

Resource	**Specifications used**
RAM	1.5 GB of 12.7 GB available
Disk	32.5 GB to 107.7 GB
GPU	No used (CPU mode)

The selection of optimal parameters was made using a Grid Search schema in combination with 5-fold cross-validation, which allowed evaluating the performance of the model in multiple subsets of data and reducing the risk of overfitting. The validation procedure followed cross-validation and hold-out testing practices recommended for robust ML evaluation [35-37].

TABLE III. HYPERPARAMETER TUNING

Hyperparameter	**Selected value**	**Justification**
n_estimators	200 trees	Improves prediction stability
max_depth	10 Levels	Overfitting control
min_samples_split	4 Samples	Allows for more meaningful divisions
max_features	sqrt	Increases randomness and generalizability

The T-RAPPI model demonstrated remarkable performance during the validation phase, achieving an average accuracy of approximately 91.7%, while the MSE remained low and showed little variance across folds. The analysis also revealed that the three most influential predictor variables were the time of day, the line number, and the station. These results confirm the model's ability to effectively generalize bus arrival patterns, even when applied to new temporal or spatial scenarios within the Metropolitan system. Figure 3 provides a flowchart representation of the methodology applied.

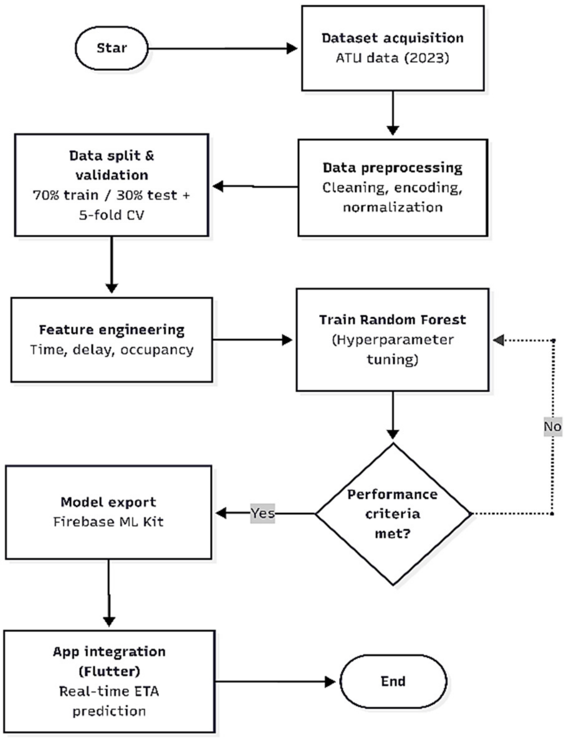


Fig. 3. Flowchart of the method used in developing the T-RAPPI model and the Metropolitano+ application.

III. RESULTS

The performance of the T-RAPPI predictive model is examined through accuracy metrics, error analysis, and cross-validation. In addition, the validation with end users of the Metropolitano Plus mobile application is detailed, carried out through a questionnaire based on the ISO/IEC 25010 standard, an instrument that allows assessing the perception of the quality of the software and the usability criteria of the solution.

A. *T-RAPPI Model*

1) *Global Accuracy*

Table IV summarizes the metrics obtained when evaluating T-RAPPI on the test set. The coefficient of determination ($R^2 = 0.9998$) indicates that the model explains nearly all variability of the target variable (Arrival Time Variation), while the MAE (0.0062 min) and RMSE (0.0912 min) confirm very small absolute and quadratic errors. The Average Symmetric Mean (ASM) of less than 0.06% shows that, on average, the percentage deviation between predicted and actual values is almost negligible.

TABLE IV. T-RAPPI MODEL EVALUATION

Metric	**Value**
MAE	0.0062
RMSE	0.0912
MAPE	0.0554 %
R²	0.9998
Max Error	4.0
Exp. Variance	1.0
MedAE	0.0

Although this performance demonstrates excellent predictive power, such a high R² value may suggest possible overfitting or data leakage if not properly cross-validated. To mitigate this risk, the model was evaluated using a 5-fold cross-validation scheme and separate training-testing partitions (70:30 split), ensuring that the test data remained unseen during training. The consistency of the results across folds supports the model's generalization capacity within the dataset used.

However, it is acknowledged that the homogeneity and limited variability of the dataset—originating from the same transport system and operational year—could contribute to the extremely high accuracy. This situation often occurs when the model learns highly repetitive patterns typical of a closed operational environment like the Metropolitano, where route timing and frequency are strongly standardized.

2) *Chart Analysis*

Figure 4 shows two visual analyses used to assess the predictive performance of the T-RAPPI Random Forest model:

a) Scatter plot - Predictions vs. Actual Values: This plot compares the model's predicted values against the actual recorded arrival-time variations. The points align almost perfectly along the 45-degree red reference line, indicating a near-one-to-one correspondence between predicted and observed values. This visual alignment corroborates the very high coefficient of determination (R² = 0.9998) and confirms that the model accurately captures the temporal behavior of bus arrivals in the Metropolitano system.

b) Histogram of residuals: The residual distribution (Actual-Predicted) is sharply centered around zero, forming a narrow and symmetric peak. This confirms that errors are minimal, evenly distributed, and lack systematic bias—an indicator of good model calibration and reliability. The absence of extreme residuals suggests that T-RAPPI maintains consistent performance across all operational intervals.

