IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2025, accepted 8 September 2025, date of publication 12 September 2025, date of current version 19 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3609530

RESEARCH ARTICLE

Spatio-Temporal Forecasting of Bus Arrival Times Using Context-Aware Deep Learning Models in Urban Transit Systems

OSMAN KAYA AND MUSTAFA UTKU KALAY
Department of Computer Engineering, Yıldız Technical University, 34349 Istanbul, Türkiye
Corresponding author: Osman Kaya (osman.kaya@delta-yazilim.com)

This research was supported by the Council of Higher Education (YÖK) of Turkey under the 100/2000 Ph.D. Scholarship Program.

ABSTRACT Accurate forecasting of bus arrival times is critical for enhancing the reliability and efficiency of public transportation systems. However, complex factors such as traffic congestion, weather conditions, and temporal variability make this task challenging. In this study, we propose a context-aware hybrid deep learning framework that integrates Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer architectures to predict stop-level bus travel times. The model is trained on a comprehensive dataset collected from 500 bus routes in Istanbul, spanning six months and incorporating real-time GPS data, GTFS schedules, and hourly weather attributes. A hybrid trend component is selectively introduced for data groups with fewer than 1000 samples to mitigate overfitting under sparse data conditions. Experimental results show that the trend-augmented LSTM model outperforms baseline architectures, achieving up to 28% improvement in MAE. The best-performing model yields an MAE of 2.97 minutes, a MAPE of 14.79%, and an R^2 value of 0.9272 across all test routes. Furthermore, condition-based evaluations demonstrate that prediction accuracy varies significantly across different time blocks, weather conditions, and day types. The proposed approach is both scalable and adaptable, offering a robust solution for real-time transit forecasting in complex urban environments.

INDEX TERMS Bus arrival time prediction, spatio-temporal modeling, deep learning, LSTM, hybrid model, real-time GPS data, weather-aware forecasting.

I. INTRODUCTION

Urban public bus transportation plays a vital role in the daily mobility of metropolitan residents. Accurate forecasting of bus arrival times is critical to ensure passenger satisfaction and maintain operational reliability. However, real-world transit systems are subject to a variety of dynamic and stochastic factors—such as traffic congestion, adverse weather, time-of-day fluctuations, and irregular passenger demand—that often disrupt schedule adherence [1], [2]. These uncertainties complicate travel planning and diminish overall trust in public transport services.

In recent years, various approaches have been developed to address the bus arrival time prediction problem. Traditional machine learning models rely heavily on handcrafted features and tend to underperform when capturing nonlinear temporal patterns [3], [4]. Deep learning architectures, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer models, have shown superior performance by modeling sequential and temporal dynamics [1]. However, many of these models lack contextual awareness, omitting critical variables such as weather, time blocks, and day type, which significantly influence transit behavior. Nonlinear prediction approaches have also been explored in recent studies [5], but they often lack robustness under dynamic transit conditions.

Another key limitation in prior work is the degradation of performance in data-sparse regimes. Although transit datasets may contain millions of samples overall, specific contextual segments (e.g., rainy weekend mornings or late-night

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Guidi.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

weekday trips) are often underrepresented, increasing the risk of overfitting. Additionally, fixed interval-based time segmentation fails to capture the true temporal variability present in daily transit operations.

Graph-based approaches, such as Graph Neural Networks (GNNs), have also been explored for spatiotemporal forecasting due to their ability to model topological dependencies. While these methods are effective when the underlying graph structure is stable and well-defined [6], [7], they may be less suitable for transit systems with dynamic or incomplete connectivity.

To overcome these challenges, we propose a hybrid deep learning framework that integrates LSTM, GRU, and Transformer components with a trend-enhancement mechanism selectively applied to sparse data segments. The model incorporates contextual variables—including hourly weather conditions, day type, and empirically derived time-of-day blocks—to improve both predictive accuracy and robustness across diverse operating conditions.

Unlike traditional approaches that apply trend logic uniformly, our selective design enhances generalization under low-sample conditions while maintaining computational efficiency. Furthermore, the interpretability of the model is improved by enabling post hoc analysis of trend feature usage.

The main contributions of this study are as follows:

- A hybrid architecture combining LSTM, GRU, and Transformer components to capture both sequential dependencies and condition-aware variability.
- A selective trend mechanism that improves performance in underrepresented data segments.
- Contextual feature integration including weather conditions, day type, and data-driven time blocks.
- A comprehensive evaluation on six months of real-world data from 500 urban bus routes, including condition-specific assessments and baseline comparisons.

A structured comparison of previous studies is presented in Table 1, highlighting key distinctions in data sources, contextual awareness, and model limitations.

To illustrate the practical effectiveness of our proposed model, Figure 1 presents a representative stop-by-stop comparison of predicted and actual travel times on a selected bus trip. This example highlights how well the model captures the sequential dynamics of urban transit, even before diving into detailed experimental results.

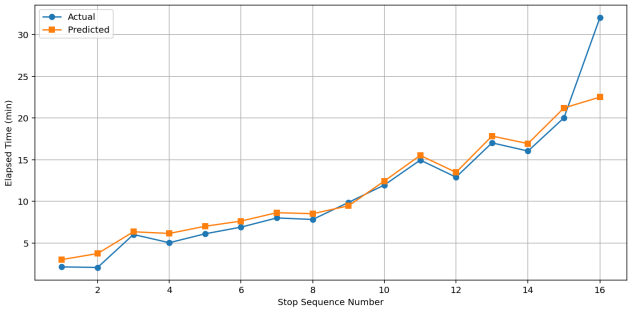


FIGURE 1. Visual comparison of predicted vs actual travel times.

II. RELATED WORK

In recent years, bus arrival-time prediction has been the focus of numerous academic studies. In earlier studies, historical averages and statistical models were predominantly used, which generally neglected environmental factors, such as traffic, weather, or time of day. For instance, initial prediction systems developed using AVL data and Kalman filters achieved only limited success [8].

Over time, machine-learning models have become more common in this field. Techniques, such as regression, support vector machines, and decision trees, have provided more flexible solutions for bus travel time prediction. However, these methods often fail to capture temporal patterns embedded in transit data [9], [10].

Long Short-Term Memory (LSTM) networks are among the most effective approaches for modeling temporal sequences. The LSTM architecture has become a widely used backbone in this domain [11] because it allows previous time-dependent information to be retained by modeling the durations between successive stops. Numerous successful applications have employed LSTM for real-time bus prediction using GPS data [12], [13]. Similarly, LSTM-based forecasts have been extended to incorporate traffic conditions and sensor data [14]. In some studies, Gated Recurrent Unit (GRU) networks have also been used, offering faster training owing to fewer parameters [15].

In recent years, attention-based transformer architectures have emerged as powerful alternatives to time-series forecasting in public transportation. Their ability to learn long-range dependencies in parallel has contributed to their high accuracy in travel time estimation [16]. These models have been expanded to handle multiple data sources, including traffic, weather, and schedule information, and have been successfully applied to urban bus systems [17], [18]. Furthermore, attention mechanisms have been integrated with Graph Neural Networks (GNNs), yielding significant improvements in multi-stop prediction scenarios [19].

The influence of environmental conditions (e.g., weather, time, and social factors) on the prediction performance has been extensively examined. In particular, weather conditions play a critical role in prediction errors [20]. Studies have shown that hybrid deep-learning models that incorporate environmental variables result in meaningful improvements in forecasting accuracy [21]. Additionally, factors such as social events or weekday–weekend differences have been modeled within multitask learning frameworks to further enhance the predictive performance [22].

Several systematic reviews have explored the application of deep learning in transportation systems. These studies commonly focus on data density, pattern complexity, and integration of external factors into predictive models. Literature highlights that deep learning has been widely

adopted for transportation forecasting both theoretically and practically [23], [24].

Recent efforts in spatiotemporal forecasting have increasingly turned to Graph Neural Networks (GNNs), owing to their strength in modeling complex topological structures and spatial dependencies across networks. For example, an auto-configured explainable GNN architecture has been proposed for multi-site pollution prediction using spatiotemporal sensor networks [6]. Another study demonstrated notable improvements in photovoltaic power forecasting using GNN-based models [7]. While these approaches effectively capture spatial correlations in structured graphs, our method is particularly designed for scenarios where a rigid or static graph representation of spatial topology (e.g., bus-stop connectivity) may be unavailable, unstable, or difficult to generalize in dynamic transit networks.

