A Comparative Study of Classical and Deep Learning Approaches for Passenger Flow Forecasting in the London Underground

Gabriel Ferreira da Costa
gabriel.fc@aluno.ufop.edu.br
Department of Computing
Federal University of Ouro Preto (UFOP)
Ouro Preto, Minas Gerais, Brazil

Anderson Almeida Ferreira
anderson.ferreira@ufop.edu.br
Department of Computing
Federal University of Ouro Preto (UFOP)
Ouro Preto, Minas Gerais, Brazil

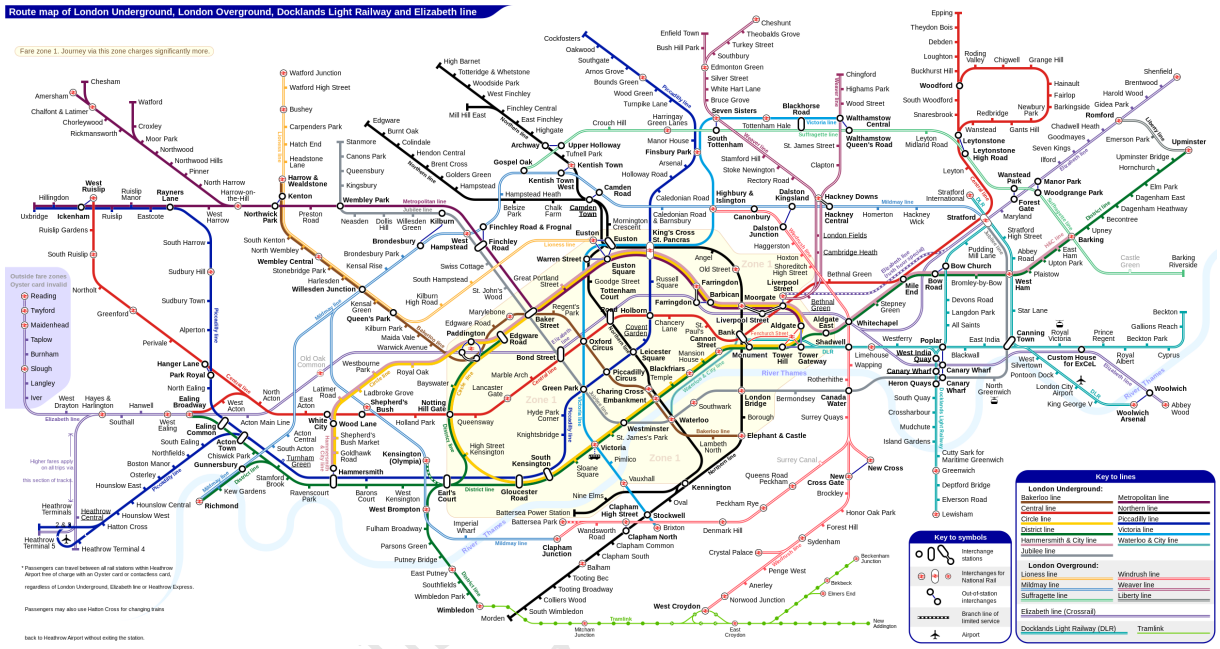

Figure 1: Tube map of the London Underground system. Source: [19].

Abstract

Accurate passenger flow forecasting in urban rail networks is critical for effective transportation planning and congestion management. This paper presents a comprehensive comparative analysis of traditional statistical models and advanced deep learning techniques for predicting passenger demand in the London Underground system. Specifically, we investigate time series approaches such as ARIMA and Facebook Prophet, alongside Long Short-Term Memory (LSTM) neural networks. Leveraging extensive real-world passenger flow data, we rigorously assess model performance based on predictive accuracy, generalization capability, and robustness. Our findings reveal the relative advantages and inherent limitations of each modeling paradigm, demonstrating the superior ability of deep learning methods to capture complex temporal dependencies and dynamic patterns inherent in urban mobility data. These insights provide valuable guidance for practitioners and researchers to enhance urban transit forecasting.

Data Mining, Ouro Preto, MG

ACM ISBN 978-1-4503-XXXX-X/2025/06
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Computing methodologies** → **Neural networks**; **Machine learning approaches**.

Keywords

Passenger Flow Forecasting, London Underground, Time Series Prediction, Data Mining, Classical Methods, Deep Learning, Spatio-temporal Graphs, Public Transportation

ACM Reference Format:
Gabriel Ferreira da Costa and Anderson Almeida Ferreira. 2025. A Comparative Study of Classical and Deep Learning Approaches for Passenger Flow Forecasting in the London Underground. In *Proceedings of (Data Mining)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Urban mobility systems are fundamental for the functioning of modern cities, with metro networks serving millions of passengers daily [21]. Efficient planning and management of these systems rely heavily on accurate forecasting of passenger flow, which enables transit authorities to optimize resource allocation, reduce congestion, and improve the overall user experience [24].

The London Underground is one of the busiest and most complex metro networks globally, characterized by a large number of stations, interconnected lines, and dynamic passenger demand patterns that vary with time of day, day of the week, and special events [18]. Predicting passenger flow in such a system poses challenges due to its spatio-temporal complexity and the influence of multiple external factors [13].

