Babylonian Journal of Artificial Intelligence
Vol. 2024, **pp**. 73–72
DOI: https://doi.org/10.58496/BJAI/2024/010 ; ISSN: 3006-5437
https://mesopotamian.press/journals/index.php/BJAI

Research Article

Improving Bus Passenger Flow Prediction Using Bi-LSTM Fusion Model and SMO Algorithm

Karthika Balasubramani[1, *], Uma Maheswari Natarajan [2]

[1] *Department of Information Technology, PSNA College of Engineering and Technology, Kothandaraman Nagar, Tamil Nadu 624622, India.*

[2] *Department of Computer Science and Engineering, PSNA College of Engineering and technology, Kothandaraman Nagar, Tamil Nadu 624622, India.*

ARTICLE INFO

Article History

Received 01 May 2024
Revised 02 Jun 2024
Accepted 15 Jun 2024
Published 05 Jul 2024

Keywords

Data analytics

Bi-LSTM

Traffic management

Bus passenger

Urban mobility

ABSTRACT

Bus passenger flow prediction integrates data analytics and modelling techniques to forecast the number of passengers using bus services, incorporating historical usage patterns, demographics, weather, and events for optimal scheduling and resource allocation. The Bi-LSTM fusion model enhances accuracy by processing past and future features simultaneously, leveraging bidirectional LSTM layers and an attention mechanism to capture temporal dependencies. This approach not only refines insights crucial for urban mobility challenges like traffic management and demand forecasting but also improves route planning and service efficiency. The SMO algorithm initializes with a diverse spider monkey population exploring solution spaces. Through local and global leader phases, it iteratively updates positions based on fitness and probabilistic selections, maintaining a balance between exploration and exploitation. Perturbation-based updates in the local leader phase ensure adaptability, preventing premature convergence, while the global leader phase guides towards better solutions, enhancing efficiency in complex optimization tasks and promoting dynamic adaptation. In Dataset 1, the proposed model achieved a training time of 137 seconds, slightly longer than HA (115s), SARIMA (112s), GRU (123s), and ST-ResNet (113s). It demonstrated superior accuracy at 89%, surpassing HA (66%), SARIMA (68%), GRU (63%), DeepST (78%), and ST-ResNet (84%). In Dataset 2, the model exhibited the lowest RMSE, MAE, and MAPE%, indicating superior predictive accuracy over SVR, CNN, GCN, LSTM, and CONV LSTM models. These findings validate the proposed model's effectiveness in enhancing predictive capabilities for transit forecasting, underscoring its potential to optimize urban mobility and transportation management strategies significantly.

1. INTRODUCTION

Predicting passenger flow at bus stations is a crucial task for improving capacity planning, enhancing security, enhancing the passenger traveling experience, and promoting infrastructure development. Accurate predictions can help transit authorities allocate resources efficiently, prevent overcrowding, and ensure a smooth operation of services [1]. However, many existing methods have failed to accurately predict passenger flow due to several challenges, including incorrect or incomplete data, and external factors such as weather conditions, traffic information, holidays, and city events. These external factors introduce a high degree of variability and complexity into the prediction models, making accurate forecasting difficult [2]. Additionally, these traditional methods are inefficient because they do not employ advanced deep learning techniques, which are better suited for capturing complex patterns in data. One of the key issues with existing methods is their inability to handle the performance degradation of transportation systems during traffic flow calculations [3]. Traditional models often lack the robustness needed to adapt to the dynamic nature of passenger flow, leading to less reliable predictions. This study aims to address these limitations by predicting the future inbound passenger flow at bus stations using a deep learning approach [4]. Specifically, it proposes the use of a Bidirectional Long Short-Term Memory (Bi-LSTM) network model to capture data features in both forward and backward directions. Bi-LSTM networks are

*Corresponding author. Email: bkarthikapsnacet@gmail.com

particularly well-suited for time series prediction tasks due to their ability to retain information from both past and future data points, providing a more comprehensive understanding of the temporal dependencies in the data [5].