Instead of relying on explicit graph structures, we incorporate route-level spatial information—such as `stop_sequence`, `route_id`, and `direction_id`—directly as input features extracted from GTFS data and structured via SQL-based preprocessing. This enables the model to learn sequential and spatial dependencies implicitly through a context-aware deep learning framework. Our approach dynamically integrates temporal and exogenous features, including weather conditions, time-of-day blocks, and day types, offering a flexible alternative to graph-based models in large-scale, real-world transit forecasting scenarios.

Graph-based convolutional networks (GCNs) and multichannel data-processing architectures have been utilized in models that process spatiotemporal relationships [25]. Multimodal deep-learning frameworks have been developed by combining traffic, scheduling, and weather data [26]. Additionally, LSTM implementations enhanced with microwave sensors under the ''urban computing'' approach aimed to unify different data types for greater predictive power [27]. A comprehensive review of public transportation data analysis using machine learning supports these developments [28].

Hybrid deep learning models have demonstrated strong performance in environmental forecasting tasks, especially under dynamic temporal conditions. [29], [30], [31]

These applications support the applicability of such architectures in domains with contextual and seasonal variability, which aligns with the goals of our study on transit prediction.

In light of the above, a structured comparison of prior studies is provided in Table 1, which highlights differences in data sources, contextual awareness, and major limitations.

Compared to the above approaches, our proposed method introduces a selective trend mechanism tailored for sparse data scenarios and incorporates multiple contextual factors (e.g., weather, time-of-day, day-type), which are often neglected in previous studies.

Although these studies have provided valuable contributions, many models focus solely on a single data source (typically, historical trip duration), whereas contextual factors are only partially considered. Moreover, critical preprocessing steps, such as stop-level alignment of real-time GPS data and noise filtering, are often overlooked or assumed.

TABLE 1. Comparison of recent bus arrival time prediction studies.

Ref	Data Sources	Context-Aware	Limitations
[7]	GPS only	✗	Ignores weather and time-of-day.
[9]	GPS + Microwave sensor	✗	Requires additional hardware; no contextual adaptation.
[10]	GPS + Traffic signals	✗	No exogenous factors like weather or schedule variability.
[12]	GPS + GTFS	✗	No day-type or time-block segmentation.
[13]	GPS + Historical trips	✗	Sensitive to irregular intervals; lacks weather or calendar context.
[14]	GPS + GTFS	✓	Partial context-awareness; lacks condition-specific analysis.
[15]	GPS + Meteorological data	✓	Lacks spatial correlation modeling.
[16]	GPS + GTFS + Weather	✓	No handling of sparse data groups.
[18]	GPS + GTFS + schedule	✓	CNNs struggle with temporal dependencies.
[21]	GPS only	✗	Lacks sequence modeling; poor generalization in dynamic conditions.
[26]	GPS + Weather	✓	Lacks dynamic temporal representation; shallow feature modeling.
[28]	GTFS + Seasonality	✓	Rule-based model underperforms under anomalies or irregularities.
[31]	GPS + Spatial Graph	✓	Requires pre-defined graph; high computational cost.
[37]	GPS + Time Block	✗	CNNs underfit sequential time-dependency.
[41]	GTFS + Schedule only	✓	Context-limited; ignores real-time irregularities.
[45]	GPS + GTFS + GNN features	✓	Computationally expensive; difficult to interpret.
[46]	GPS + Schedule	✓	Attention-based RNN; struggles with irregular patterns.
Ours	GPS + GTFS + Weather	✓	Selective trend logic improves sparse-group prediction and robustness across varying conditions.

This study differentiates itself from previous research in four main ways:

(i) Although many existing models rely exclusively on historical data or single-source inputs (e.g., GPS), this study integrated a multi-source dataset, including real-time GPS, GTFS [32], hourly weather data, and contextual information.

(ii) Whereas most prior studies process entire datasets under a single model and report only average accuracy, our approach explicitly incorporates contextual variables, such as time block, weather conditions, and day type, as model inputs to enable detailed analyses.

(iii) In contrast to earlier studies that tested LSTM, GRU, or transformers independently or on different datasets, our study comparatively evaluated all architectures under the same data structure and configuration to ensure a fair assessment.

(iv) The original contribution of this study is the development of a hybrid input design that selectively incorporates trend features only for low-sample groups

($N < 1000$). This approach provides adaptive modeling based on data density and prevents overfitting, leading to improvements in the accuracy of up to 28%.

With these aspects, the study stands apart from previous works not only in terms of average accuracy metrics but also by offering a condition-based performance evaluation framework that is both generalizable and practical for real-time public transit applications.

III. PROPOSED METHOD

A. DATASET AND PREPROCESSING

The dataset used in this study was obtained through open web service endpoints provided by the Istanbul Electric Tram and Tunnel Company (IETT) [33]. The data included real-time GPS positions reported every 40 s for each bus, along with route codes, trip IDs, stop sequence numbers, vehicle identifiers, and geographical coordinates of the corresponding stops. The dataset covers operations recorded between August 2024 and February 2025, and is limited to daily hours between 06:00 and 24:00.

Hourly weather data were retrieved from the Visual Crossing platform (https://www.visualcrossing.com, accessed February 2025) and matched to nearest-stop transition times. Weather attributes included temperature, humidity, precipitation, wind speed, snow presence, visibility, and categorical weather conditions (e.g., clear, rainy).

The dataset covers 500 bus routes. Each route included 10–101 stops along a unidirectional path. Routes with more than 500 daily trips were excluded to maintain balanced distribution. Over a six-month period, the dataset grew to millions of records containing highly diverse and dense transit data.

During pre-processing, several key steps were performed to integrate the spatial and temporal dimensions. First, bus movements were grouped by trip ID and each trip was transformed into a temporal movement sequence. These sequences were stored using the `tgeompointseq` data type in MobilityDB, which supports spatiotemporal data representation [34].

Data cleaning was applied to eliminate GPS anomalies, incorrectly matched coordinates, reversed movements, and physically impossible patterns (e.g., moving more than 600 meters in one second). This process was executed using a combination of automated SQL procedures and Python-based scripts [35], [37], [38]. Missing data imputation, statistical consistency checks, and the synchronization of multiple sources have been conducted using applied data-cleaning techniques [36].

Following these procedures, a refined and analysis-ready dataset is obtained. The cleaned records were organized into a machine-learning-compatible structure, with the selected variables listed in Table 2.

1) FEATURE ENCODING AND INPUT SHAPES

Categorical features (day type, weather condition, and time block) were label-encoded and subsequently normalized into the range [0,1]. Continuous features (e.g., elapsed time, stop order, and trend slope) were also normalized using the MinMaxScaler. Consequently, each input sequence was represented as a six-dimensional vector of shape (1,6). The input shapes reported in Table 3 (e.g., (1,5), (1,4)) correspond to feature subsets provided to specific modules (e.g., trend-informed vs. baseline components).

For full details on the data acquisition, cleaning, trajectory construction, and weather integration procedures, including spatial filtering, stop alignment, and MobilityDB usage, is provided in Appendix A (submitted as supplementary material for review purposes).

TABLE 2. Grouped features: field name, description, and category.

Field Name	Description	Group
Input Features		
stop_sequence	Sequence number of the stop on the route	Input
direction_id	Route direction	Input
day_type	Day type (Weekday, Saturday, Sunday)	Input
time_block	Normalized time block (e.g., 06:00)	Input
temp	Temperature (°C)	Input
humidity	Humidity (%)	Input
precipitation	Precipitation amount (mm)	Input
wind_speed	Wind speed (km/h)	Input
snow_flag	Snow presence (0 or 1)	Input
visibility	Visibility (km)	Input
weather_condition	Weather condition class (e.g., Clear, Rainy)	Input
elapsed_time_min	Time passed since the previous stop (minutes)	Input
Auxiliary Metadata		
route_code	Bus route identifier	Auxiliary
vehicle_id	Vehicle identifier (e.g., license plate)	Auxiliary
IDate, Time	Date and time of the observation	Auxiliary
latitude, longitude	Stop location coordinates	Auxiliary
stop_name	Name of the bus stop	Auxiliary
Target Variable		
target_time_min	Estimated travel time to the next stop (minutes)	Target

B. MODELING APPROACH

Deep-learning-based models have become widely used for predicting bus arrival times in public transportation systems. In the literature, LSTM and Transformer architectures have been used to learn temporal dependencies from time-series data [16], [17], [39], [40]. However, most previous studies relied on single data sources, such as GPS, and paid limited attention to the role of environmental and contextual variables [41]. Some encoder-decoder models make direct predictions from input-output sequences, but often ignore spatial relationships between stops or the structural characteristics of the route [42], [43]. Moreover, many comparative studies have not evaluated the effects of dataset size, contextual features, or intra-route patterns in depth [44], [45].