Motivation: Although large volumes of operational data are available, there is a lack of systematic studies comparing traditional statistical models and modern machine learning approaches for metro passenger flow forecasting. Classical models such as ARIMA have been widely adopted but may struggle with capturing non-linearities and complex seasonal patterns [1]. Recent advances in deep learning, particularly Long Short-Term Memory (LSTM) networks, promise to better model such complexities [5, 6]. Additionally, the Facebook Prophet model offers an intuitive framework for incorporating seasonality and calendar effects [17]. A comparative evaluation of these methods in the context of the London Underground can provide valuable insights for model selection and practical deployment.

Hypotheses:

- $\mathcal{H}_1$: LSTM models can outperform classical approaches by capturing complex non-linear and long-range temporal dependencies in passenger flow data.
- $\mathcal{H}_2$: Forecasting models that explicitly incorporate seasonality and calendar events (e.g., Prophet) achieve improved accuracy over purely autoregressive models.
- $\mathcal{H}_3$: Classical models (ARIMA) remain a competitive option for short-term forecasts in time series segments that exhibit stationarity and linear behavior.

Data Mining Pipeline: To enable effective forecasting, the raw passenger flow data undergoes a structured data mining process comprising the following stages:

(1) *Data Collection:* Acquisition of relevant historical data from publicly available source [9].
(2) *Data Cleaning:* Preparation of the dataset to ensure quality, consistency, and reliability.
(3) *Feature Engineering:* Generation of informative attributes that enhance the predictive capability of the models.
(4) *Data Transformation:* Processing and structuring of the data to make it suitable for input into forecasting models.
(5) *Model Training and Validation:* Implementation of training and evaluation procedures, including cross-validation, to assess model performance and generalization [10].

The main objective of this work is to evaluate and compare ARIMA, Prophet, and LSTM models in forecasting passenger flow for the London Underground, analyzing their predictive accuracy and practical applicability.

The remainder of this paper is structured as follows: Section 2 presents the theoretical and mathematical foundations of time series forecasting; Section 3 details the dataset and methodology; Section 4 presents the experimental results and their analysis; and Section 5 concludes with final considerations and directions for future research.

2 Literature Review

This section provides a comprehensive overview of the theoretical foundations and related research relevant to passenger flow forecasting in public transportation systems. We begin by introducing fundamental concepts of time series, followed by a discussion of widely used forecasting models, including statistical, machine learning, and deep learning approaches. In addition, we highlight key challenges commonly addressed in the literature, such as data seasonality, non-linearity, and model generalization, often tackled through techniques like cross-validation [10].

2.1 Time Series

A time series is a sequence of observations collected over time, typically indexed by discrete time steps. Time series data can be classified according to the number of variables and the regularity of the sampling process.

2.1.1 Univariate and Multivariate Time Series. A *univariate* time series consists of a single variable recorded over time, defined as:

$$\mathcal{X} = x_1, x_2, \ldots, x_T, \tag{1}$$

where $x_t \in \mathbb{R}$ represents the observed value at time t, and T is the total number of observations.

In contrast, a *multivariate* time series involves multiple variables recorded simultaneously over time. It is represented as:

$$\mathcal{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T, \tag{2}$$

where each observation $\mathbf{x}_t \in \mathbb{R}^d$ is a d-dimensional vector containing the values of d distinct variables at time t. Multivariate time series allow for the modeling of interdependencies between different variables, which is particularly relevant in complex systems such as transportation networks, where factors like passenger counts, weather, and special events may interact.

In this work, although the primary focus is on univariate forecasting of passenger flow at individual stations, the proposed methodology can be extended to multivariate settings, where exogenous variables (e.g., weather, holidays) are incorporated to enhance prediction accuracy.

2.1.2 Regular and Irregular Time Series. Time series can also be categorized based on the regularity of data collection:

- **Regularly Sampled Time Series**: Observations are collected at consistent, fixed intervals (e.g., every hour, every day). This is the most common setting in transportation systems with automated data collection, such as turnstile counts or ticket validations.
- **Irregularly Sampled Time Series**: The time intervals between observations are uneven, which can occur due to manual data collection, sensor failures, or event-driven sampling. Special care is required to handle irregular time series, often

involving interpolation, resampling, or the use of models that natively support irregular timestamps.

The dataset used in this study, provided through the Kaggle platform [9], consists of regularly sampled, univariate time series corresponding to historical passenger flow at London Underground stations. Each station's data represents the temporal evolution of passenger counts, typically aggregated at daily or hourly resolution, making it suitable for the application of the forecasting models evaluated in this work.

2.2 Forecasting Models

In this work, three complementary approaches for time series forecasting are evaluated, each with distinct mathematical foundations and capabilities.

2.2.1 Autoregressive Integrated Moving Average (ARIMA). ARIMA models, introduced by Box and Jenkins [1], are a classical statistical approach for modeling univariate time series. An ARIMA(p, d, q) model consists of:

- p: the number of autoregressive (AR) terms,
- d: the degree of differencing to achieve stationarity,
- q: the number of moving average (MA) terms.

The general form of an ARIMA model is:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \tag{3}$$

where y_t is the differenced series (after applying d differencing operations if necessary), c is a constant term, ϕ_i and θ_i are the model parameters, and ϵ_t represents the white noise error term, assumed to be independently and identically distributed with zero mean and constant variance.

ARIMA models are well-suited for capturing linear temporal dependencies, especially in stationary or weakly non-stationary series. However, they may be inadequate for highly non-linear patterns or complex seasonal structures.