Figure 5 provides additional insights into the internal behavior of the T-RAPPI Random Forest model by analyzing variable importance and the distribution of prediction errors across value ranges.

a) Feature importance chart: This plot shows the relative contribution of each input variable to the model's predictive performance. The attributes STATUS (representing the operational state of the service) and REFERENCE (the bus identifier and route code) exhibit the highest aggregated importance, confirming their strong influence on arrival-time estimation. The remaining features contribute marginally, reflecting the model's efficiency in prioritizing the most informative operational variables. This result is consistent with prior transport prediction studies using ensemble learning, where operational status and reference identifiers often dominate the predictive hierarchy [38, 39].

b) RMSE by interval of actual values: This chart evaluates the Root Mean Squared Error (RMSE) within successive ranges of true values. The low and stable error observed across most intervals demonstrates the model's uniform accuracy. Slight increases in RMSE at extreme intervals correspond to infrequent operational conditions or atypical delays, which are expected in real-world transport systems. Overall, the distribution confirms that T-RAPPI maintains robust performance across different service states and value ranges.

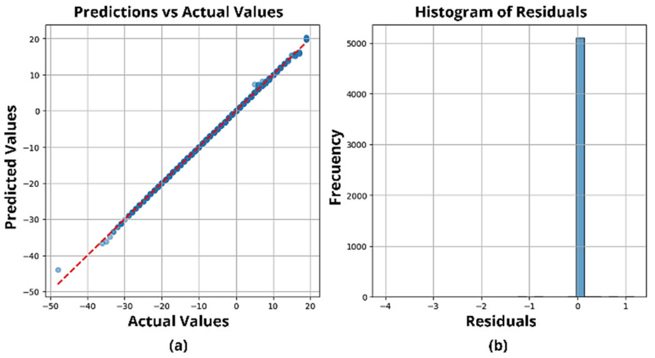


Fig. 4. Graphical analysis of T-RAPPI: (a) Scatter plot of Predictions vs. Real values, (b) Error histogram.

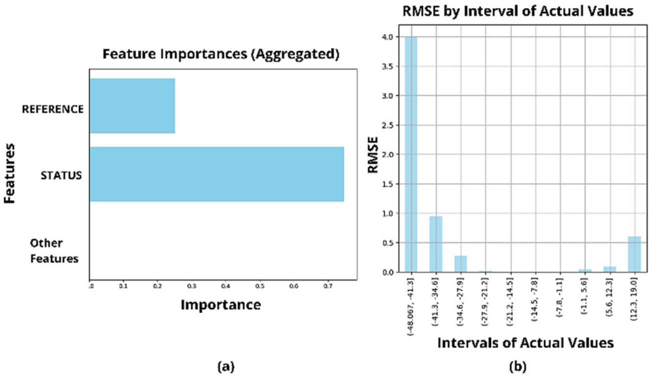


Fig. 5. Graphical analysis of T-RAPPI: (a) Feature importance chart, (b) RMSE chart by real value intervals.

Together, these analyses provide transparency regarding how T-RAPPI weights key variables and sustains low prediction errors, reinforcing its interpretability and reliability for operational deployment.

3) Cross-Validation

Figure 6 illustrates the MSE obtained for each fold during the 5-fold cross-validation of the T-RAPPI Random Forest model. Each bar represents the average error for one of the five partitions of the dataset used to evaluate generalization performance. The results show a mean MSE of 0.0323 with a standard deviation of ±0.0292, indicating low dispersion between folds and consistent predictive accuracy. Although folds 1 and 5 exhibit slightly higher MSE values, the remaining folds maintain near-zero errors, confirming that the model performs uniformly across different data subsets. These variations are expected in operational datasets where certain time periods contain irregular service behaviors or outlier events.

The overall pattern validates the robustness and stability of the T-RAPPI model under repeated sampling, demonstrating that its predictive capacity is not dependent on a specific training subset but rather generalizes effectively across the entire dataset.

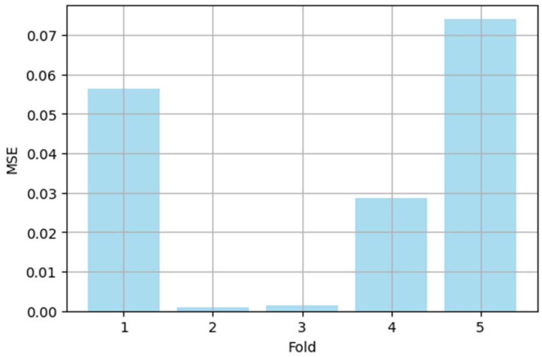


Fig. 6. MSE graph per fold (cross-validation).