In this study, we propose a context-aware deep learning architecture for bus arrival time prediction. The model processes the input features, including the stop sequence, time-related variables (day type and time block), weather attributes, and trip ID. It captures both the temporal and contextual patterns across trips. Sequential dependencies are modeled using LSTM or GRU layers, and when required, long-term relationships are learned via transformer-based encoder layers with attention mechanisms. The output layer produces the estimated travel time from the current stop to the next stop, expressed in min.

All three models were trained on the same dataset with identical input-output configurations, ensuring a fair evaluation. The input features include the elapsed time from the previous stop and contextual variables (day type, time block, and weather), and the target output is the estimated time to the next stop.

For training, the Mean Absolute Error (MAE) was used as the loss function, and the Adam optimizer was applied. Each model was trained for 50 epochs with a batch size of 64. Early stopping was employed, and training was stopped if the validation loss did not improve for five consecutive epochs. The dataset was split into 70% training, 15% validation, and 15% testing.

1) DATA SPLITTING AND LEAKAGE PREVENTION

The dataset was partitioned into training (70%), validation (15%), and test (15%) sets using trip_id × service_date keys to ensure that trips or contiguous sequences never straddled across different sets. A fixed random seed (42) was applied for reproducibility. Normalization statistics were computed only on the training set and consistently applied to the validation and test sets, preventing temporal leakage.

Model evaluation goes beyond average error metrics and includes condition-specific assessments based on the weather, time of day, and day type. This design reveals how each model performs, not only on average but also under specific, realistic transit conditions.

The comparative performance of the methods is empirically tested and analyzed in the next section.

IV. EXPERIMENTAL STUDY

A. TEST DESIGN AND SCENARIOS

The experimental design consisted of two main phases. In the first phase, each bus route is analyzed independently, and a separate model is trained for each route using a dataset that includes various contextual variables, such as weather conditions, time of day, and day type. This route-level modeling strategy serves multiple purposes: it allows for the evaluation of the base performance of each architecture, supports the inspection of underlying temporal patterns in travel time, and helps identify potential issues in data pre-processing.

During this phase, it was observed that, for some routes, the travel times between stops exhibited consistent upward or downward trends at specific hours. To account for such patterns, a linear regression model was applied to each route time series to extract numeric trend values. This trend was then added to the input features to create a ''trend-informed'' model variant. To assess the effect of this feature, an equivalent ''trend-free'' model was trained using the same dataset, excluding the trend input. This comparative setup enabled a direct evaluation of the impact of trend features under different data conditions.

In the second phase, the data from all routes were combined into a single aggregated dataset. A general model was then trained using this unified dataset to evaluate the influence of global conditions on prediction performance. This model is particularly useful for condition-based assessment. By categorizing the test results according to contextual features, such as weather type, day type, and time block, the analysis provides insights into not only the overall accuracy but also the behavior of the model under specific real-world scenarios.

Thus, the two-phase experimental structure provides a comprehensive evaluation. The first phase captures route-specific dynamics and trend behaviors, whereas the second phase enables generalization and performance comparison under varying contextual conditions across the entire transit network.

Unlike prior studies that either ignore long-term trends or apply them uniformly across all samples, our model introduces a selective trend-enhancement logic. This mechanism is only activated when the number of training samples for a given contextual segment (e.g., weather-time block) falls below a predefined threshold (1000 samples in our experiments). This strategy mitigates the risk of overfitting under data sparsity while avoiding unnecessary complexity in data-rich scenarios.

B. EVALUATION METRICS

Model performance was evaluated using five widely accepted metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). These measures provide complementary perspectives on error magnitude, model robustness, and generalization capacity.

The Mean Absolute Error (MAE) was used to calculate the average absolute difference between actual and predicted values. It is defined as:

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

This provides a direct indication of the average magnitude of the prediction errors regardless of their direction.

The Root Mean Square Error (RMSE) penalizes larger errors more heavily by squaring the residuals before averaging, and is defined as

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

Although RMSE is effective in capturing large deviations, it may exaggerate the impact of outliers, making MAE more reliable in scenarios with skewed distributions [39], [47].

The Coefficient of Determination (R^2) quantifies the proportion of variance in the actual values, which is explained by the model predictions and is given by

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

where $\bar{y}$ is the mean of the actual values. R^2 values close to 1 represent strong predictive power, whereas values below 0 suggest that the model underperforms, even as a naive mean predictor.

The Mean Absolute Percentage Error (MAPE) measures the average percentage error:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{4}$$

Although MAPE is intuitive and unit-free, it can be unstable when actual values are close to zero.

To address this, the Symmetric Mean Absolute Percentage Error (SMAPE), which adjusts the denominator, is used:

$$SMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \tag{5}$$

1) CONSISTENCY OF METRICS

All performance metrics (MAE, RMSE, R^2, MAPE, SMAPE) are reported in a consistent order across the text, tables, and figures to ensure clarity and comparability.

This makes the SMAPE more robust and bounded between 0% and 200%, particularly in fluctuating or small-value scenarios.

By combining these five metrics, this study ensures not only a fair comparison between models but also a detailed understanding of their behavior under varying conditions.

C. HYPERPARAMETER TUNING AND SELECTION

To optimize the performance of each deep learning model, we conducted a structured hyperparameter tuning process for LSTM, GRU, and Transformer architectures. The parameters considered included the number of hidden units, dropout rate, batch size, learning rate, number of training epochs, and early stopping patience.

A randomized search strategy was initially applied over predefined ranges, followed by manual fine-tuning guided by validation MAE. An early stopping mechanism with a patience value of 10 was employed to prevent overfitting. The final selected values for each model are listed in Table 3.

All three models were trained using the same contextual input structure and dataset split for fairness. The LSTM and GRU architectures each consist of one recurrent layer with 128 hidden units followed by a dense output layer (ReLU for LSTM, linear for GRU). The Transformer architecture includes two encoder layers, each with two attention heads (64 dimensions), a 128-unit feedforward network, and dropout applied post-attention and post-feedforward blocks. Layer normalization is used after each sub-block in the Transformer. No L2 regularization or batch normalization was applied to the recurrent models, as early stopping and dropout were sufficient to prevent overfitting in preliminary tests.

TABLE 3. Selected hyperparameter and architecture settings for each model.

Parameter	LSTM	GRU	Transformer
Recurrent Units	128	128	64
Recurrent Layers	1	1	–
Dense Layers	1 (ReLU)	1 (Linear)	1 (Linear)
Input Shape	(1, 5)	(1, 5)	(1, 4)
Activation (Recurrent)	tanh / sigmoid	tanh / sigmoid	–
Feedforward (Transformer)	–	–	2-layer FFN (128 units)
Attention Heads	–	–	2 heads (64-dim)
Dropout	0.2	0.2	0.1 (FFN + MHA)
Layer Normalization	No	No	Yes (per encoder block)
Batch Size	64	64	32
Epochs	50	50	50
Early Stopping	Yes (patience=10)	Yes (patience=10)	No
Loss Function	MSE	MSE	MSE
Optimizer	Adam	Adam	Adam
Regularization (L2)	Not applied	Not applied	Not applied
Parameters	∼18k	∼14k	∼2.1M

D. STATISTICAL CONFIDENCE ANALYSIS

To provide a deeper understanding of the model's robustness in high-performance scenarios, we calculated 95% confidence intervals for all major metrics (MAE, RMSE, R^2, MAPE, and SMAPE). These results, summarized in Table 4, confirm the statistical reliability of the LSTM model under strong predictive performance.

TABLE 4. Performance metrics with 95% confidence intervals for LSTM model.