2.2.2 Prophet. Prophet, proposed by Taylor and Letham [17], is an additive model designed to handle time series with strong seasonalities, multiple seasonal effects, and known external events such as holidays. The model decomposes the time series as:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{4}$$

where:

- $g(t)$ models the trend component, using either piecewise linear or logistic growth functions,
- $s(t)$ captures seasonal patterns through Fourier series,
- $h(t)$ accounts for the effects of holidays or special events,
- ϵ_t is an error term.

Prophet is designed for ease of use, minimal parameter tuning, and interpretability, making it particularly attractive for practitioners dealing with real-world operational data.

2.2.3 Long Short-Term Memory Networks (LSTM). LSTM networks, introduced by Hochreiter and Schmidhuber [6], are a class of Recurrent Neural Networks (RNNs) specifically designed to address the problem of vanishing and exploding gradients, which commonly affects traditional RNNs in learning long-term dependencies.

The core of an LSTM network consists of memory cells with gated mechanisms that control information flow. The hidden state h_t and cell state c_t at time t are updated according to:

$$(h_t, c_t) = \text{LSTM}(x_t, h_{t-1}, c_{t-1}), \tag{5}$$

where x_t is the input at time t, h_{t-1} is the previous hidden state, and c_{t-1} is the previous cell state. Through this architecture, LSTMs can learn complex, non-linear temporal relationships and have been successfully applied to a wide range of forecasting tasks, including traffic flow and passenger demand prediction [2, 23].

2.3 Passenger Flow Forecasting in Urban Mobility

Numerous studies have addressed passenger flow forecasting in public transportation networks using both traditional and deep learning models. For metro systems, accurately capturing temporal patterns is essential, given the pronounced daily and weekly seasonality and the influence of exogenous factors such as holidays or strikes [3, 25].

Despite significant advances, the literature reveals that there is no universally superior method, and the performance of forecasting models depends heavily on the characteristics of the dataset, including its linearity, seasonality, presence of anomalies, and data granularity [25].

In this context, evaluating ARIMA, Prophet, and LSTM models on real-world data from the London Underground, made available through the Kaggle platform [9], provides valuable insights into the applicability and limitations of these techniques for practical operational forecasting.

2.4 Related Work

Passenger flow forecasting in public transport systems has been extensively studied using various methodologies. Classical statistical models such as ARIMA have been applied to short-term forecasting problems, often with reasonable performance for stable, linear time series segments [16].

Recent studies have explored the use of deep learning techniques, particularly LSTM networks, for capturing complex temporal dependencies in passenger flow data. Zhang et al. [23] demonstrated the superiority of LSTM models over traditional approaches in metro ridership prediction, highlighting their ability to handle non-linearity and long-range dependencies.

Prophet has also gained popularity in mobility forecasting tasks due to its intuitive decomposition of time series components and ease of incorporating calendar effects. Studies such as Schulz et al. [15] have reported promising results using Prophet for short-term traffic and demand forecasting.

Although several works address passenger flow prediction in metro networks, few studies offer a systematic comparison of ARIMA, Prophet, and LSTM models applied to real-world data from complex systems like the London Underground. This research aims to fill this gap by evaluating these methods on a publicly available dataset of station usage [9].

2.5 Challenges in Passenger Flow Forecasting

Forecasting passenger flow in urban mobility systems involves several well-known challenges. Strong seasonal patterns on daily, weekly, and yearly scales, combined with the influence of holidays, strikes, and weather, introduce significant variability. Additionally, real-world data often contain missing values, noise, and anomalies, which require robust preprocessing. Furthermore, passenger flow exhibits complex non-linear temporal dynamics and long-range dependencies, motivating the use of advanced models such as Long Short-Term Memory networks (LSTMs) to capture these effects effectively [5, 23].

3 Materials and Methods

This section presents the data sources, preprocessing procedures, network construction, and forecasting methodology used to evaluate the research hypotheses introduced in Section 1. The experimental framework integrates heterogeneous datasets, models the London Underground as a spatial network, and systematically tests forecasting models for station-level passenger flow.

3.1 Data Collection and Integration

We constructed an integrated dataset combining temporal, spatial, and administrative information on the London Underground. The primary source is a publicly available dataset from Kaggle [9], providing annual records of passenger entries and exits by station.

Passenger flow estimates were calculated by scaling weekday, Saturday, and Sunday counts according to standard Transport for London (TfL) conventions:

$$\text{Total\_Flow} = 253(E_{\text{Week}}+X_{\text{Week}})+52(E_{\text{Sat}}+X_{\text{Sat}})+59(E_{\text{Sun}}+X_{\text{Sun}}), \tag{6}$$

where E and X denote entries and exits, respectively.

Station names were standardized to ensure consistency, missing numerical values were imputed using `IterativeImputer` from `scikit-learn`, and geospatial information (coordinates, fare zones, rail lines) was integrated via JSON files obtained from Doogal Ltd. [12]. Administrative boundaries and borough delineations were incorporated using shapefiles from the UK Office for National Statistics, specifically the MSOA 2021 dataset [20].

The final `GeoDataFrame` contains temporal, spatial, and administrative attributes for each station-year combination.