B. Metropolitano Plus

1) Design of the Assessment Instrument

To assess the perceived quality of the Metropolitano Plus application, a structured questionnaire of 16 items with a five-point Likert scale (1 = "Strongly disagree," 5 = "Strongly agree") was designed, aligned with the five quality characteristics of ISO/IEC 25010: Functionality, Usability, Efficiency, Reliability, and Security (see Table IV).

Each dimension was represented by a balanced number of questions (Functionality: 4, Usability: 4, Efficiency: 2, Reliability: 3, Safety: 3), which allowed maintaining the symmetry of the instrument without increasing the cognitive load of the respondent. The preliminary internal consistency reached 0.93 for the global scale and values between 0.81 and 0.88 in the subscales, indicating acceptable reliability.

The final sample consisted of 25 end users, selected through an intentional sampling of maximum variation, so that the heterogeneity of usage profiles was captured. This sample size aligns with the thematic saturation recommendations proposed in [40] and corroborated in [41], which place the saturation between 15 and 30 informants for exploratory studies of software quality perception (see Table V).

TABLE V. QUESTIONNAIRE OF VALIDATION

Dimension	ID	Question
Functionality	P01	The application shows accurate and up-to-date information on bus schedules and capacities.
	P02	I have not encountered any problems when using the bus arrival prediction functionality.
	P03	Station capacity information is useful for planning trips.
	P04	All the features I expect to find are properly implemented in the app.
Usability	P05	I find the navigation in the app intuitive and easy to understand.
	P06	I had no difficulty finding the features I needed, such as bus arrival prediction or capacity information.
	P07	The design of the app makes it easy to use in different situations, such as on a bus or in a station.
	P08	The size of the interface elements (buttons, text) is suitable for easy interaction.
Efficiency	P09	The app responds quickly when I check the bus arrival prediction.
	P10	I have not experienced slowness or freezes when using key features such as bus arrival prediction or station capacity.
Reliability	P11	I have not experienced any unexpected interruptions when using the app.
	P12	I have not experienced any errors that affect my ability to use the app properly.
	P13	I trust that the information presented by the app is correct and frequently updated.
Security	P14	I feel safe entering my information into the app.
	P15	The app offers secure methods of access and verification, such as secure authentication.
	P16	I trust that the data I provide to the app is protected.

TABLE VI. RESULTS

Dimension	Average (points)	Satisfaction (%)
Functionality	4.39	87.80 %
Usability	4.44	88.80 %
Efficiency	4.42	88.40 %
Reliability	4.36	87.20 %
Security	4.41	88.27 %

2) Analysis by Dimension

Figure 7 presents the results of the user validation survey conducted to assess the quality of the Metropolitano Plus application according to the ISO/IEC 25010 standard. Each chart represents the average satisfaction scores (in percentage) obtained for specific evaluation items (P01–P10) grouped by quality dimension:

a) Functionality: The functionality dimension evaluates the degree to which the system provides accurate and reliable results according to user expectations. Scores range from 85.8% to 90.8%, with P01 achieving the highest rating, indicating that users perceive the app's core features as stable and dependable. The slightly lower score of P03 reflects opportunities to refine the perceived usefulness of the occupancy indicator.

b) Usability: Usability metrics (P05-P08) measure ease of interaction, interface clarity, and learnability. Scores are consistently above 87%, highlighting that users found the navigation intuitive and the visual design coherent. Minor variations, such as a lower rating in P08, suggest the need for improved contextual help or guidance for first-time users.

c) Efficiency: The efficiency dimension (P09-P10) assesses performance and resource optimization. Both indicators exceed 87.8%, with P10 scoring 88.8%, confirming that the app operates fluidly, with short response times and efficient integration of Firebase resources.

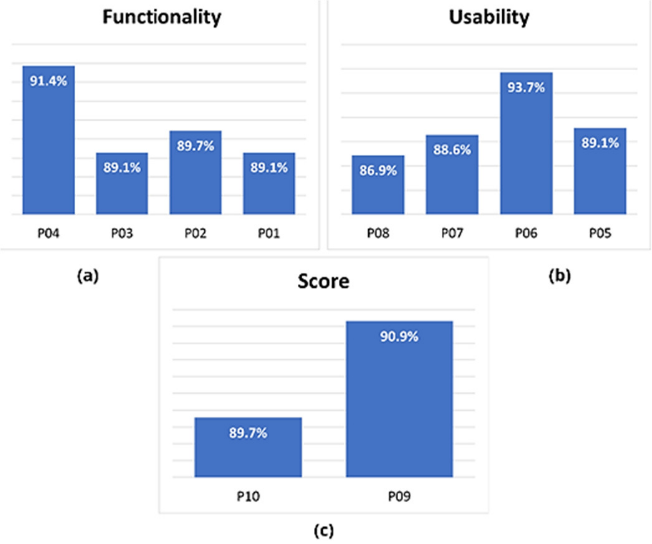


Fig. 7. Response graphs by dimension: (a) Functionality (b) Usability, (c) Efficiency.