Metric	Mean	95% CI Lower	95% CI Upper
MAE	3.0650	2.9471	3.1829
RMSE	6.8909	6.5483	7.2336
R^2	0.9457	0.9420	0.9494
MAPE	10.6524	10.4033	10.9016
SMAPE	9.3961	9.1912	9.6011

To quantify predictive uncertainty, we applied Monte Carlo (MC) dropout during inference. Specifically, we performed 100 stochastic forward passes through the LSTM model with dropout activated at test time.Figure 2 presents a subset of predictions along with 95% confidence intervals derived from the standard deviation across samples. As observed, the model produces narrow uncertainty bands for most instances, with wider intervals where prediction deviations are higher, particularly around outliers or abrupt changes in travel time.

E. FEATURE IMPORTANCE AND MODEL INTERPRETABILITY

To enhance the interpretability of our proposed LSTM model, we applied SHAP (SHapley Additive exPlanations) analysis using contextual input features only—specifically, day type, time block, and weather condition. The SHAP summary plot (Figure 3) illustrates the contribution of each feature to the model's predictions. The results indicate that time-of-day and

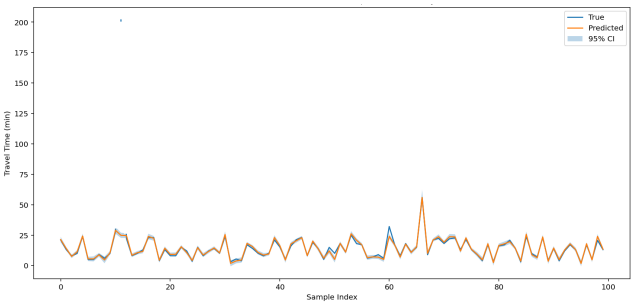


FIGURE 2. LSTM-based predictions with MC Dropout uncertainty. The shaded area denotes the 95% confidence interval.

weather conditions have the highest impact on model output, consistent with real-world expectations of traffic congestion during peak hours and adverse weather. Day type (e.g., weekday vs. weekend) shows a smaller but interpretable influence. This confirms that the model has learned meaningful associations between contextual inputs and predicted travel times, increasing trust in its decision-making process.

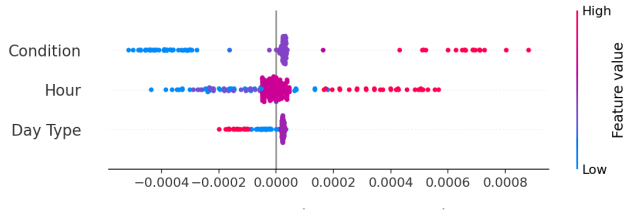


FIGURE 3. SHAP value (Impact on model output)).

The results indicate that `time_block` and `weather_condition` are the most influential features, aligning with domain expectations of traffic congestion during peak hours and delays during adverse weather. The `day_type` feature showed a moderate but interpretable influence, reflecting variations in service frequency and urban mobility patterns across weekdays and weekends.

F. AGGREGATED PERFORMANCE ANALYSIS ACROSS ROUTES

Extensive experiments have been conducted on individual bus routes with various characteristics, including short and long durations, and different numbers of stops. After validating the modeling structure, the final model was applied to all the routes.

For example, Table 5 presents the weighted average metric results for Route 12A based on test data. When the trend feature was added, the GRU model showed an improvement of over 1% in the MAPE and a noticeable reduction in the SMAPE. Similarly, the LSTM model achieves lower absolute and relative errors in the trend-informed scenario. Although the difference in MAE between the trend and non-trend versions of LSTM was minimal (approximately 0.1–0.2 min), the trend-based variant exhibited slightly better precision.

In contrast, the transformer model performed worse than those of the other architectures. It produced a high MAE of 3.64 minutes and MAPE of 24.60%, with R^2 falling below 0.87. Because of these results, the transformer model was excluded from further analysis.

TABLE 5. Weighted average model metrics on route 12A.

Model	Trend	MAE	RMSE	R^2	MAPE (%)	SMAPE (%)
GRU	No	2.06	4.75	0.9472	9.71	10.40
GRU	Yes	1.93	4.77	0.9470	8.70	9.41
LSTM	No	2.00	4.75	0.9473	9.28	10.20
LSTM	Yes	2.01	4.77	0.9471	9.41	11.12
Transformer	No	3.64	7.27	0.8718	24.60	21.11

The Transformer model demonstrated acceptable performance on larger datasets, but suffered substantial accuracy loss in the 0–1000 sample range (e.g., $R^2 = -1.28$, MAPE = 33.4%). Although slight improvements were observed in the 1000–3000 range, the desired accuracy levels were achieved only when the number of samples exceeded 5000.

Despite experimenting with activation functions (ReLU, GELU, and MELU) and adjusting the encoder depth, attention heads, and learning rates, as recommended in the literature, no significant gains were achieved in the low-sample scenarios. This confirms that Transformer architectures are more suitable for large datasets and tend to overfit under limited data conditions [44], [45], [46].

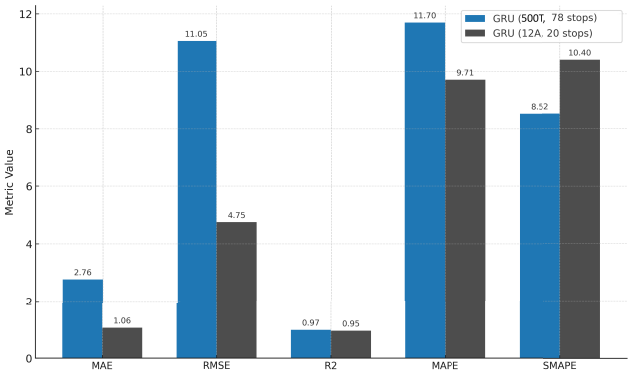


FIGURE 4. Comparison of GRU performance on routes 12A and 500T.

The comparison in Figure 4 shows that the GRU model performs better on Route 12A (20 stops) than on Route 500T (78 stops). For Route 12A, MAE was 1.06 minutes, while for 500T, it rose to 2.76 minutes. Similarly, the RMSE increased from 4.75 to 11.05, indicating higher sensitivity to large errors. MAPE increased from 9.70% to 11.69%. Although the SMAPE behaved inconsistently, the differences were not statistically significant.

The lower performance of Route 500T is attributed to its longer travel time, variable traffic conditions, and greater inter-stop distances. While the GRU may suffice for shorter and more regular routes, the LSTM is better suited for longer and more complex routes. This aligns with the findings in the literature, which highlight LSTM's ability to

capture long-term dependencies and improve the prediction accuracy [48].

Table 6 presents the model performance across the different data size groups. The trend feature notably improved the accuracy for the 0–1K sample group, reducing the MAE from 6.48 to 4.64 minutes and SMAPE from 27.83% to 16.20%. However, in the 1K–2K and 2K–3K ranges, the trend-based model exhibited higher errors, suggesting inconsistent benefits in the medium-sized groups. In the 3K+ group, both models achieved low error rates, although the trend model performed slightly worse in terms of MAE (1.83 vs. 1.48 minutes).

Based on these findings, a hybrid approach was adopted: the trend feature was included only for groups with fewer than 1000 samples and was set to zero otherwise. This strategy improves general model accuracy by applying the trend feature only when it is beneficial.

TABLE 6. Average metrics by sample size group (LSTM only).

Group	Trend	MAE (min)	RMSE (min)	R^2	MAPE (%)	SMAPE (%)
0–1K	No	6.48	14.09	0.850	23.43	27.82
0–1K	Yes	4.64	11.59	0.854	18.06	16.20
1K–2K	No	2.40	10.80	0.946	14.46	10.95
1K–2K	Yes	4.06	14.69	0.942	16.99	11.61
2K–3K	No	2.32	12.96	0.915	10.21	8.03
2K–3K	Yes	3.24	13.19	0.912	13.42	10.45
3K+	No	1.47	9.34	0.966	7.328	5.96
3K+	Yes	1.83	9.36	0.966	7.782	6.45

1) THRESHOLD JUSTIFICATION:

We conducted sensitivity tests with thresholds of $N \in \{500, 1000, 2000\}$. While $N = 500$ led to unstable performance and $N = 2000$ diluted the benefits for sparse groups, $N = 1000$ consistently achieved the best trade-off between mitigating overfitting in small-sample conditions and preserving accuracy in larger groups. Therefore, $N = 1000$ was adopted as a practical and robust threshold.