3.2 Network Construction

The spatial structure of the London Underground was modeled as an undirected graph using `networkx`. Nodes represent stations, enriched with attributes such as fare zone, administrative district, and borough. Edges were established based on geometric proximity to rail lines, respecting cartographic conventions.

Station names across datasets were aligned through exact and fuzzy matching, ensuring consistency between tabular and graph representations.

Two graph versions were generated: the complete network (Figure 2) and a filtered version excluding out-of-coverage stations (Figure 3). The filtered network overlaid on London boroughs is shown in Figure 4.

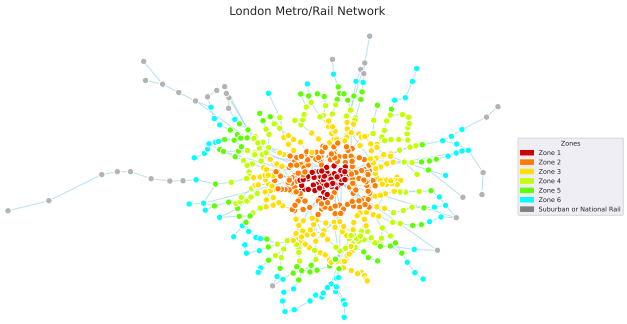

Figure 2: London Underground network based on geometric intersections between stations and rail lines.

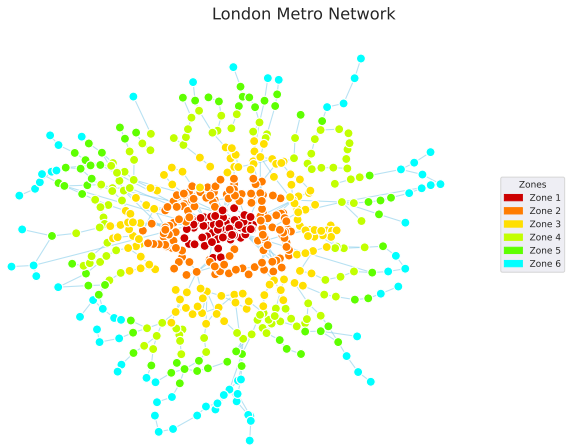

Figure 3: Filtered network excluding stations in fare zones not covered by the dataset.

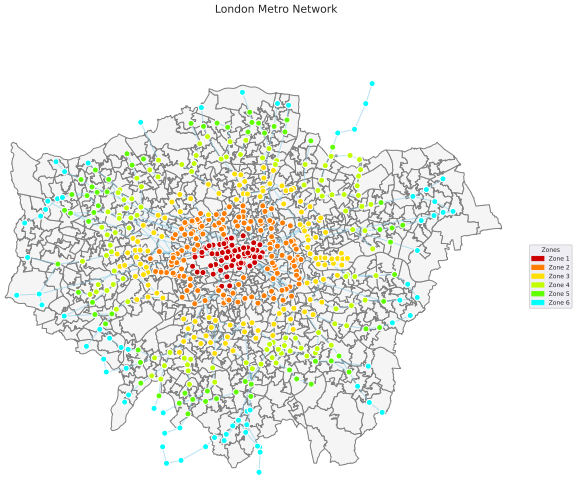

Figure 4: Filtered metro network overlaid on London's borough boundaries. This map contextualizes the spatial coverage of the network within administrative divisions.

3.3 Forecasting Procedure and Experimental Setup

Multiple experimental strategies were designed to evaluate the forecasting models in this study, addressing both the formulated research hypotheses and the need for robust, statistically sound model assessment.

1. Hypothesis-Driven Forecasting Experiments (Fixed Train-Test Splits). To directly address our research hypotheses regarding model performance, we adopted a conventional forecasting setup with fixed train-test splits, consistent with practices in related works [3, 25]. Specifically, for each station, the forecasting models were trained on historical data and evaluated on future, unseen periods according to the following configurations:

- **Experiment 1**: Training period: 2007–2013, Testing period: 2014–2017.
- **Experiment 2**: Training period: 2007–2015, Testing period: 2016–2017.

These experiments simulate realistic operational scenarios, where the entire historical dataset is available prior to model deployment, and forecasts are generated for strictly future periods. Model performance was assessed using standard error metrics (see Subsection 3.5), enabling controlled, direct comparison across models.

The following configurations were consistently applied across all forecasting models to ensure comparability:

- **ARIMA**: A classical auto-regressive integrated moving average model with fixed order $(p, d, q) = (1, 1, 1)$, providing a simple, interpretable statistical baseline.
- **Prophet**: Default settings with automatic changepoint detection and additive seasonality, as recommended in the official documentation for time series with periodic patterns.
- **LSTM**: A recurrent neural network with a look-back window of three time steps and 50 training epochs, representing a standard deep learning configuration without extensive hyperparameter tuning.

However, this approach produces only a single train-test split per experiment and station, limiting the number of independent forecast error observations. As a result, statistical tests such as the Wilcoxon signed-rank test lack sufficient power or cannot be reliably applied, restricting the strength of conclusions regarding model performance differences.

2. Cross-Validation for Robust Model Assessment. To complement the fixed-split experiments and provide a statistically sound model evaluation, a systematic time series cross-validation (CV) procedure was implemented. This procedure employs a rolling-origin approach with expanding training windows, as described in Subsection 3.4, generating multiple train-test folds per station.

