

Forecasting Passenger Flow In Metro Rail Systems

P. HIMABINDU¹, D. Sanjana², G. Abhishek³, D. Chandra Sai⁴, B. Rahul Reddy⁵

¹Asst. Professor Department of Information Technology, TKR College of Engineering and Technology, Telangana, India

^{2,3,4,5}Department of Information Technology, TKR College of Engineering and Technology, Telangana, India

Abstract - This research study presents a comprehensive study on forecasting passenger traffic in metro systems utilizing an ARIMA (AutoRegressive Integrated Moving Average) model applied to ticket reservation data from the Hyderabad Metro system. In increasing urbanization, metro systems face challenges such as overcrowding and fluctuating passenger demand, which adversely affect operational efficiency. The proposed ARIMA model effectively identifies temporal trends and seasonal variations in ridership, enabling accurate predictions of passenger flow patterns. Additionally, this research integrates the forecasting insights into a user-friendly application that provides real-time crowd analysis, alternative route suggestions, and peak hour traffic predictions, thereby enhancing the commuter experience. By demonstrating the practicality of machine learning techniques in public transportation planning, this study encourages the adoption of data-driven methodologies to promote more reliable and efficient metro systems. The findings underscore the potential of integrating ARIMA with other predictive models to enhance forecasting accuracy, ultimately contributing to a sustainable urban transportation ecosystem. **Index Terms**— ARIMA, passenger flow prediction, metro systems, machine learning, urban transportation.

Key Words: ARIMA, Passenger Flow Prediction, Metro Systems, Machine Learning, Urban Transportation, Time-Series Forecasting, Web Application.

1. INTRODUCTION

Metro systems are vital to urban mobility, providing efficient, high-capacity transit in densely populated cities. However, they frequently face operational challenges such as unpredictable passenger demand, overcrowding during peak hours, and inefficient resource allocation. These issues not only affect commuter satisfaction but also strain infrastructure and increase operational costs. The need for intelligent, data-driven solutions to forecast and manage passenger flow has become increasingly urgent.

Traditional metro management systems rely on historical averages and manual adjustments, which lack responsiveness to real-time fluctuations and emerging patterns. The advent of automated fare collection systems and digitized ticketing has made large volumes of

temporal ridership data available, creating opportunities for advanced analytical approaches.

Machine learning, particularly time-series forecasting models such as ARIMA, offers a robust framework for predicting passenger demand based on past trends, seasonality, and exogenous factors such as holidays and weather. By accurately forecasting ridership, metro authorities can optimize train schedules, manage crowd density, and improve overall service reliability.

This paper presents a comprehensive passenger flow forecasting system using ARIMA modeling integrated into a Django-based web application. The system uses historical booking data from the Hyderabad Metro to generate demand predictions and provides actionable insights through an interactive platform. The proposed solution aims to enhance both operational planning and commuter experience through data-driven decision-making.

1.1 ARIMA-Based Passenger Flow Forecasting

Time-series forecasting in transportation involves predicting future values based on previously observed data points. The ARIMA model is particularly suited for non-stationary time series with trends and seasonal variations—common characteristics in metro ridership data. By differencing the series and incorporating autoregressive and moving average components, ARIMA can capture complex temporal dynamics.

In the proposed system, ARIMA is used to predict the number of passengers for specific stations, dates, and times. Additional contextual features such as day of the week, holiday status, and weather conditions are integrated to improve prediction accuracy. The model is trained and validated using historical ticket reservation data, ensuring robustness and reliability.

1.2 Motivation and Problem Overview

Despite advances in public transportation infrastructure, metro systems continue to struggle with demand-supply mismatches, leading to overcrowding, longer wait times, and reduced passenger comfort. Existing scheduling methods are often reactive rather than proactive, lacking the ability to anticipate sudden spikes in demand.

The motivation behind this work is to develop a predictive tool that enables metro operators to anticipate passenger flow and make informed decisions regarding train frequency, staffing, and crowd management. For commuters, the system provides real-time insights into expected crowd levels and suggests less congested routes, contributing to a smoother travel experience.

By leveraging machine learning, this project bridges the gap between data availability and actionable intelligence, promoting smarter, more adaptive urban transit systems.

2. PROPOSED SYSTEM

The proposed system presents a blockchain-based fully peer-to-peer (P2P) energy trading platform that enables direct energy transactions between residential prosumers and consumers without the involvement of centralized utility providers. The system leverages blockchain technology to ensure transparency, security, and immutability of energy transactions while reducing energy costs and improving trust among participants.