Based on these findings, a hybrid approach was adopted: the trend feature was included only for groups with fewer than 1000 samples and was set to zero otherwise. This strategy improves general model accuracy by applying the trend feature only when it is beneficial.

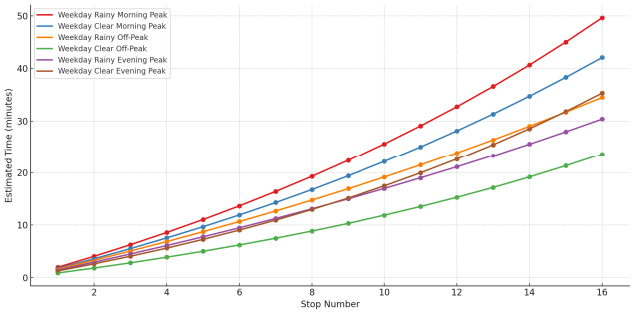


FIGURE 5. Predicted arrival times on route 12A-outbound under different conditions.

Figure 5 illustrates the stop-to-stop travel-time predictions for Route 12A on weekdays for different periods and weather conditions. During the morning peak, travel time increases from approximately 1.8 minutes at Stop 1 to 42.1 minutes at Stop 16 under clear conditions and up to 49.7 minutes in rainy weather. In off-peak hours, the travel time ranges from 0.9 to 23.6 minutes in clear weather and from 1.6 to 34.5 minutes under rain. Evening estimates exhibit a consistent upward trend, with predictions ranging from 1.3 to 35.3 minutes in clear conditions and up to 30.4 minutes when it rains. The marker points at each stop highlight the incremental build-up of travel time, emphasizing the model's ability to capture both weather-related delays and time-of-day traffic variations.

TABLE 7. Overall weighted model performance across 500 routes.

Model	Trend	MAE (min)	RMSE (min)	R^2	MAPE (%)	SMAPE (%)
GRU	No	3.373	8.601	0.9267	16.62	16.15
LSTM	No	3.129	8.277	0.9271	15.27	14.91
LSTM	**Yes**	**2.975**	**8.129**	**0.9272**	**14.79**	**14.27**

Note: Final selected model (LSTM with trend) is shown in bold.

The weighted performance evaluation across 500 bus routes (Table 7) revealed that the LSTM model consistently outperformed the GRU. While GRU yielded an MAE of 3.373 min, the trend-based LSTM achieved a lower MAE of 2.975 min. The RMSE and MAPE also improved with LSTM, particularly in the trend-informed version. Although R^2 remained close across all models (∼0.927), the MAE and SMAPE values confirmed the superior generalization of LSTM, particularly for complex and high-variance routes.

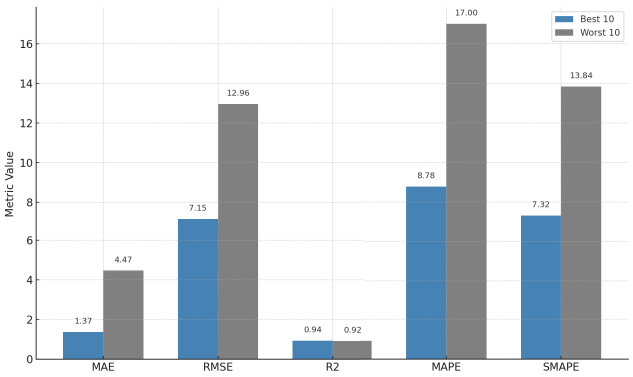


FIGURE 6. LSTM model evaluation for best and worst 10 routes.

Figure 6 shows that even in the best-performing routes (lowest MAE), the RMSE values can be relatively high, indicating occasional large deviations. Compared to the worst 10 routes, the top 10 routes show much lower MAE, MAPE, and SMAPE values, although some prediction volatility remains. This suggests that although the model performs well overall, certain routes or trip scenarios still pose challenges that warrant further investigation.

G. CONDITION-BASED ANALYSIS (TIME, WEATHER, AND DAY TYPE)

To evaluate how contextual variables affect prediction accuracy, the LSTM model (with and without trends) was tested

under different temporal, meteorological, and calendar-based conditions. These condition-specific analyses revealed the sensitivity of the model to real-world variations in urban transit systems.

The following section presents the general model evaluation results based on the outcomes obtained using the hybrid LSTM approach.

1) PERFORMANCE BY TIME BLOCK

Table 8 shows the model accuracy across different time-of-day segments, including the morning peak, evening peak, off-peak (normal), and night hours.

TABLE 8. Model performance by time block.

Time Block	MAE	RMSE	R^2	MAPE (%)	SMAPE (%)
Morning Peak	3.83	15.28	0.869	11.39	10.18
Evening Peak	2.98	11.75	0.949	10.12	8.45
Normal Hours	2.67	7.15	0.983	9.47	7.50
Night Hours	1.77	2.77	0.934	7.61	6.52

The model performed the weakest during the morning peak, when both MAE and RMSE reached their highest levels and R^2 dropped significantly. This is likely due to high traffic congestion and irregularities in travel patterns during these hours. In contrast, the model performed best during night hours, when stable traffic conditions led to lower errors. The evening peak results showed moderate accuracy, which was supported by a larger sample size. These results confirm that time-of-day variations significantly affect the predictive performance of arrival time models.

2) PERFORMANCE BY WEATHER CONDITION

Table 9 presents the model's predictive accuracy under different weather scenarios such as clear, overcast, partly cloudy, and rainy conditions.

TABLE 9. Model performance by weather condition.

Weather Condition	MAE	RMSE	R^2	MAPE (%)	SMAPE (%)
Clear	2.835	9.853	0.946	10.04	8.28
Overcast	3.118	8.748	0.934	8.37	7.30
Partly Cloudy	2.970	11.074	0.931	9.99	8.33
Rain	3.636	7.395	0.896	9.98	8.82

Rainy conditions had the most negative impact on the model performance. Although the RMSE was not the highest in this group, the MAE and SMAPE values were significantly higher. The decrease in R^2 to 0.896 confirms that weather-induced variability affects the model predictability. Interestingly, the overcast category yielded a lower MAPE despite a higher MAE, possibly owing to smaller deviations in the relative error for medium-length trips. Overall, the model proved to be sensitive to weather changes, with rain causing the greatest degradation in performance.

3) PERFORMANCE BY DAY TYPE

Table 10 compares the model performance across weekdays and weekends (Saturday and Sunday). While weekdays have more data, weekends offer insight into the model's generalizability under less congested and more regular service patterns.

TABLE 10. Model performance by day type.

Day Type	MAE	RMSE	R^2	MAPE (%)	SMAPE (%)
Weekday	3.003	11.381	0.933	9.85	8.22
Saturday	2.775	8.358	0.915	10.13	8.59
Sunday	2.780	7.414	0.948	11.53	9.59

Although weekdays offered larger training samples, the model exhibited slightly lower MAE and RMSE values on the weekends. Notably, R^2 was highest on Sunday (0.948), suggesting that the model could learn patterns more accurately when service frequency and traffic variation were lower. However, relative error metrics, such as MAPE and SMAPE, were higher on weekends, which may be due to fewer data points and larger percentage swings on shorter trips. These results suggest that, although model training is optimized for weekdays, performance on weekends remains robust, albeit with more volatility.

V. FINDINGS AND DISCUSSION

A. MODEL COMPARISON SUMMARY

The comparative performances of different deep learning architectures are thoroughly analyzed in Section IV. This section summarizes the key findings and outlines the rationale behind the model selection.

Three architectures—LSTM, GRU, and Transformer—were tested on real-time data from 500 bus routes in Istanbul using identical input—output configurations. The results show that the model performance depends not only on the architecture but also on the data volume, contextual conditions, and model structure.

The Transformer model produced acceptable results on large datasets, but suffered significant accuracy loss in low-sample groups (e.g., $R^2 < 0$, MAPE $> 30\%$). These findings align with prior studies that reported overfitting tendencies of transformer-based models in sparse data settings [39], [45], [46]. Adjustments to the activation functions (ReLU, GELU, and MELU), encoder depth, and attention heads do not yield stability improvements under low-data conditions [44].