Figure 8 presents the results for the last two quality dimensions assessed in the user validation of the Metropolitano Plus application, Reliability and Security, both evaluated according to the ISO/IEC 25010 model.

a) Reliability: This dimension evaluates the system's consistency and fault tolerance during operation. Indicators P10, P12, and P13 obtained satisfaction levels between 86.5% and 88.4%, reflecting stable app performance without service interruptions or unexpected crashes. The highest value for P13 indicates users' confidence in the app's capacity to maintain functionality during continuous use, even under varying network conditions.

b) Security: The security dimension measures user confidence in data protection and access control mechanisms. Indicators P14-P16 achieved scores between 84% and 94%, with P16 reaching the highest overall rating (92%). This result validates the effectiveness of authentication processes and secure data handling through Firebase Authentication and Firestore, ensuring that user information is managed safely and transparently. The slightly lower score in P15 suggests opportunities to enhance user awareness about privacy settings and permissions.

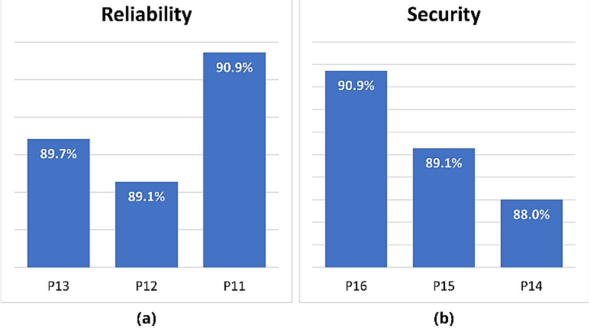


Fig. 8. Response graphs by dimension: (a) Reliability, (b) Security.

Overall, the reliability and security dimensions confirm that Metropolitano Plus provides a stable and trustworthy environment for end users, meeting critical quality standards for intelligent transport applications.

3) *Global Interpretation*

The validation results demonstrate strong user acceptance across all five ISO/IEC 25010 quality dimensions—Functionality, Usability, Efficiency, Reliability, and Security. All dimensions exceeded the 4.0 threshold (80%), confirming that Metropolitano Plus meets user expectations in both performance and design. The high ratings in functionality and usability show that users find the app intuitive, easy to navigate, and effective for checking routes, bus arrivals, and station occupancy. The results also highlight the app's technical stability and reliability. Users rated efficiency and reliability positively, indicating that the system runs smoothly, responds quickly, and maintains stable operation even with variable connectivity. The highest score was obtained in the security dimension (92%), reflecting confidence in the protection of user data and the secure implementation of authentication and storage mechanisms through Firebase.

Although the overall evaluation was highly favorable, two areas for improvement were identified. Users suggested enhancing the way station capacity data is displayed, making it more dynamic and detailed, and further refining error handling to ensure smoother performance in exceptional cases. Overall, the results confirm that Metropolitano Plus provides a reliable, user-friendly, and secure experience, strengthening its potential as a modern tool to improve public transport management and passenger satisfaction in Lima.

IV. DISCUSSIONS

A. *Implications of the Results*

In the context of the T-RAPPI model, the performance metrics ($R^2 = 0.9998$; MAE = 0.0062 min) demonstrate that the Random Forest–based method can accurately reproduce the temporal variations of the Corredor Metropolitano services. This outstanding level of fit not only reflects the model’s ability to replicate historical patterns but also confirms its predictive capacity, as validated through 5-fold cross-validation (mean MSE = 0.0323±0.0292). From an operational standpoint, these metrics imply prediction errors of less than ten seconds on average—considered functionally negligible in the context of fleet scheduling and real-time decision-making.

The performance of the model is consistent with findings from other studies that used ensemble learning for transport prediction. In [7, 8], Random Forest and multi-graph deep models effectively captured stable temporal behaviors in structured public transit systems. Similarly, ensemble learning frameworks have been recognized for their robustness and adaptability in data-limited transportation environments [29]. Recent studies also support these findings, as in [38], it was reported that ensemble models, including Random Forest and bagging, achieved superior results for smart traffic flow prediction, while in [39], the efficiency of ensemble techniques was demonstrated on proactive road-safety prediction using roadway geometry features. These studies reinforce the reliability of the ensemble approach implemented in T-RAPPI and its suitability for real-world intelligent transportation systems.

Regarding the Metropolitano Plus mobile application, the validation survey—based on the ISO/IEC 25010 quality model—yielded average scores of more than 4.36 in all dimensions, confirming that the system meets functional and usability expectations. The Security dimension achieved the highest score (P16 = 4.60; 92%), indicating that the authentication and data management mechanisms implemented via Firebase meet users' confidence standards. The Usefulness indicator (P03 = 4.28; 85.6%) suggests a potential area for improvement, particularly in the visualization of real-time information. Overall, the results confirm that both research objectives were successfully achieved:

- The Random Forest–based T-RAPPI model reached high predictive accuracy and stability under cross-validation.
- The Metropolitano Plus application, integrating the model, achieved strong user acceptance and quality validation scores.