To further enhance the predictive power of the Bi-LSTM model, this study incorporates an attention mechanism. The attention mechanism enables the model to focus on the most important features of the data, thereby improving prediction results [6]. This is particularly beneficial for handling the nonlinear and periodic nature of bus-passenger-flow data, where certain time periods or external conditions might have a more significant impact on passenger flow than others. By selectively concentrating on these critical aspects, the attention mechanism helps the model make more accurate predictions. Moreover, to address the inherent complexities and variations in passenger flow patterns, the study combines the attention-based Bi-LSTM model with Spider Monkey Optimization (SMO). SMO is an optimization algorithm inspired by the social behavior of spider monkeys, and it enhances the model's ability to detect and adapt to changing patterns in passenger flow [7]. This combination leverages the strengths of both deep learning and optimization techniques, providing a more robust and adaptive approach to passenger flow prediction. The efficacy of the proposed model is validated using a pair of transit-passenger-flow datasets, which include diverse and representative samples of passenger flow data under various conditions [8]. The results from these datasets indicate that the accuracy of the Attention-based Bi-LSTM model is superior to that of baseline models, demonstrating the effectiveness of the proposed approach. The incorporation of the attention mechanism and SMO significantly elevates the prediction accuracy, making this approach more effective for practical applications in the Bus Transit System (BTS) [9]. This enhanced predictive capability can lead to better-informed decisions, ultimately contributing to improved transit service quality and passenger satisfaction.

2. LITERATURE REVIEW

Historically, statistical models such as autoregressive models (AR), moving average models (MA), and ARIMA (autoregressive integrated moving average) have been pivotal in predicting passenger flow. These methods leverage historical data to forecast future trends based on assumptions of linear relationships and stationary data patterns. While effective for capturing basic trends and seasonality, traditional statistical models struggle with nonlinearity and fail to incorporate dynamic external factors like weather conditions and special events [10]. This limitation significantly impacts their predictive accuracy, making them less reliable for real-time decision-making in bus station management. Moreover, these models are sensitive to outliers and require consistent data quality and rigorous preprocessing efforts to maintain accuracy, which can be resource-intensive and challenging to implement effectively in dynamic urban environments [11].

In recent years, machine learning techniques such as support vector machines (SVMs) and decision trees have gained popularity for their ability to handle nonlinear relationships and complex data patterns. SVMs, for instance, utilize kernel functions to transform data into higher-dimensional spaces, enabling them to capture intricate relationships in passenger behaviour [12]. Decision trees segment data into subsets based on feature values, offering interpretable models capable of handling categorical data effectively. However, both SVMs and decision trees may struggle with capturing temporal dependencies and long-term patterns inherent in passenger flow data [13]. SVMs can be computationally expensive and require meticulous parameter tuning to achieve optimal performance, while decision trees are susceptible to overfitting without proper regularization techniques. These challenges limit their applicability in accurately predicting passenger flow under varying conditions and in scenarios requiring robust generalization capabilities.

Ensemble learning methods, such as random forests and gradient boosting machines (GBMs), have emerged as powerful tools for enhancing prediction accuracy by combining multiple models. Random forests aggregate predictions from diverse decision trees to mitigate variance and improve robustness against noisy data and outliers [14]. GBMs sequentially build trees to minimize prediction errors, focusing iteratively on challenging data points to enhance accuracy. Despite their effectiveness, ensemble techniques necessitate significant computational resources and expertise for optimizing model parameters effectively. Moreover, interpreting ensemble models can be complex, posing challenges in transparent decision-making processes [15]. While these methods address some limitations of individual models, they still face constraints in handling the dynamic and nonstationary nature of passenger flow data, limiting their practical applicability in real-world transit management scenarios.

3. PROPOSED WORK

The proposed Bi-LSTM fusion model offers a novel approach to forecasting bus passenger flow by leveraging the bidirectional nature of LSTM networks. This model extracts intricate temporal features from time-series data by processing both past and future instances simultaneously. It comprises two LSTM layers: one handling forward data flow to capture features up to the prediction moment, and the other conducting backward calculation to extract features after the prediction moment. This ensures comprehensive information from both past and future contexts. During training, the model predicts in both directions, refining its understanding of temporal patterns. The predictions are then fused to generate the final output, h_t representing comprehensive bus passenger flow information at a specific time. This dual-directional processing enhances prediction accuracy and robustness, aiding transportation authorities in resource allocation, route optimization, and service planning. Additionally, the Bi-LSTM fusion model addresses challenges like traffic congestion management, demand forecasting, and public transportation efficiency, contributing to the development of smart and sustainable urban mobility systems.