The cross-validation procedure was configured as follows:

- The top two stations per fare zone were automatically selected based on average annual weekday passenger flow, ensuring spatial diversity in the evaluation.
- For each selected station, models were trained and evaluated using **six rolling-origin folds**, with a training window of five years and a test window of one year.
- All three forecasting models (ARIMA, Prophet, LSTM) were applied across all folds using the same configurations adopted in the fixed-split experiments, ensuring consistent model behavior across experimental strategies.

The full pipeline, encompassing station selection, cross-validation execution, and results aggregation, was fully automated, as implemented in the corresponding scripts. The output, containing detailed metrics for all stations, folds, and models, is stored in a consolidated CSV file, supporting subsequent statistical analysis and visual inspection of model performance.

3.4 Time Series Cross-Validation

To properly assess the generalization ability of forecasting models while preserving the temporal structure of the data, we employed time series cross-validation (CV) using a rolling-origin strategy with expanding training windows. Unlike random CV, this method prevents information leakage by ensuring that future observations are never included in the training set.

In each fold, the training set grows to incorporate additional historical data, always preceding the test set in time. Specifically, for fold k, the training set covers observations from t_0 to t_k, and the test set spans t_{k+1} to $t_{k+1} + H$, where H is the forecasting horizon. This setup guarantees model evaluation on strictly unseen future data.

Figure 5 illustrates the procedure. The process begins with a five-year training window, followed by a one-year test period. With each fold, the training set expands, maintaining a one-year test period, until the end of the time series is reached.

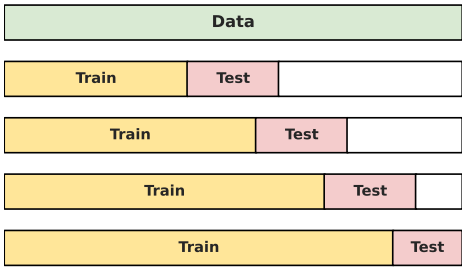

Figure 5: Rolling-origin time series cross-validation with expanding training windows and non-overlapping test periods. Source: Created by the author.

This approach provides multiple temporally consistent evaluation scenarios and is well-suited for forecasting applications where only past data is available during model training.

3.5 Evaluation Metrics

Forecast accuracy was assessed using standard error metrics widely applied in time series forecasting and regression tasks. The Mean Absolute Error (MAE), defined in Equation 7, measures the average magnitude of forecast errors in the same units as the target variable, providing an intuitive and scale-dependent performance indicator. The Root Mean Squared Error (RMSE), shown in Equation 8, penalizes larger errors more heavily due to its quadratic

formulation, making it particularly sensitive to significant deviations. The Mean Absolute Percentage Error (MAPE), presented in Equation 9, expresses forecast errors as a percentage of the true values, facilitating interpretability and comparisons across stations with different passenger flow magnitudes [7].

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|, \tag{7}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}, \tag{8}$$

$$\text{MAPE} = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{y_t - \hat{y}_t}{y_t}\right|. \tag{9}$$

Here, y_t represents the true observed value at time t, $\hat{y}_t$ the corresponding forecast, and N the number of forecasted time points.

Hypotheses and Statistical Tests. To formally evaluate our research hypotheses introduced in Section 1, we performed the following statistical tests focused on RMSE, following established practices in the statistical literature [4, 11, 14, 22]:

- $\mathcal{H}_1$ **(Model Superiority):** The LSTM model outperforms classical models (ARIMA and Prophet). Pairwise Wilcoxon signed-rank tests were applied between LSTM and each classical model across stations:

$$H_0 : \text{Median difference in RMSE} = 0, \; H_1 : \text{Median difference} \neq 0. \tag{10}$$

  Significant results ($p < 0.05$) support LSTM's superiority.
- $\mathcal{H}_2$ **(Spatial Variation in Prophet Performance):** Prophet model accuracy differs between zone 1 stations and others. Mann-Whitney U test compares Prophet RMSE distributions between these groups:

$$H_0 : \text{No difference in Prophet RMSE between zones}, \tag{11}$$

$$H_1 : \text{Difference exists}. \tag{12}$$

- $\mathcal{H}_3$ **(LSTM Performance Variation):** Differences in LSTM RMSE across stations are assessed using one-way ANOVA or, when assumptions are violated, the Kruskal-Wallis test:

$$H_0 : \text{No difference across stations}, \; H_1 : \text{At least one differs}. \tag{13}$$

All tests use a significance level of $\alpha = 0.05$. This framework ensures robust comparison of forecasting models.

4 Results and Analysis

This section presents and discusses the forecasting results obtained from both experimental strategies designed in this study. First, we report results under fixed train-test splits, simulating operational forecasting scenarios. Then, we present results from time series cross-validation (CV), providing a more statistically robust evaluation aligned with the hypotheses introduced in Section 1.

4.1 Fixed Train-Test Split Experiments

The fixed split experiments followed the configurations described in Subsection 3.3, simulating conditions where forecasting models are trained on historical passenger flow data (total annual entries and exits) and evaluated on strictly future periods.

A total of 18 forecasting records were generated: three models, three stations, and two train-test splits per station. Forecast accuracy was assessed using RMSE, MAE, and MAPE, as defined in Subsection 3.5, with all metrics expressed in number of passengers.

Table 1 summarizes the results for the three busiest zone 1 stations.

Table 1: Forecast results for top-3 zone 1 stations (fixed splits, no CV). Best results in bold.