Thus, this study provides evidence that combining predictive analytics with user-centered design can enhance both operational efficiency and passenger satisfaction in Latin American public transport systems.

B. Comparison with Previous Studies

Previous studies have shown significant improvements in demand prediction and travel times in subway and BRT systems. In [18], ML models were applied to evaluate satisfaction in public transportation, showing their potential to capture user-related dynamics. In [7], Random Forest outperformed SVM and semiparametric approaches in passenger demand prediction using smart card data. The T-RAPPI model exhibits outstanding performance, reaching a value of R^2 close to 1, thanks to its minute-by-minute data logging and the incorporation of contextual variables such as "time elapsed since last arrival" and "historical average delay", which allow it to capture temporal dependencies that traditional methods often ignore by relying exclusively on GPS or Automatic Fare Collection (AFC) data.

Furthermore, while the study in [9] analyzed the efficiency of Random Forest using spatial and demographic data in rural transport and [14] applied Random Forest-based approaches to predict metro ridership, the results of this study illustrate the flexibility of Random Forest by validating its performance with operational records in the Latin American context. Finally, user testing confirms that Metropolitano Plus meets the quality dimensions defined by ISO/IEC 25010, and user acceptance levels exceed 88%.

C. Utility in the Operating Environment

The integration of T-RAPPI into the Metropolitano Plus system provides a feedback mechanism that harmonizes operating schedules between transport system operators and passengers. This ecosystem allows station navigators to dynamically modify passage intervals in real time, optimize fleet distribution, and immediately prioritize corrective actions in the event of operational discrepancies. From the user's perspective, the availability of information on punctuality reduces uncertainty, which increases satisfaction levels; this is supported by the direct relationship already established between the reliability of information and the perceived quality of public transport [21]. In addition, the time accuracy achieved enables interdisciplinary applications in various fields, such as incident management in urban environments, emergency route planning, and critical infrastructure design.

D. Limitations and Threats

The predictive power of T-RAPPI is inherently dependent on historical data, implying that unpredictable and exceptional circumstances, such as unusual traffic or heavy rainfall, could negatively affect its ability to predict. This limitation aligns with the findings of [24], which highlighted that the accuracy of predictive models is strongly constrained by the quality of the input data, limiting their generalization when exposed to unforeseen conditions or outliers.

The perception survey was applied to only 25 participants; although this number is sufficient to support theoretical saturation, a larger sample would allow the results to be generalized through inferential analyses. Likewise, a geographical limitation is observed, since the model was built exclusively with data from the central section of the Metropolitano, so it requires additional validation before being applied in other segments of the system. Finally, the use of purposive sampling could also have introduced self-selection bias, since the participants already showed previous interest in the application, which could have influenced an overestimation of satisfaction levels.

E. Future Research Perspectives

The future implementation of T-RAPPI could benefit substantially from adaptive learning techniques and early detection of anomalies. The use of Random Forest has proven to be particularly effective in environments where data is noisy, seasonal, or discontinuous. In [41], it was shown that even under anomalous peaks and missing values in agricultural time series, RF maintains reliable accuracy due to its ensemble structure and the capacity to handle nonlinear relationships. Similarly, an Android malware detection framework using RF [42] achieved more than 98.6% accuracy across categories—highlighting its robustness in recognizing complex, multiclass behaviors.

Building on these properties, future versions of T-RAPPI could integrate continuous retraining mechanisms based on logical data segmentation (by station, route, or time slot), enabling the system to evolve into a self-adaptive model capable of real-time learning. This architecture could incorporate early warning modules to detect operational anomalies and support proactive decision-making. Additionally, several research directions are proposed:

- Integration of real-time data: Combine live GPS feeds, congestion indicators, and environmental factors to enhance adaptability in dynamic urban conditions.
- Expansion to other transport systems: Extend T-RAPPI to additional public transport modes in Lima and similar Latin American cities with limited digital infrastructure.
- Comparative modeling: Benchmark RF performance against alternative algorithms such as Gradient Boosting, XGBoost, or Deep Learning models to determine the most effective predictive strategy.
- User experience enhancement: Improve personalization and accessibility features in the Metropolitano Plus app to increase engagement and inclusion.

These proposals aim to strengthen T-RAPPI's scalability, adaptability, and contribution to intelligent transportation systems in developing urban contexts.

V. CONCLUSIONS

This study designed, evaluated, and deployed an intelligent transport solution for Lima's Corredor Metropolitano by integrating the Random Forest–based T-RAPPI model with the Metropolitano Plus mobile application. The objective was to enhance operational management and improve the user experience by providing accurate arrival-time predictions and real-time service information. The results demonstrate the effectiveness of the proposed system, as T-RAPPI achieved an R^2 of 0.9998 with absolute errors below ten seconds, while 5-fold cross-validation (MSE = 0.0323±0.0292) confirmed its robustness and generalizability. User validation based on ISO/IEC 25010 yielded average scores of more than 4.36 across all dimensions, with Security obtaining the highest evaluation (4.60 or 92%) and Capacity identified as the main area for improvement. Together, these findings validate both the predictive accuracy of the model and the perceived quality of the solution among ATU users.