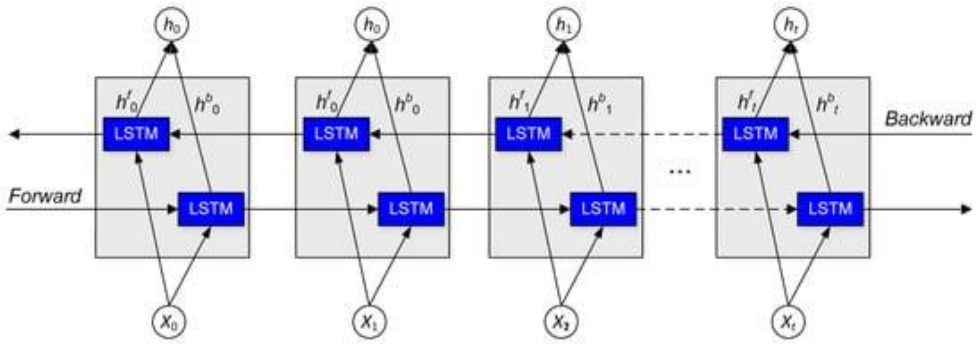

Fig. 1 The structure of Bi-LSTM network model.

Incorporating an attention mechanism with the Bi-LSTM model enhances the capture of temporal dynamics in transit passenger flow data. By assigning weights to features extracted by the Bi-LSTM model, the attention mechanism refines the analysis of variable relevance within the hidden layers. This is crucial for bus operations, where passenger volumes fluctuate across weekdays, weekends, and peak hours. The interaction between passenger numbers and preceding time periods highlights complex temporal dependencies in transit dynamics. As bus routes evolve, enhancing the Bi-LSTM model's ability to recognize long-term dependencies in time-series data is essential. The attention mechanism addresses this by dynamically assessing the significance of input properties over time, using key-value pairs assigned to a query. The resemblance between the query and each keyword determines the weighting coefficient, reflecting its importance in prediction.

$$A(Query, Source) = \Sigma Lxi = 1 Similarity(Query, keyi) * Value \qquad (1)$$

Here, $A(Query, Source)$represents the final weight assigned to the input features, Lx denotes the length of the original data, and $Similarity(Query, keyi)$signifies the resemblance between the query and the ith key. By computing this weighted sum, the attention mechanism effectively prioritizes and amplifies the influence of salient features within the time-sequence data, thereby enhancing the predictive accuracy of the Bi-LSTM model. This synergistic fusion of Bi-LSTM with an attention mechanism holds immense promise in advancing the modeling and forecasting capabilities in transit passenger flow analysis, offering insights crucial for optimizing bus operations and urban mobility management.

3.1 SMO algorithm

Fig.2 Spider Monkey Flow diagram.

In the initialization stage of the SMO algorithm, a population of N spider monkeys is generated, where each spider monkey SM_i (i =1, 2, ..., N) is represented as a D-dimensional vector. Here, D signifies the number of variables in the optimization problem, and each SM_i serves as a potential solution to be considered. This initial population is crucial as it lays the bedrock for the subsequent exploration and exploitation of the search space. The initialization process begins by defining the bounds for each dimension of the spider monkey's position. Let SM_{minj} and SM_{maxj} represent the lower limit and upper limit, respectively, for the jth dimension of the spider monkey SM_i. These bounds essentially confine the search space within which the spider monkeys will operate. To initialize each spider monkey SM_i, a random value is generated for each dimension within the specified bounds. This random value is drawn from an even distribution ranging between 0 and 1. The equation used for initialization is:

$$\mathrm{SM}_{ij} = SM_{minj} + U(0,1) \times (\mathrm{SM}_{maxj} - SM_{minj}) \quad (2)$$

SM_{ij} denotes the jth dimension value of the ith spider monkey SM_i. U(0,1) represents a uniformly distributed random number between 0 and 1. Multiplying this by the difference between the upper and lower bounds of the jth dimension and adding the lower bound ensures each spider monkey's position is within the search space. This initialization creates diverse solutions across the search space for the SMO algorithm. Each position is a candidate solution refined during optimization to converge towards an optimal solution. The algorithm starts by initializing a diverse population, setting leader boundaries, and defining a probability parameter. It then calculates the fitness of each individual, often quantified as the distance from vital food sources. Through a greedy selection process, global and local leaders are identified. During iterations, individuals update their positions based on personal experiences, local leaders' insights, and group wisdom. Probability computations guide individual selection towards convergence. The algorithm monitors local and global leadership efficacy, adjusting strategies if deviations are detected, and concludes when all criteria are fulfilled, achieving its objectives.