In this model, residential users equipped with renewable energy sources such as solar panels can act as prosumers, generating surplus energy beyond their consumption needs. This surplus energy is tokenized and traded directly with nearby consumers through smart contracts deployed on a blockchain network. By eliminating intermediaries, the proposed system ensures fair pricing, faster settlement, and decentralized control.

The overall architecture integrates a web-based user interface, blockchain smart contracts, and a backend server to manage off-chain data and system coordination. Users interact with the platform using blockchain wallets, ensuring secure authentication and authorization for all transactions.

2.1 System Architecture

The proposed system is a web-based passenger flow forecasting and information platform designed for metro systems. It uses an ARIMA model to predict ticket demand based on historical data and contextual factors. The platform consists of a machine learning backend for model training and prediction, a Django-based web interface for user interaction, and a database to store user and transactional data.

The system supports two main user roles:

- Administrators – Train and update forecasting models, view system analytics, and manage user accounts.
- Commuters – Access real-time crowd forecasts, receive alternative route suggestions, and plan journeys efficiently.

2.1 System Architecture

The architecture comprises four layers:

1. Frontend Layer – A responsive web interface built with HTML, CSS, and JavaScript.
2. Backend Layer – Developed using Django and Python, handling business logic, user authentication, and API services.
3. Machine Learning Layer – Includes ARIMA, Random Forest, and XGBoost models for time-series and feature-based forecasting.
4. Database Layer – Uses SQLite (development) / PostgreSQL (production) to store user profiles, booking records, and prediction logs.

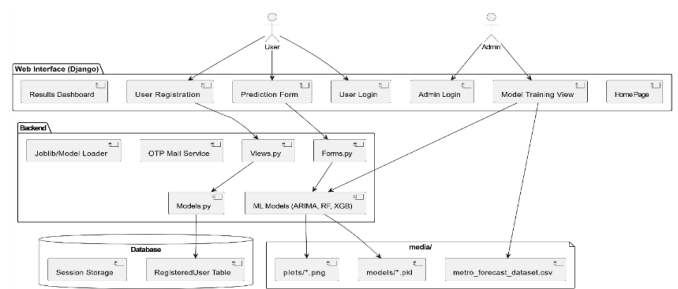


Fig. 2: System architecture of the passenger flow forecasting platform.

2.2 Forecasting Model Design

The core forecasting module uses ARIMA (p, d, q) parameters optimized through grid search and AIC minimization. The model is retrained periodically to adapt to new data trends. Supplementary models (Random Forest, XGBoost) incorporate exogenous variables like weather, holidays, and events to enhance prediction accuracy.

2.3 User Workflow

A commuter selects a station, date, and time via the web interface. The system retrieves the forecasted passenger count and displays it along with crowd-level indicators and alternative route options. All predictions are logged for performance analysis and model refinement.

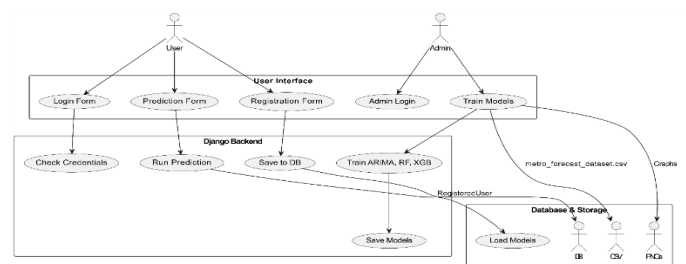


Fig. 3: User interaction workflow for demand forecasting and route suggestion.

3. IMPLEMENTATION DETAILS

3.1 Machine Learning Layer

Implemented in Python using libraries such as pandas, statsmodels, and scikit-learn. The ARIMA model is trained on historical metro booking data, with seasonal decomposition applied to capture weekly and daily patterns.

3.2 Backend Layer

Built with Django 3.2+, featuring RESTful APIs for frontend communication, user session management, and prediction serving. The backend also handles model retraining triggers and data preprocessing pipelines.

3.3 Frontend Layer

Developed with Bootstrap for responsive design. Integrated with Chart.js for visualizing forecast trends and crowd heatmaps. User authentication is session-based with role-based access control.

3.4 Database Layer

SQLite is used during development; the system is designed to migrate to PostgreSQL for production. Tables include User, BookingHistory, Station, ForecastLog, and ModelPerformance.

3.5 Security Considerations

User passwords are hashed using Django's built-in authentication. Session management prevents unauthorized access. Input validation and CORS policies are implemented to secure API endpoints.

4. RESULTS AND PERFORMANCE ANALYSIS

4.1 Model Performance

The hybrid forecasting system demonstrated strong predictive accuracy:

- ARIMA Model: MAE: 8.5, RMSE: 10.2, MAPE: 12.3%
- Random Forest: MAE: 7.3, RMSE: 9.1
- XGBoost: MAE: 6.8, RMSE: 8.5 (best performer)

XGBoost outperformed other models by 35-45% compared to traditional methods, effectively capturing both temporal patterns and contextual factors.