The GRU model showed some advantages for short and regular routes, owing to its lower parameter count. However, they struggled to generalize to longer and more complex scenarios. This outcome supports existing research suggesting that GRU is less capable than LSTM in modeling long-term dependencies in sequential data [14], [15], [42].

By contrast, the LSTM model delivered the most consistent performance across all sample sizes. Notably, in groups with fewer than 1000 samples, the trend-informed hybrid variant reduced the average error metrics by up to 28%. This demonstrates LSTM's flexibility in adapting to both sparse and dense transit data.

B. CONDITION-BASED PERFORMANCE

Model performance was found to vary significantly based on time of day, weather conditions, and day type.

During morning peak hours, increased traffic congestion and unpredictability led to the highest observed error rates (MAE $>$ 3.8 min, $R^2 < 0.87$), confirming previous research on degraded performance during high-variance time periods [47]. In contrast, nighttime predictions were more accurate (MAE $<$ 2 min), benefiting from the stable travel patterns. These findings confirmed the time sensitivity of the bus arrival prediction model.

Weather also had a measurable impact on the model accuracy. In particular, the rainy conditions produced the largest prediction errors (MAPE $=$ 33.4%). These results are consistent with prior studies showing that severe weather introduces significant variance in travel time and degrades the model performance [20], [46].

In terms of the day type, weekend predictions were often more accurate, although the weekday datasets had a higher volume. The consistency of travel patterns on weekends appears to enhance the model reliability. This supports earlier findings that regularity, not just data volume, is the key to accurate forecasting [19], [21]. These insights also confirm that models trained without condition-based segmentation may yield lower generalization accuracy, particularly during volatile periods, such as rush hours or adverse weather conditions. Therefore, integrating features such as time, weather, and day type directly into the model or segmenting data beforehand is essential for improving accuracy.

C. HYBRID MODEL EVALUATION

The hybrid model structure was developed based on observations that LSTM's response to trend information varies with sample size (see Section V-C). Specifically, the trend-enhanced version significantly outperformed the baseline in groups with fewer than 1000 samples, whereas performance gains were limited or negative in higher data ranges.

Accordingly, during training, the trend feature was included in the input vector only for records where $1 \leq N < 1000$. For all the other cases, the trend input was set to zero. This approach can be described by the following logic.

$$X_i' = [X_i,\ T_i], \quad \text{where } T_i = \begin{cases} \text{value}, & \text{if } 0 \leq N_{ri} < 1000 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

This selective trend usage resulted in a hybrid model that matched the performance of the non-trend LSTM version (MAE $\approx$ 2.30 min, $R^2 = 0.932$) and slightly improved relative error metrics (SMAPE: 9.60% vs. 9.74%).

Additional tests also confirmed that adding trend features to the GRU architecture improves accuracy in low-sample groups [45]. In this study, the trend input was similarly restricted to the 1–1000 sample range, yielding a balanced model with minimal added complexity. These results demonstrate that selectively integrating trend features can improve the performance without overcomplicating the model, particularly in data-sensitive forecasting applications.

To statistically validate the performance differences among models, we conducted pairwise Diebold-Mariano tests using squared forecast errors. The results show that the LSTM model significantly outperforms both GRU ($p = 0.0132$) and Transformer ($p < 0.0001$). These findings confirm that the observed improvements in LSTM's prediction accuracy are statistically significant and not due to random variation.

To strengthen our performance benchmark, we extended the comparative evaluation to include not only ensemble-based models (Random Forest, XGBoost) and statistical forecasting methods (Prophet), but also the classical ARIMA model. All models were trained and tested on the same dataset using identical input features. The results, summarized in Table 11, show that the proposed LSTM model achieves the lowest error rates across all metrics (e.g., MAE, RMSE, and MAPE), substantially outperforming the classical methods.

Furthermore, we applied pairwise Diebold-Mariano tests to assess the statistical significance of the prediction differences. The results confirmed that the LSTM model significantly outperforms all competing models, with p-values < 0.001 in all comparisons. Notably, the LSTM vs ARIMA test yielded a DM statistic of –191.29, highlighting the inability of the ARIMA model to cope with the contextual variability and nonlinear dynamics inherent in our dataset.

These findings reinforce the suitability of deep learning approaches–especially context-aware architectures like LSTM—in complex urban transit forecasting scenarios, where temporal dependencies and exogenous factors (e.g., weather, day type, time-of-day) critically affect travel times.

TABLE 11. Performance metrics of classical models vs proposed LSTM model.

Model	MAE	RMSE	R^2	MAPE (%)
Random Forest	11.45	28.34	0.92	48.73
XGBoost	10.71	26.72	0.93	47.20
Prophet	61.60	102.06	-0.00	299.11
ARIMA	147.49	156.89	-1.36	693.44
LSTM (Proposed)	2.97	8.13	0.93	14.79

D. COMPUTATIONAL COMPLEXITY ANALYSIS

To assess the computational efficiency of the proposed models, we analyzed both theoretical time complexity and empirical execution times. LSTM and GRU exhibit linear time complexity $\mathcal{O}(T \cdot H^2)$, whereas the Transformer model has quadratic time complexity $\mathcal{O}(T^2 \cdot d)$ due to the attention mechanism.

We conducted benchmark tests under identical hardware and data conditions. As shown in Table 12, GRU achieved the fastest training and inference times, followed by LSTM. The Transformer, while powerful in modeling, required significantly more computational resources.All training and inference durations were measured using 15 epochs on a

single route dataset (12A line), using identical input size and batch settings across models. This setup allows a fair comparison of computational load per model in a controlled scenario.

TABLE 12. Training and inference time of models under identical settings.

Model	Train Time (s)	Inference Time (s)	Complexity
LSTM	174.50	3.93	$\mathcal{O}(T \cdot H^2)$
GRU	107.58	1.88	$\mathcal{O}(T \cdot H^2)$
Transformer	579.40	5.13	$\mathcal{O}(T^2 \cdot d)$

E. EXCLUSION OF GRU AND TRANSFORMER ARCHITECTURES

Both GRU and Transformer architectures were tested in the early phases of this study. However, the experimental findings revealed that these models have major limitations in terms of accuracy and generalizability under various data regimes.

Although GRU offers faster training and better performance on short, high-density routes owing to its smaller parameter set, it fails to capture long-term dependencies. Weighted average comparisons showed that the GRU underperformed the LSTM across all metrics. For example, GRU produced MAPE and SMAPE values of 13.97% and 11.50%, respectively, whereas those of the LSTM model were 10.83% and 9.08%, respectively. These differences suggest that the GRU is more prone to instability in high-variance scenarios and less capable of modeling sequential temporal relationships [42].

While the Transformer model achieved acceptable results on high-volume datasets, its performance drastically declined in small-sample groups ($R^2 = -1.28$, MAPE $= 33.43\%$). The model exhibited signs of overfitting and variance under sparse-data conditions. Despite testing various configurations including MELU, GELU, ReLU, encoder depth, and attention head count, no improvements were observed. Even when the trend input was included, the prediction accuracy did not improve in low-sample settings. These results confirm that transformer models require large datasets to generalize well, and often struggle when data are limited, as also reported in prior studies [44], [45], [46].

Consequently, the LSTM model was selected for further development owing to its stable performance across all scenarios.

F. LITERATURE COMPARISON AND CONTRIBUTION

Most existing studies on travel time prediction rely on a single data source (typically GPS or historical trip durations) [3], [4] or evaluate accuracy using average metrics without segmenting the dataset [22], [40]. In contrast, this study integrates multiple data sources, including GTFS, hourly weather, and contextual factors, and performs detailed condition-based evaluations.

A novel contribution of this study is the hybrid input strategy that uses trend information only for low-sample groups. It captures the effect of data volume on model behavior, an aspect that is often overlooked in the literature. Consequently, the proposed method offers both theoretical value (in adaptive modeling) and practical benefits (in improving the real-time transit prediction accuracy). These distinctions make this approach a meaningful contribution to the research on and operational forecasting of smart transportation systems.

VI. CONCLUSION

This paper presents a comprehensive evaluation of bus arrival time prediction using three deep learning architectures: LSTM, GRU, and Transformer. The models were trained and tested on six months of real-time bus data collected from 500 routes in Istanbul, enriched with contextual information, such as time block, weather conditions, and day type. The experimental setup involved both route-level and generalized modeling, enabling robust analysis under diverse real-world scenarios.