3.2 Local leader phase

During the Local Leader phase, each solution member (SM) adjusts its present position based on both the experience of its local leader and the experiences of other members within the local group. The fitness value of this newly computed position is then evaluated. If the fitness value of the updated position surpasses that of the previous one, the SM updates its position to this new value. The equation governing the position update for the i-th SM (part of the k-th local group) is given by:

$$\mathrm{SMnew}_{ij} = \mathrm{SM}_{ij} + U(0,1) \times U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}) \quad (3)$$

Here, SM_{ij} denotes the j-th dimension of the i-th SM's position, and LL_{kj} represents the j -th dimension of the position of the local group leader for the kk-th group. SM_{rj} is the j -th dimension of a randomly selected SM within the kk-th group, distinct from the i -th SM. $U(0,1)$is a evenly distributed random number between 0 and 1, while $U(-1,1)$ is a evenly distributed random number between -1 and 1.

Algorithm-1 Perturbation-Based Local Leader Update

for each member $sm_i \in$ kth group do
    for each j ∈ {1, ..., D} do
        if U(0, 1) ≥ pr then
$$\mathrm{SMnew}_{ij} = SM_{ij} + U(0,1) \times U(0,1) \times (LL_{kj} - sm_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij})$$
        Else
$$SMnew_{ij} = SM_{ij}$$
        end if
    end for
end for
position. The range of pr is [0.1, 0.8]

During the Local Leader Phase, the algorithm updates group members' positions by perturbing their dimensions. Each dimension j of member sm_i in group k is updated based on the perturbation rate pr (0.1 to 0.8). If a random number U(0,1) exceeds pr, the dimension is updated using the local leader's position $LL_{kj}$and a randomly selected member's position;

otherwise, it remains unchanged. This helps the swarm explore the solution space. The local leader's position is updated through greedy selection within the group, with the member having the best fitness value becoming the new local leader. If the local leader's position remains unchanged, the LocalLimitCount increments by 1 to monitor stagnation. If unchanged beyond a threshold (LocalLeaderLimit), a reinitialization process updates group members' positions using random initialization or combined data from the Global and Local Leaders. The position update follows this equation:

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - \mathrm{SM}_{ij}) + U(0,1) \times (SM_{ij} - LL_{kj}) \quad (4)$$

In this equation, the new position of each member (SM) is determined by two key influences: an attraction towards the global leader, represented by the term U(0,1)×(GLj−SMij)U(0,1)×(GLj−SMij), and a repulsion from the local leader, represented by the term U(0,1)×(SMij−LLkj)U(0,1)×(SMij−LLkj). This dual mechanism ensures that the members are guided towards potentially better solutions in the global context while being pushed away from the local leader to avoid local optima. The process focuses to enhance the diversity of the search space, thereby improving the chances of finding the global optimum.

3.3 Global leader phase

Following the completion of the LLP, the Global Leader phase (GLP) commences, ushering in a period where all SMs update their positions leveraging insights from the Global Leader and their local group members. The position update equation for this phase is articulated as follows:

$$\mathrm{SMnew}_{ij} = \mathrm{SM}_{ij} + U(0,1) \times U(0,1) \times (LL_{kj} - \mathrm{SM}_{ij}) + U(-1,1) \times (SM_{rj} - \mathrm{SM}_{ij}) \quad (5)$$

where GL_j denotes the jth dimension of the global leader position, and j spans the range from 1 to D, representing a randomly chosen index. During this phase, SM_i position is recalibrated based on a probability, $prob_i$, which is intricately linked to their fitness. Consequently, candidates with superior fitness stand a greater chance of enhancing their positions. This probability, $prob_i$, can be computed utilizing the expression.

$$prob_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i} \quad (6)$$

where $fitness_i$ signifies the fitness value of the ith SM. Additionally, the fitness of the newly generated positions of the SMs is evaluated and juxtaposed against their predecessors, with the superior option being embraced.