A. Theoretical Implications

From an academic standpoint, this work provides a clear and replicable methodological framework for developing intelligent transportation solutions in data-limited environments, demonstrating that historical operational records are sufficient to train high-performance ML models for arrival-time prediction. Furthermore, this study contributes empirical evidence that supports the use of Random Forest as a robust baseline for public-transport forecasting and establishes guidelines for integrating ML models into mobile environments using lightweight architectures.

B. Practical Implications

From a practical perspective, the study delivers a scalable, low-cost, and easily deployable architecture built with Flutter and Firebase, enabling real-time inference and supporting operational decision-making for Corredor Metropolitano. The system directly improves service reliability, enhances user satisfaction, and offers ATU a viable digital tool that can be extended to additional corridors or replicated across other public-transport networks in Latin America. Overall, this work effectively bridges the gap between data analytics and public-service usability, presenting a functional model that aligns with the region's technological and financial constraints while delivering tangible operational benefits.

C. Theoretical Implications

From an academic perspective, this work contributes a replicable methodological framework that combines ML prediction (Random Forest) with mobile-app integration for intelligent transportation systems in data-limited environments. It demonstrates that historical operational data can effectively support arrival-time prediction without requiring costly real-time sensors.

D. Limitations and Future Work

This study presents limitations, including the absence of real-time contextual variables, a relatively small user-sample size (n = 25), and a focus restricted to a single transport corridor. Future research should incorporate external factors (e.g., weather, incidents), explore hybrid architectures such as RF-LSTM, and evaluate the transferability of the framework to other systems—such as Línea 1 Metro de Lima or additional BRT corridors in the region.

ACKNOWLEDGMENT

The authors are grateful to the Dirección de Investigación of the Universidad Peruana de Ciencias Aplicadas for the support provided for this research work through the UPC-EXPOST-2026-1 incentive.

REFERENCES

[1] F. Gonzales, "¿Cuánto se pierde en Lima Metropolitana por el tráfico?," *Instituto Peruano de Economía*, Feb. 06, 2024. https://ipe.org.pe/cuanto-se-pierde-en-lima-metropolitana-por-el-trafico/.

[2] Infraestructura Vial, "Desafíos y Recomendaciones para Mejorar el Metropolitano en Lima -," Feb. 04, 2024. https://infraestructuravial.pe/infraestructura-vial/desafios-y-recomendaciones-para-mejorar-el-metropolitano-en-lima/.

[3] A. Calatayud, S. Sánchez González, F. Bedoya-Maya, F. Giraldez Zúñiga, and J. M. Márquez, *Urban Road Congestion in Latin America and the Caribbean: Characteristics, Costs, and Mitigation*. Inter-American Development Bank, 2021.

[4] J. Rivas, M. Lujan, and R. Palma, "El estrés y su relación con el rendimiento laboral en conductores de transporte público de la empresa Allin Group-Javier Prado S.A, Lima-2022," *Ciencia Latina Revista Científica Multidisciplinar*, vol. 6, no. 2, pp. 2144–2163, Apr. 2022, https://doi.org/10.37811/cl_rcm.v6i2.2016.

[5] E. V. Vargas and H. J. P. Roque, "Evaluación de la problemática del servicio de las líneas alimentadoras del Metropolitano en el sistema de transporte urbano de Lima y Callao y propuesta de mejora basada en sistemas inteligentes de transporte," M.S. Thesis, Esan Graduate School of Business, 2023.

[6] G. Santos and N. Nikolaev, "Mobility as a Service and Public Transport: A Rapid Literature Review and the Case of Moovit," *Sustainability*, vol. 13, no. 7, Mar. 2021, Art. no. 3666, https://doi.org/10.3390/su13073666.

[7] I. Zimmo, D. Hörcher, R. Singh, and D. J. Graham, "Benchmarking Travel Time and Demand Prediction Methods Using Large-scale Metro Smart Card Data," *Periodica Polytechnica Transportation Engineering*, vol. 51, no. 4, pp. 357–374, Sept. 2023, https://doi.org/10.3311/PPtr.22252.

[8] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach," *Transportation Research Part C: Emerging Technologies*, vol. 127, June 2021, Art. no. 103063, https://doi.org/10.1016/j.trc.2021.103063.

[9] S. Imhof and K. Blättler, "Assessing spatial characteristics to predict DRT demand in rural Switzerland," *Research in Transportation Economics*, vol. 99, June 2023, Art. no. 101301, https://doi.org/10.1016/j.retrec.2023.101301.

[10] S. Cerqueira, E. Arsenio, J. Barateiro, and R. Henriques, "Moving from classical towards machine learning stances for bus passengers' alighting estimation: A comparison of state-of-the-art approaches in the city of Lisbon," *Transportation Engineering*, vol. 16, June 2024, Art. no. 100239, https://doi.org/10.1016/j.treng.2024.100239.