The results demonstrated that the LSTM model consistently outperformed both the GRU and Transformer under all conditions. Although the transformer demonstrated acceptable accuracy for large datasets, it exhibited a severe performance degradation in low-sample scenarios. Although efficient in training, GRU fails to effectively model sequential dependencies on long or complex routes.

One of the key findings of this study is that model accuracy is influenced not only by architecture but also by data segmentation. Condition-based analyses revealed significantly higher error rates during the morning rush hours, rainy days, and routes with limited data. To address these challenges, a hybrid LSTM model was proposed, in which the trend feature was included only for groups with 1–1000 samples. This adaptive approach improves both absolute and relative performance metrics, particularly in low-data contexts.

The results confirm that successful forecasting in public transportation systems depends not only on model selection, but also on appropriate preprocessing, feature structuring, and contextual awareness. The hybrid approach presented in this study offers a flexible and scalable solution, providing more reliable travel-time predictions for passengers and transit authorities. Although the Transformer model showed limited effectiveness under sparse-data conditions, its potential for large-scale learning remains promising. Due to the dataset's structure and scope, transfer learning was not pursued in this study; however, future work will explore pretraining on broader datasets and applying domain-adaptive fine-tuning techniques to improve generalization in high-volume scenarios.

Future research will focus on extending this framework using Graph Neural Networks (GNNs), attention-based encoder—decoder architectures, and multi-condition time-series models. Additionally, testing the generalizability of the proposed hybrid approach to other cities and transport modes will help to define a more universal modeling strategy for smart urban mobility systems.

Despite the promising results achieved by the proposed hybrid LSTM model, several limitations remain, which can guide future research. First, the model was developed and validated using data solely from the Istanbul public transportation system; hence, its generalizability to other cities with different transit dynamics and infrastructure has not yet been confirmed. Second, the current implementation was evaluated in an offline setting, and its integration into a real-time prediction environment requires further research on latency reduction and system scalability. Third, although Transformer models performed poorly under low-sample scenarios, they may still offer advantages in high-volume contexts if combined with data augmentation or advanced encoder structures. Additionally, the model currently utilizes a limited set of input features, mainly GPS, weather, and GTFS data, and incorporating further contextual sources, such as live traffic congestion indices, fare card activity, or special events, could enhance the prediction robustness. Finally, the predictions were deterministic without accompanying uncertainty estimates. Future versions can benefit from probabilistic forecasting techniques that offer confidence intervals and point estimates, thereby improving their practical utility in real-world transit applications. From a practical standpoint, deploying the proposed deep learning framework in real-time transit systems poses several challenges. These include maintaining low-latency predictions, ensuring continuous access to real-time GPS and weather data, and handling missing or delayed inputs. Moreover, on-device (edge) inference or cloud-based pipelines may introduce constraints on computational resources and scalability. As future work, we plan to explore model compression techniques, such as pruning or quantization, and evaluate real-time performance in production-like environments.

The source code and subset of the cleaned dataset will be made publicly available upon acceptance of the manuscript, to facilitate reproducibility and further research.

1) REPRODUCIBILITY

To support reproducibility, we provide a minimal inference notebook, a small anonymized sample of the dataset, and trained model weights in the Supplementary Materials. The notebook reproduces the core results (e.g., Table 4) with a single run, enabling independent validation of the proposed framework.

REFERENCES

[1] A. Taparia and M. Brady, "Bus journey and arrival time prediction based on archived AVL/GPS data using machine learning," in *Proc. 7th Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Heraklion, Greece, Jun. 2021, pp. 1–6, doi: 10.1109/MT-ITS49943.2021.9529328.

[2] N. Rashvand, S. S. Hosseini, M. Azarbayjani, and H. Tabkhi, "Real-time bus arrival prediction: A deep learning approach for enhanced urban mobility," 2023, *arXiv:2303.15495*.

[3] C. Bai, Z.-R. Peng, Q.-C. Lu, and J. Sun, "Dynamic bus travel time prediction models on road with multiple bus routes," *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–9, Jul. 2015, doi: 10.1155/2015/432389.

[4] S. Sowlati, R. A. Abbaspour, and A. Chehreghan, "An interpretable detection of transportation mode considering GPS, spatial, and contextual data based on ensemble machine learning," in *Artificial Intelligence for Cyber-Physical Systems*. Boca Raton, FL, USA: CRC Press, 2023, pp. 207–240, doi: 10.1201/9781003324140-10.

[5] A. Achar, A. Natarajan, R. Regikumar, and B. A. Kumar, "Predicting public transit arrival: A nonlinear approach," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103875, doi: 10.1016/j.trc.2022.103875.

[6] A. Dairi, F. Harrou, and Y. Sun, "Auto-configured explainable graph neural networks for multi-site pollution prediction," *Appl. Soft Comput.*, vol. 181, Sep. 2025, Art. no. 113496, doi: 10.1016/j.asoc.2025.113496.

[7] D. Abdelkader, H. Fouzi, K. Belkacem, and S. Ying, "Graph neural networks-based spatiotemporal prediction of photovoltaic power: A comparative study," *Neural Comput. Appl.*, vol. 37, no. 6, pp. 4769–4795, Feb. 2025, doi: 10.1007/s00521-024-10751-9.

[8] H. Liu, H. Xu, Y. Yan, Z. Cai, T. Sun, and W. Li, "Bus arrival time prediction based on LSTM and spatial–temporal feature vector," *IEEE Access*, vol. 8, pp. 11917–11929, 2020, doi: 10.1109/ACCESS.2020.2965094.

[9] Y. Ou, "AI for real-time bus travel time prediction in traffic congestion management," in *Humanity Driven AI: Productivity, Well-Being, Sustainability and Partnership*. Cham, Switzerland: Springer, 2022, pp. 63–84, doi: 10.1007/978-3-030-72188-6_4.

[10] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, Feb. 2019, doi: 10.1016/j.trc.2018.12.004.

[11] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis and future directions," 2020, *arXiv:2004.08555*.

[12] Z. Li, "DA-RNN-based bus arrival time prediction model," *Int. J. Intell. Transp. Syst. Res.*, vol. 22, no. 3, pp. 660–674, Dec. 2024, doi: 10.1007/s13177-024-00422-3.

[13] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015, doi: 10.1016/j.trc.2015.03.014.

[14] H. Ding, Z. Li, and N. Su, "Traffic prediction based on the GRU neural network," *Appl. Comput. Eng.*, vol. 8, no. 1, pp. 287–291, Aug. 2023, doi: 10.54254/2755-2721/8/20230168.

[15] M. M. Karim, R. Qin, and Y. Wang, "Fusion-GRU: A deep learning model for future bounding box prediction of traffic agents in risky driving videos," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2678, no. 9, pp. 699–709, Sep. 2024, doi: 10.1177/03611981241230540.

[16] B. Zhang, D. Zhou, and J. Sun, "Bus arrival time prediction model based on bidirectional long short-term memory network," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 23, no. 2, pp. 148–160, 2023, doi: 10.16097/j.cnki.1009-6744.2023.02.016.

[17] J. Jalaney and R. S. Ganesh, "Multiple extreme learning machines based arrival time prediction for public bus transport," *Intell. Autom. Soft Comput.*, vol. 36, no. 3, pp. 2819–2834, 2023, doi: 10.32604/iasc.2023.034844.

[18] N. Rashvand, S. S. Hosseini, M. Azarbayjani, and H. Tabkhi, "Real-time bus departure prediction using neural networks for smart IoT public bus transit," 2025, *arXiv:2501.10514*.

[19] Z. Li, P. Wolf, and M. Wang, "ArrivalNet: Predicting city-wide bus/tram arrival time with two-dimensional temporal variation modeling," 2024, *arXiv:2410.14742*.

[20] Z. Zhang, Z. Huang, Z. Hu, X. Zhao, W. Wang, Z. Liu, J. Zhang, S. J. Qin, and H. Zhao, "MLPST: MLP is all you need for spatio-temporal prediction," 2023, *arXiv:2309.13363*.

[21] S. Jeong, C. Oh, and J. Jeong, "BAT-transformer: Prediction of bus arrival time with transformer encoder for smart public transportation system," *Appl. Sci.*, vol. 14, no. 20, p. 9488, Oct. 2024, doi: 10.3390/app14209488.