Algorithm 2: Iterative Position Updating Algorithm

count = 0;
while count < group size do
for each member $\mathrm{SM}_i \in$ group do
if U(0, 1) < probi then
  count = count + 1.
  Randomly select j ∈ {1...D}.
  Randomly select $\mathrm{SM}_r \in$ group s.t. r ≠ i.
  $SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - \mathrm{SM}_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij})$
    end if
  end for
end while

Algorithm 2 updates the positions of group members iteratively. A loop ensures each member can adjust their position, guided by probabilities and random selections, enhancing group dynamism and adaptability. This iterative process refines member positions, improving group performance and efficiency. A key operation in the algorithm is updating the global leader's position through greedy selection. The algorithm scans the population to identify the member with the optimal fitness level, replacing the current global leader with this elite member. A check ensures continuous evolution of the global leader's position. If this process stalls, the GlobalLimitCount variable is incremented by 1, monitoring potential stagnation. These procedures optimize the global leadership position, enhancing the algorithm's efficiency and efficacy.

4. RESULT

The dataset utilized is from Hangzhou, China, captures metro passenger flow over 25 days from January 1st to January 25th, 2019. It includes over 70 million swiping records from 81 subway stations, stored in daily CSV files, e.g., 'record_2019-01-01.csv'. It also contains a road network map, 'Metro_roadMap.csv'. With 23 days for training and 2 for testing, this dataset is ideal for analyzing metro passenger behavior and optimizing transit operations.

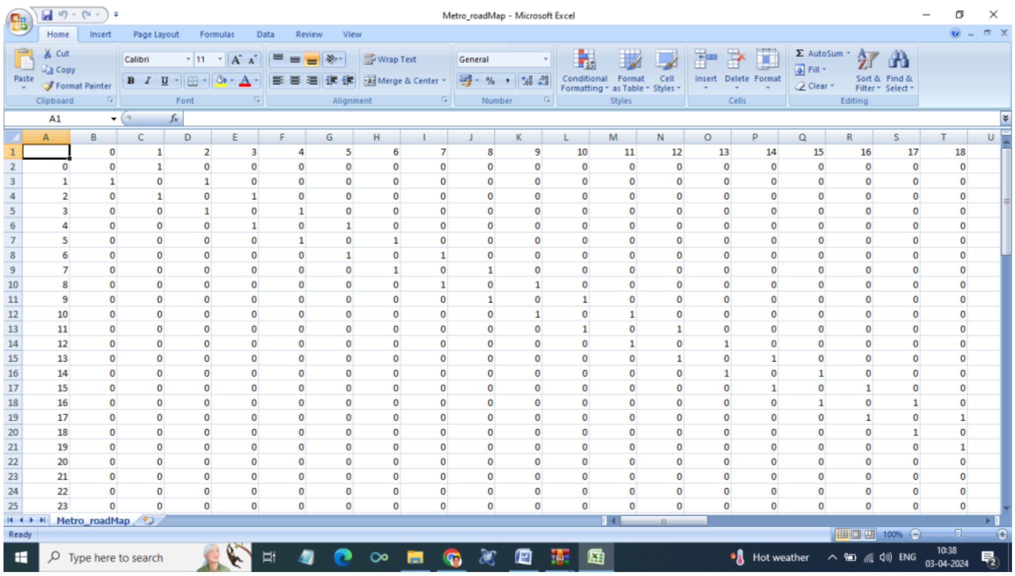

Fig.3. CSV Roadmap

The second dataset from Harbin, China, focuses on the city's bus transit network. It captures boarding information from the bus IC card system, including routes, card numbers, and times. Data from bus lines 363 and 68, covering March to October 2021, were filtered for peak hours (5:30 to 19:45). With 870,000 swipes and 30,000 records reserved for the last 5 days of testing, this dataset provides insights into bus patronage levels and aids in evaluating transit models and optimizing bus services in Harbin. After undergoing extensive training, the efficacy of the proposed model alongside various benchmark models is meticulously outlined in the provided table. A comprehensive examination of the table reveals that the predictive capabilities of the proposed model surpass those of conventional machine learning models, exemplified by the Support Vector Regression (SVR) model. This notable disparity underscores the superiority of the proposed model in accurately forecasting outcomes, signaling a significant advancement in predictive analytics.

TABLE I. TRAINING TIME COMPARISON OF VARIOUS MODELS

Models	**Training time(s)**
HA	115
SARIMA	112
GRU	123
DeepST	125
ST-ResNet	113
Proposed Model	137

Table 1 provides a succinct comparison of the training times (in seconds) for several models, including traditional methods like SARIMA, as well as deep learning architectures like GRU, DeepST, ST-ResNet, and a proposed model. The data showcases the varying computational requirements for training each model, with the proposed model exhibiting the longest training time at 137 seconds. This information aids in understanding the computational resources necessary for implementing these models in real-world scenarios, facilitating informed decision-making during model selection and deployment.