[11] Z. Wang, A. J. Pel, T. Verma, P. Krishnakumari, P. Van Brakel, and N. Van Oort, "Effectiveness of trip planner data in predicting short-term bus ridership," *Transportation Research Part C: Emerging Technologies*, vol. 142, Sept. 2022, Art. no. 103790, https://doi.org/10.1016/j.trc.2022.103790.

[12] M. Müller-Hannemann, R. Rückert, A. Schiewe, and A. Schöbel, "Estimating the robustness of public transport schedules using machine learning," *Transportation Research Part C: Emerging Technologies*, vol. 137, Apr. 2022, Art. no. 103566, https://doi.org/10.1016/j.trc.2022.103566.

[13] L. A. Makara, P. Maric, and A. Pekar, "Public transport congestion detection using incremental learning," *Pervasive and Mobile Computing*, vol. 91, Apr. 2023, Art. no. 101769, https://doi.org/10.1016/j.pmcj.2023.101769.

[14] A. H. AlKhereibi, T. G. Wakjira, M. Kucukvar, and N. C. Onat, "Predictive Machine Learning Algorithms for Metro Ridership Based on Urban Land Use Policies in Support of Transit-Oriented Development," *Sustainability*, vol. 15, no. 2, Jan. 2023, Art. no. 1718, https://doi.org/10.3390/su15021718.

[15] V. M. Orlando, E. G. Baquela, N. Bhouri, and P. A. Lotito, "Public transport demand estimation by frequency adjustments," *Transportation Research Interdisciplinary Perspectives*, vol. 19, May 2023, Art. no. 100832, https://doi.org/10.1016/j.trip.2023.100832.

[16] N. Nagaraj, H. L. Gururaj, B. H. Swathi, and Y.-C. Hu, "Passenger flow prediction in bus transportation system using deep learning," *Multimedia Tools and Applications*, vol. 81, no. 9, pp. 12519–12542, Apr. 2022, https://doi.org/10.1007/s11042-022-12306-3.

[17] U. K. Lilhore *et al.*, "Design and Implementation of an ML and IoT Based Adaptive Traffic-Management System for Smart Cities," *Sensors*, vol. 22, no. 8, Apr. 2022, https://doi.org/10.3390/s22082908.

[18] E. Ruiz, W. F. Yushimito, L. Aburto, and R. De La Cruz, "Predicting passenger satisfaction in public transportation using machine learning models," *Transportation Research Part A: Policy and Practice*, vol. 181, Mar. 2024, Art. no. 103995, https://doi.org/10.1016/j.tra.2024.103995.

[19] Y. Rong *et al.*, "Du-Bus: A Realtime Bus Waiting Time Estimation System Based On Multi-Source Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24524–24539, Dec. 2022, https://doi.org/10.1109/TITS.2022.3210170.

[20] L. Zhu, C. Chen, H. Wang, F. R. Yu, and T. Tang, "Machine Learning in Urban Rail Transit Systems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2182–2207, Mar. 2024, https://doi.org/10.1109/TITS.2023.3319135.

[21] S. Güner, K. Taşkın, H. İ. Cebeci, and E. Aydemir, "Service Quality in Rail Systems: Listen to the Voice of Social Media," *Transportation Research Record*, vol. 2678, no. 6, pp. 824–847, June 2024, https://doi.org/10.1177/03611981231200225.

[22] Z. Yin and B. Zhang, "Bus Travel Time Prediction Based on the Similarity in Drivers' Driving Styles," *Future Internet*, vol. 15, no. 7, June 2023, Art. no. 222, https://doi.org/10.3390/fi15070222.

[23] A. Gal-Tzur and S. Albagli-Kim, "Systematic Analysis of the Literature Addressing the Use of Machine Learning Techniques in Transportation—A Methodology and Its Application," *Sustainability*, vol. 16, no. 1, Dec. 2023, Art. no. 207, https://doi.org/10.3390/su16010207.

[24] A. Nuzzolo and A. Comi, "Dynamic Optimal Travel Strategies in Intelligent Stochastic Transit Networks," *Information*, vol. 12, no. 7, July 2021, Art. no. 281, https://doi.org/10.3390/info12070281.

[25] D. Traverso, G. Pacheco, and P. Castañeda, "T-RAPPI: A Machine Learning Model for the Corredor Metropolitano," in *Proceedings of the 11th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2025, pp. 374–381, https://doi.org/10.5220/0013220700003941.

[26] J. Machado, R. Sousa, H. Peixoto, and A. Abelha, "Ethical Decision-Making in Artificial Intelligence: A Logic Programming Approach," *AI*, vol. 5, no. 4, pp. 2707–2724, Dec. 2024, https://doi.org/10.3390/ai5040130.

[27] S. Oktaviana R, P. W. Handayani, A. N. Hidayanto, and B. B. Siswanto, "Healthcare data governance assessment based on hospital management perspectives," *International Journal of Information Management Data Insights*, vol. 5, no. 1, June 2025, Art. no. 100342, https://doi.org/10.1016/j.jjimei.2025.100342.