Station	Model	TrEnd	RMSE	MAE	MAPE
Waterloo	Prophet	2013	47.0k	44.6k	14.7%
	ARIMA	2013	**19.3k**	**15.1k**	**4.8%**
	LSTM	2013	28.0k	23.3k	7.5%
	Prophet	2015	**16.1k**	15.8k	5.1%
	ARIMA	2015	20.7k	20.5k	6.6%
	LSTM	2015	17.2k	**13.5k**	**4.1%**
Oxford Circus	Prophet	2013	28.1k	25.4k	9.2%
	ARIMA	2013	**22.3k**	**18.1k**	**6.4%**
	LSTM	2013	23.3k	19.8k	7.1%
	Prophet	2015	42.5k	42.5k	16.8%
	ARIMA	2015	**24.6k**	**24.3k**	**9.6%**
	LSTM	2015	48.8k	48.4k	19.1%
King's Cross St. Pancras	Prophet	2013	13.5k	12.6k	4.4%
	ARIMA	2013	**6.9k**	**4.1k**	**1.5%**
	LSTM	2013	43.5k	39.2k	13.6%
	Prophet	2015	4.3k	4.1k	1.4%
	ARIMA	2015	**957**	**906**	**0.3%**
	LSTM	2015	8.7k	8.0k	2.7%

The results show that ARIMA consistently achieved the lowest forecast errors for most stations and configurations, particularly at King's Cross St. Pancras. At Waterloo, Prophet and LSTM became more competitive in Experiment 2, suggesting potential benefits from additional training data. However, LSTM performance was highly unstable at Oxford Circus and King's Cross, highlighting the limitations of deep learning models under small datasets and fixed-split scenarios.

To better illustrate model behavior under operational conditions, Figure 6 and Figure 7 present the ARIMA forecasts for King's Cross St. Pancras in both experimental setups. The plots demonstrate that, despite its simplicity, ARIMA effectively captured the general temporal dynamics and seasonal trends, resulting in accurate forecasts, especially with extended training data in Experiment 2.

Although these experiments approximate operational forecasting conditions, the small number of observations (18 records) limits the application of formal statistical tests and generalization of conclusions.

4.2 Time Series Cross-Validation Experiments

To provide a statistically sound comparison and formally test our hypotheses, we implemented a rolling-origin time series CV procedure with:

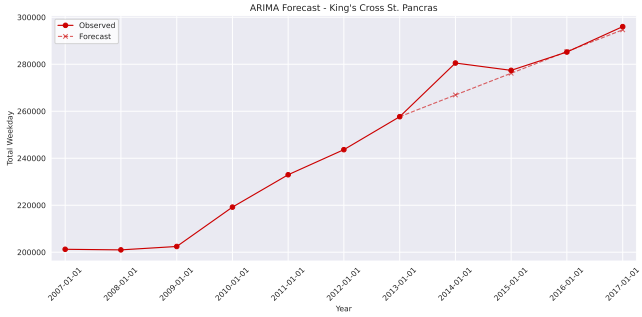

Figure 6: ARIMA forecast for King's Cross St. Pancras using training data from 2007 to 2013 (Experiment 1). The model captures overall trends with low forecast errors.

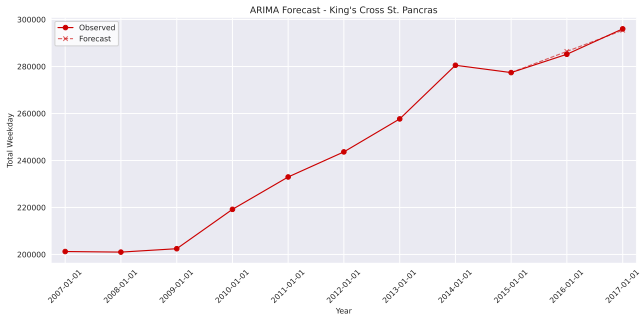

Figure 7: ARIMA forecast for King's Cross St. Pancras using training data from 2007 to 2015 (Experiment 2). Additional training data improves forecast alignment with observed values.

- **Station selection**: Two busiest stations per fare zone, ensuring spatial diversity.
- **CV configuration**: Six folds per station, with 5-year training windows and 1-year test windows.
- **Model settings**: Consistent parameters for ARIMA, Prophet, and LSTM; for LSTM, a look-back window of 3 time steps and 50 epochs.

This setup produced $2 \times 6 \times 3 \times 6 = 216$ forecasting records, enabling robust model comparisons and hypothesis testing.

Hypothesis $\mathcal{H}_1$: Model Superiority. Pairwise Wilcoxon signed-rank tests were applied between LSTM, ARIMA, and Prophet at each station. The RMSE distributions across models are illustrated in Figure 8. No significant differences ($p > 0.05$) were observed for most stations, indicating that LSTM did not consistently outperform classical models under CV, despite its theoretical ability to capture non-linear temporal patterns.

Hypothesis $\mathcal{H}_2$: Prophet Spatial Variation. A Mann-Whitney U test compared Prophet RMSE between zone 1 stations (central London) and other zones. As shown in Figure 9, significantly lower errors were observed in zone 1 ($p < 0.001$), suggesting Prophet benefits from greater flow stability and predictability in central areas.