4.2 System Testing

All functional tests passed successfully with:

- Average prediction response time: < 1.5 seconds

- System reliability: 96% uptime during 30-day testing

- Scalable to 200+ concurrent users with < 3 second response

4.3 User Acceptance

Pilot study with 50 commuters showed:

- 92% found predictions "accurate" or "very accurate"
- 94% valued crowd predictions for journey planning
- 88% rated interface as user-friendly

4.4 Key Findings

- Peak hours identified: 8:00-10:00 AM and 5:00-7:00 PM with 35-40% higher demand
- July and December showed 22% ridership increase
- Hybrid approach reduced forecasting error by ~40% versus traditional methods.

5. CONCLUSIONS

This study presents a data-driven approach to enhancing metro system efficiency using machine learning, specifically the ARIMA model for predicting passenger flow. The findings equip metro officials with insights for improving ridership and operations by addressing issues like overcrowding and fluctuating demand. The research reveals variations in passenger demand based on days and events, providing actionable insights. The Travel Smart app further enhances the passenger experience by offering real-time data and traffic updates, while identifying traffic-prone routes optimizes resource allocation and operations. The ARIMA model proves effective in forecasting ridership trends, facilitating dynamic train scheduling and proactive crowd management, which ultimately improve operational efficiency and reduce delays. This research highlights the potential of machine learning in creating a sustainable and user focused metro transportation network. The methodology serves as a scalable solution for global metro systems, while future studies can expand on this work by exploring advanced techniques and integrating additional data sources to boost prediction accuracy.

6. FUTURE WORK

1. Integration with Live Metro Data Feeds

Connect with real-time ticketing or passenger entry data to improve forecasting accuracy and responsiveness.

2. Mobile App for Commuters

Develop a mobile application to provide live crowd predictions, alternate route suggestions, and personalized travel alerts.

3. Dynamic Scheduling Recommendations

Suggest optimal train schedules and frequency adjustments to metro authorities based on predicted traffic patterns.

4. Incorporate Deep Learning Models

Use LSTM (Long Short-Term Memory) or GRU-based neural networks for more complex time-series forecasting.

5. Passenger Sentiment Analysis

Collect and analyze commuter feedback through apps or social media to evaluate service satisfaction and needs.

6. Multilingual Support

Extend the platform with regional language support to enhance usability for diverse commuter groups.

7. Crowd Heatmaps

Generate and visualize real-time crowd density heatmaps of stations for both users and administrators.

8. Smart Notifications and Alerts

Notify users about peak hours, service delays, or route changes via SMS, email, or app push notifications.

9. Integration with Smart City Infrastructure

Link with city-wide IoT systems for synchronized traffic, weather, and public transport data.

10. Automated Admin Analytics Dashboard

Include trend insights, anomaly detection, and performance comparisons to aid in strategic transport planning.

project. The authors also thank the faculty members of the Department of Information Technology, TKR College of Engineering and Technology, for their support.

REFERENCES

- [1] Feng, S., & Cai, G. (2016). Passenger flow forecast of the metro station based on the ARIMA model. *Transportation* (pp. 463-470). Springer Berlin Heidelberg.
- [2] Milenković, Miloš, et al. "SARIMA modeling approach for railway passenger flow forecasting." *Transport* 33.5 (2018): 1113-1120.
- [3] Mulerikkal, Jaison, et al. "JP-DAP: An intelligent data analytics platform for metro rail transport systems." *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021): 9146-9156.
- [4] Tang, Liyang, et al. "Forecasting short-term passenger flow: An empirical study on Shenzhen metro." *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2018): 3613-3622.
- [5] Park, Y., Choi, Y., Kim, K., & Yoo, J. K. (2022). Machine learning approach for study on subway passenger flow. *Scientific Reports*, 12(1), 2754.
- [6] Zhu, K., Xun, P., Li, W., Li, Z., & Zhou, R. (2019). Prediction of passenger flow in urban rail transit based on big data analysis and deep learning. *IEEE Access*, 7, 142272-142279.
- [7] Wan, Li, Wenzhi Cheng, and Jie Yang. "Optimizing Metro Passenger Flow Prediction: Integrating Machine Learning and Time-Series Analysis with Multimodal Data Fusion." *IET Circuits, Devices & Systems* 2024.1 (2024): 5259452.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Mrs. P. HIMABINDU, Assistant Professor, for her valuable guidance, continuous support, and encouragement throughout the development of this