[28] A. Panthakkan, A. Gurjarand, J. Patel, H. Patel, S. Parikh, and W. Mansoor, "A Comparative Analysis of Ensemble Strategies for Enhanced Machine Learning Results," in *Applications of Artificial Intelligence and Data Science*, vol. 2601, M. Mahmud, N. Pillay, and M. S. Kaiser, Eds. Springer Nature Switzerland, 2026, pp. 49–59.

[29] A. M. Quadir, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Enhanced Preprocessing Approach Using Ensemble Machine Learning Algorithms for Detecting Liver Disease," *Biomedicines*, vol. 11, no. 2, Feb. 2023, Art. no. 581, https://doi.org/10.3390/biomedicines11020581.

[30] H. B. Waseem and M. A. H. Talpur, "Impact Assessment of Urban Pull-factors to cause Uncontrolled Urbanization: Evidence from Pakistan," *Sukkur IBA Journal of Computing and Mathematical Sciences*, vol. 5, no. 1, pp. 37–52, Mar. 2021, https://doi.org/10.30537/sjcms.v5i1.589.

[31] M. A. H. Talpur, S. H. Khahro, T. H. Ali, A. K. Hindu, and H. Marvi, "Time-space accessibility modeling: an individual-centered approach to develop a transport policy framework in remote environments," *Discover Cities*, vol. 2, no. 1, Sept. 2025, Art. no. 76, https://doi.org/10.1007/s44327-025-00123-w.

[32] M. Hassan, A. Al Nafees, S. S. Shraban, A. Paul, and H. D. Mahin, "Application of machine learning in intelligent transport systems: a comprehensive review and bibliometric analysis," *Discover Civil Engineering*, vol. 2, no. 1, May 2025, Art. no. 98, https://doi.org/10.1007/s44290-025-00256-2.

[33] G. Chen and J. W. Zhang, "Intelligent transportation systems: Machine learning approaches for urban mobility in smart cities," *Sustainable Cities and Society*, vol. 107, July 2024, Art. no. 105369, https://doi.org/10.1016/j.scs.2024.105369.

[34] J. Zhang *et al.*, "The Application of Machine Learning and Deep Learning in Intelligent Transportation: A Scientometric Analysis and Qualitative Review of Research Trends," *Sustainability*, vol. 16, no. 14, July 2024, Art. no. 5879, https://doi.org/10.3390/su16145879.

[35] M. Mohammadagha, S. Asadi, and H. Kazemi Naeini, "Evaluating machine learning performance using python for neural network models in urban transportation in New York city case study," *Journal of Economy and Technology*, vol. 4, pp. 266–283, 2026, https://doi.org/10.1016/j.ject.2025.11.001.

[36] J. Qiu, "An Analysis of Model Evaluation with Cross-Validation: Techniques, Applications, and Recent Advances," *Advances in Economics, Management and Political Sciences*, vol. 99, no. 1, pp. 69–72, Sept. 2024, https://doi.org/10.54254/2754-1169/99/2024OX0213.

[37] I. Gharbi, F. Taia-Alaoui, H. Fourati, N. Vuillerme, and Z. Zhou, "Transportation Mode Detection Using Learning Methods and Self-

Contained Sensors: Review," *Sensors*, vol. 24, no. 22, Nov. 2024, Art. no. 7369, https://doi.org/10.3390/s24227369.

[38] A. S. Alkarim, A. S. Al-Malaise Al-Ghamdi, and M. Ragab, "Ensemble Learning-based Algorithms for Traffic Flow Prediction in Smart Traffic Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13090–13094, Apr. 2024, https://doi.org/10.48084/etasr.6767.

[39] P. Manoj, K. C. Manjunath, and P. B. Kotagi, "Machine Learning Models for Proactive Road Safety: Evaluating Regression and Ensemble Techniques Based on Road Geometry," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25951–25958, Aug. 2025, https://doi.org/10.48084/etasr.11569.

[40] S. Rahimi and M. Khatooni, "Saturation in qualitative research: An evolutionary concept analysis," *International Journal of Nursing Studies Advances*, vol. 6, June 2024, Art. no. 100174, https://doi.org/10.1016/j.ijnsa.2024.100174.

[41] J. Fondaj, M. Hamiti, S. Krrabaj, X. Zenuni, and J. Ajdari, "Comparison of Predictive Algorithms for IOT Smart Agriculture Sensor Data," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 21, pp. 65–78, Nov. 2023, https://doi.org/10.3991/ijim.v17i21.44143.

[42] A. Al Sharah, Y. Abu Alrub, H. Abu Owida, E. Abu Elsoud, N. Alshdaifat, and H. Khtatnaha, "An Adaptive Framework for Classification and Detection of Android Malware," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 18, no. 21, pp. 59–73, Nov. 2024, https://doi.org/10.3991/ijim.v18i21.49669.