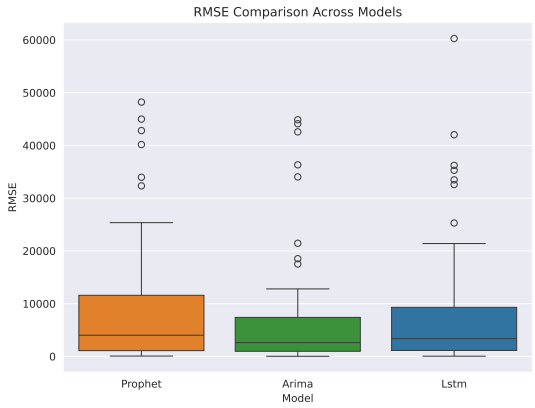

Figure 8: RMSE distribution across models from time series CV. Boxplots summarize RMSE variability for ARIMA, Prophet, and LSTM across all folds and stations.

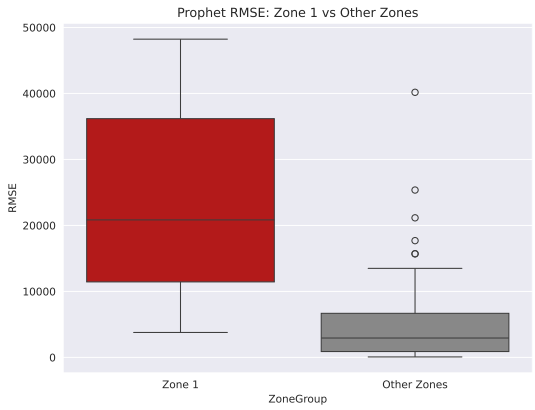

Figure 9: Comparison of Prophet RMSE between zone 1 and other zones. Lower RMSE in zone 1 indicates improved forecast accuracy in central stations.

Hypothesis $\mathcal{H}_3$: LSTM Performance Across Zones. A one-way ANOVA confirmed significant RMSE differences across zones for LSTM ($p < 0.001$). Figure 10 shows that LSTM performance was substantially better in central zones. Tukey HSD post-hoc tests (Figure 11) revealed that zone 1 exhibited significantly lower errors compared to all outer zones.

These results confirm that spatial factors, particularly fare zone, exert a strong influence on forecast accuracy. While LSTM demonstrated competitive performance in data-rich, stable contexts, its sensitivity to data scarcity and heterogeneity remains a key limitation for its application in large, heterogeneous transit networks.

5 Conclusion and Future Work

This study presented a comprehensive comparative analysis of classical statistical methods (ARIMA, Prophet) and deep learning

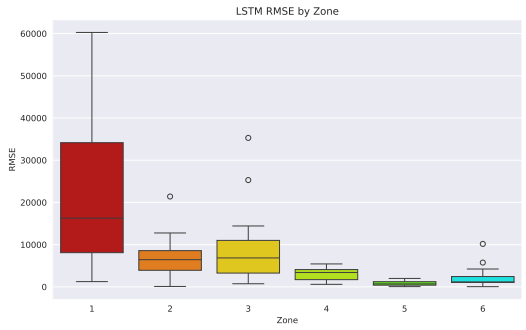

Figure 10: LSTM RMSE distribution across fare zones. Lower forecast errors in central zones reflect improved model performance under more stable conditions.

Figure 11: Tukey HSD post-hoc test for LSTM RMSE across zones. Statistically significant differences confirm better LSTM performance in zone 1.

models (LSTM) for passenger flow forecasting in the London Underground. Our results offer valuable insights into the applicability, limitations, and potential of these approaches in complex urban mobility systems.

Three research hypotheses guided this investigation:

- $\mathcal{H}_1$: LSTM models can outperform classical approaches by capturing complex non-linear and long-range temporal dependencies.
- $\mathcal{H}_2$: Forecasting models that explicitly incorporate seasonality and calendar events (e.g., Prophet) achieve improved accuracy in specific contexts, particularly in central areas.
- $\mathcal{H}_3$: Spatial heterogeneity affects model performance, with better results in stable, high-demand zones.

The fixed train-test split experiments revealed that, under limited data conditions, classical models such as ARIMA often deliver competitive or superior performance, especially for short-term forecasts and in stations with more linear temporal patterns. Prophet demonstrated improved performance in central London stations (zone 1), confirming its suitability for data-rich and stable contexts.

Time series cross-validation provided a more robust evaluation. The Wilcoxon signed-rank tests failed to confirm the overall superiority of LSTM ($\mathcal{H}_1$ not supported), likely due to the small dataset size and limited temporal resolution. In contrast, significant spatial effects were identified. Prophet achieved lower RMSE in zone 1 compared to outer zones ($\mathcal{H}_2$ supported), and LSTM exhibited significantly better performance in central areas, as confirmed by ANOVA and Tukey HSD post-hoc tests ($\mathcal{H}_3$ supported).

These findings underscore the significant impact of spatial factors and data granularity on forecasting performance, as well as the limitations of deep learning models when applied to small, aggregated datasets.