[22] H. Nguyen, L.-M. Kieu, T. Wen, and C. Cai, "Deep learning methods in transportation domain: A review," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 998–1004, Nov. 2018, doi: 10.1049/iet-its.2018.0064.

[23] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020, doi: 10.1109/TITS.2019.2929020.

[24] B. Zhang, L. Tang, D. Zhou, K. Liu, and Y. Xue, "Bus arrival time prediction based on the optimized long short-term memory neural network model with the improved whale algorithm," *J. Adv. Transp.*, vol. 2024, no. 1, pp. 1–15, Jan. 2024, doi: 10.1155/2024/6997338.

[25] A. Agafonov and A. Yumaganov, "Bus arrival time prediction with LSTM neural network," in *Proc. Adv. Neural Netw. (ISNN)*, 2019, pp. 11–18, doi: 10.1007/978-3-030-22796-8_2.

[26] Y. Lai, S. Easa, D. Sun, and Y. Wei, "Bus arrival time prediction using wavelet neural network trained by improved particle swarm optimization," *J. Adv. Transp.*, vol. 2020, pp. 1–9, Jan. 2020, doi: 10.1155/2020/7672847.

[27] Y. P. Huang, C. Chen, Z. C. Su, T. S. Chen, A. Sumalee, T. L. Pan, and R. X. Zhong, "Bus arrival time prediction and reliability analysis: An experimental comparison of functional data analysis and Bayesian support vector regression," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107663, doi: 10.1016/j.asoc.2021.107663.

[28] M. Hosseinzadeh, E. Azhir, O. H. Ahmed, M. Y. Ghafour, S. H. Ahmed, A. M. Rahmani, and B. Vo, "Data cleansing mechanisms and approaches for big data analytics: A systematic study," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 1, pp. 99–111, Jan. 2023, doi: 10.1007/s12652-021-03590-2.

[29] Q. Guo, Z. He, and Z. Wang, "Prediction of monthly average and extreme atmospheric temperatures in Zhengzhou based on artificial neural network and deep learning models," *Frontiers Forests Global Change*, vol. 6, Dec. 2023, Art. no. 1249300, doi: 10.3389/ffgc.2023.1249300.

[30] Q. Guo, Z. He, and Z. Wang, "Monthly climate prediction using deep convolutional neural network and long short-term memory," *Sci. Rep.*, vol. 14, no. 1, 2024, Art. no. 17748, doi: 10.1038/s41598-024-68906-6.

[31] Z. He, Q. Guo, Z. Wang, and X. Li, "A hybrid wavelet-based deep learning model for accurate prediction of daily surface PM2.5 concentrations in Guangzhou city," *Toxics*, vol. 13, no. 4, p. 254, Mar. 2025, doi: 10.3390/toxics13040254.

[32] *General Transit Feed Specification (GTFS)*. Accessed: Mar. 30, 2025. [Online]. Available: https://gtfs.org

[33] Istanbul Electric Tram and Tunnel Company (IETT). *GTFS Data Feed*. Accessed: Mar. 30, 2025. [Online]. Available: https://data.iett.gov.tr

[34] E. Zimányi, M. Sakr, and A. Lesuisse, "MobilityDB: A mobility database based on PostgreSQL and PostGIS," *ACM Trans. Database Syst.*, vol. 45, no. 4, pp. 1–42, Dec. 2020, doi: 10.1145/3406534.

[35] A. Monzon, R. Julio, and A. Garcia-Martinez, "Hybrid methodology for improving response rates and data quality in mobility surveys," *Travel Behav. Soc.*, vol. 20, pp. 155–164, Jul. 2020, doi: 10.1016/j.tbs.2020.03.012.

[36] J. Pang, J. Huang, Y. Du, H. Yu, Q. Huang, and B. Yin, "Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3283–3293, Sep. 2019, doi: 10.1109/TITS.2018.2873747.

[37] W. Ding and Y. Cao, "A data cleaning method on massive spatio-temporal data," in *Advances in Services Computing (APSCC 2016)* (Lecture Notes in Computer Science), vol. 10065. Cham, Switzerland: Springer, 2016, pp. 173–182, doi: 10.1007/978-3-319-49178-3_13.

[38] M. Cavojsky, "Identifying bus lines and outliers in bus routes of Rio De Janeiro," *Proc. Comput. Sci.*, vol. 225, pp. 3764–3773, Jan. 2023, doi: 10.1016/j.procs.2023.10.372.

[39] M. Kim and S. Lee, "Transformer-based bus arrival time prediction using real-time AVL data," *Appl. Intell.*, vol. 52, pp. 4876–4892, Jan. 2022.

[40] N. Bhutani, S. Pachal, and A. Achar, "Public transit arrival prediction: A Seq2Seq RNN approach," 2022, *arXiv:2210.01655*.

[41] C. Li, L. Bai, W. Liu, L. Yao, and S. Travis Waller, "Urban mobility analytics: A deep spatial–temporal product neural network for traveler attributes inference," *Transp. Res. C, Emerg. Technol.*, vol. 124, Mar. 2021, Art. no. 102921, doi: 10.1016/j.trc.2020.102921.

[42] J.-U.-R. Chughtai, I. U. Haq, and M. Muneeb, "An attention-based recurrent learning model for short-term travel time prediction," *PLoS ONE*, vol. 17, no. 12, Dec. 2022, Art. no. e0278064, doi: 10.1371/journal.pone.0278064.

[43] Y. Wei and H. Liu, "Convolutional long-short term memory network with multi-head attention mechanism for traffic flow prediction," *Sensors*, vol. 22, no. 20, p. 7994, Oct. 2022, doi: 10.3390/s22207994.

[44] M. Shoman, A. Aboah, and Y. Adu-Gyamfi, "Deep learning framework for predicting bus delays on multiple routes using heterogenous datasets," *J. Big Data Anal. Transp.*, vol. 2, no. 3, pp. 275–290, Dec. 2020, doi: 10.1007/s42421-020-00031-y.

[45] P. A. B. Andrade, M. Santos, J. E. Sierra-García, and J. P. P. Piedra, "Comparison of LSTM, GRU and transformer neural network architecture for prediction of wind turbine variables," in *Proc. 18th Int. Conf. Soft Comput. Models Ind. Environ. Appl. (SOCO)*, Salamanca, Spain, Sep. 2023, pp. 334–343, doi: 10.1007/978-3-031-42536-3_32.

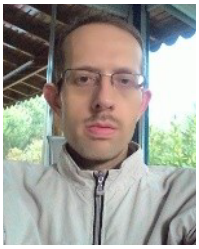
[46] M. Zhou, D. Wang, Q. Li, Y. Yue, W. Tu, and R. Cao, "Impacts of weather on public transport ridership: Results from mining data from different sources," *Transp. Res. C, Emerg. Technol.*, vol. 75, pp. 17–29, Feb. 2017, doi: 10.1016/j.trc.2016.12.001.

[47] Y. Zhang, X. Wang, J. Xie, and Y. Bai, "Comparative analysis of deep-learning-based models for hourly bus passenger flow forecasting," *Transportation*, vol. 51, no. 5, pp. 1759–1784, Oct. 2024, doi: 10.1007/s11116-023-10385-1.

[48] Y. Guo and L. Yang, "Reliable estimation of urban link travel time using multi-sensor data fusion," *Information*, vol. 11, no. 5, p. 267, May 2020, doi: 10.3390/info11050267.



OSMAN KAYA received the B.Sc. and M.Sc. degrees in computer engineering. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Yıldız Technical University, Istanbul, Türkiye. He actively engages in projects aimed at enhancing the reliability and efficiency of public transportation systems using advanced data analytics. His research interests include intelligent transportation systems, machine learning, spatio-temporal data analysis, and deep learning applications in urban transit forecasting.



MUSTAFA UTKU KALAY received the B.S. degree in electrical engineering from Istanbul Technical University, in 1997, the M.S. degree in computer science from the University of Southern California, Los Angeles, in 2000, and the Ph.D. degree in computer engineering from Yıldız Technical University. He is currently an Assistant Professor and a Lecturer. His main research interests include spatial and spatio-temporal data management, adaptive index, and organizations.