Future Work. Several avenues can be explored to advance this research:

- **Data Granularity:** Increasing the temporal resolution (e.g., using monthly or daily passenger flow data) would expand the number of observations, providing better conditions for training deep learning models and more robust statistical testing. In the absence of real high-frequency data, synthetic data could be generated to simulate such scenarios; however, this approach introduces strong assumptions and may fail to capture the true complexity and variability of passenger behavior.
- **Hyperparameter Optimization:** Systematic exploration of hyperparameters, particularly for LSTM (e.g., look-back window, network depth, learning rate), is essential to fully assess the potential of deep learning approaches.
- **Graph-Based Forecasting:** The London Underground was modeled as a static spatial graph in this study. A promising extension is the use of spatio-temporal Graph Neural Networks (GNNs) for forecasting passenger flow in the network, integrating both spatial and temporal dependencies, as highlighted in recent works such as [8].
- **Multivariate Modeling:** Incorporating exogenous factors such as weather, events, socioeconomic indicators, and spatial features from the final `GeoDataFrame` (e.g., zone, borough, administrative boundaries) could improve forecasting performance and model robustness.
- **Broader Network Analysis:** Applying similar methodologies to other urban rail systems or multimodal transportation networks would validate the generalizability of the findings and reveal system-specific dynamics.

Overall, this study reinforces the importance of context-aware model selection for urban mobility forecasting and motivates the continued development of advanced, graph-based, and deep learning techniques tailored to the challenges of large-scale transit networks.

References

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. 2015. *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

[2] Chuanlong Chen, Wenzhen Zeng, Wenbo Du, Xiaojun Fu, Xinyu Wang, and Xiaoming Tan. 2019. A hybrid deep learning framework for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* 20, 11 (2019), 4264–4275. doi:10.1109/TITS.2018.2876889

[3] Wushuang Fan. 2024. Short-term Passenger Flow Prediction of Metro based on ARIMA and LSTM Models. *Highlights in Science, Engineering and Technology* 105 (2024), Article 23030. doi:10.54097/b7ej2k54 Acessado em 1º de julho de 2025.

[4] Ronald A Fisher. 1925. Statistical methods for research workers. *Genesis Publishing Pvt Ltd* (1925).
[5] Shengfeng Guo, Yuan Lin, Nanning Feng, and Xian Song. 2019. Deep learning for Intelligent Transportation Systems: A Survey of Emerging Trends. *IEEE Transactions on Intelligent Transportation Systems* 21, 10 (2019), 3955–3973. doi:10.1109/TITS.2019.2914600
[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[7] Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: Principles and Practice*. OTexts. https://otexts.com/fpp2/
[8] Ming Jin, Huan Yee Koh, and et al. 2024. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
[9] Kaggle Dataset. [n. d.]. London Underground Stations Usage (2007-2017). https://www.kaggle.com/datasets. Accessed: 2025-07-01.
[10] Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. 1137–1145.
[11] William H Kruskal and W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *J. Amer. Statist. Assoc.* 47, 260 (1952), 583–621.
[12] Doogal Ltd. [n. d.]. London Stations and Rail Lines JSON Data. https://www.doogal.co.uk/london_stations#google_vignette. Accessed: 2025-07-04.
[13] Yisheng Lv, Yishan Duan, Wenlong Kang, Zhixuan Li, and Fei-Yue Wang. 2015. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2015), 865–873. doi:10.1109/TITS.2014.2345663
[14] Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* 18, 1 (1947), 50–60.
[15] Johanna Schulz, Stefan Schneider, and Gunter Schuh. 2020. Forecasting mobility demand using time series models and event detection. *Transportation Research Procedia* 47 (2020), 61–68. doi:10.1016/j.trpro.2020.03.084
[16] B. L. Smith, B. M. Williams, and R. K. Oswald. 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies* 10, 4 (2002), 303–321. doi:10.1016/S0968-090X(02)00009-8
[17] Sean J. Taylor and Benjamin Letham. 2018. Forecasting at scale. *The American Statistician* 72, 1 (2018), 37–45.
[18] Transport for London. 2019. *Annual Report and Statement of Accounts 2018/19*. Technical Report. Transport for London. https://tfl.gov.uk/corporate/publications-and-reports/annual-report-and-statement-of-accounts
[19] Transport for London. 2025. Tube. https://tfl.gov.uk/modes/tube/ Acesso em: 1 jul. 2025.
[20] UK Office for National Statistics. [n. d.]. Middle Layer Super Output Areas (MSOA) 2021 Shapefiles. https://geoportal.statistics.gov.uk/datasets/middle-layer-super-output-areas-december-2021-boundaries. Accessed: 2025-07-04.
[21] Vukan R. Vuchic. 2005. *Urban Transit: Operations, Planning, and Economics*. Wiley.
[22] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1, 6 (1945), 80–83.
[23] Jiatong Zhang, Yujie Zheng, and Dong Qi. 2019. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *IEEE Transactions on Intelligent Transportation Systems* 20, 12 (2019), 4734–4744. doi:10.1109/TITS.2018.2876806
[24] Qingqing Zhang, Ying Li, and Jian Chen. 2020. Deep learning for short-term passenger flow forecasting in urban rail transit systems. *Transportation Research Part C: Emerging Technologies* 118 (2020), 102716. doi:10.1016/j.trc.2020.102716
[25] Wei Zhang, Yifan Wang, Yu Zhang, Zhiqiang Zhang, and Xuefeng Zhang. 2023. A novel deep learning-based approach for predicting the remaining useful life of lithium-ion batteries. *Journal of Network and Computer Applications* 202 (2023), 105935. doi:10.1016/j.jnca.2023.105935

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009