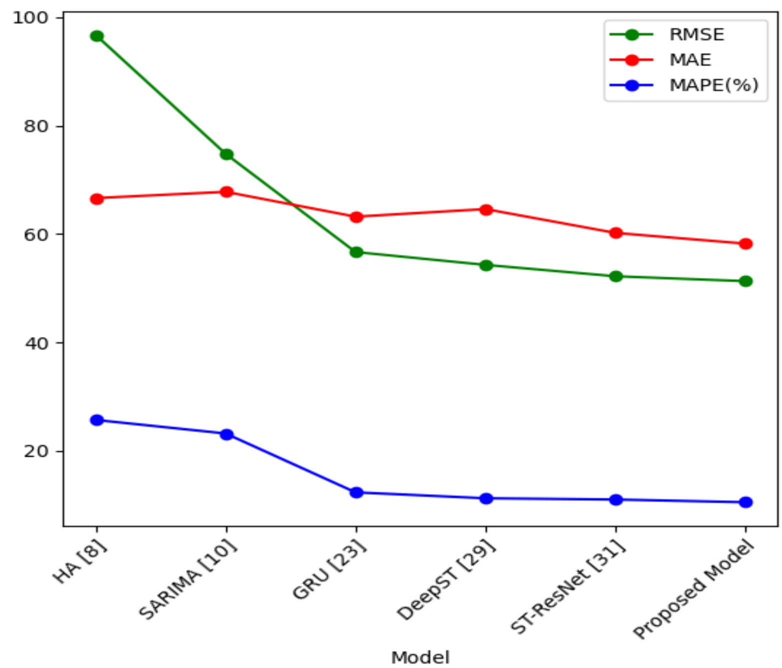

Fig.4. Evaluation Metrics for Dataset 1- 30 min

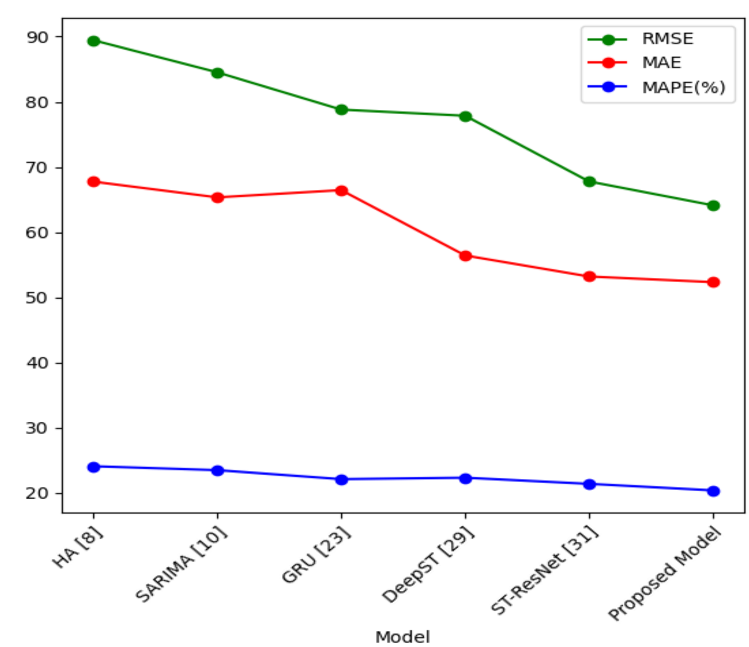

Fig.5. Evaluation Metrics for Dataset 1- 60 min

Figure 4, the proposed model shows a significant reduction in RMSE, MAE, and MAPE% compared to other models, indicating superior predictive performance. The RMSE and MAE decrease steadily across the models, with the proposed model achieving the lowest errors. Figure 5 similarly depicts the proposed model's performance, confirming its effectiveness. RMSE and MAE consistently decline as the models progress from HA to the proposed model, with the latter achieving the best results. MAPE% also shows a downward trend, further validating the proposed model's accuracy.

TABLE II. COMPARATIVE ANALYSIS OF PREDICTION MODELS

Models	Accuracy(%)
HA	66
SARIMA	68
GRU	63
DeepST	78
ST-ResNet	84
Proposed Model	89

Table.2 presents a comparative analysis of various prediction models based on their accuracy percentages. The models evaluated include the Historical Average (HA), SARIMA, GRU, DeepST, ST-ResNet, and a Proposed Model. The accuracy percentages range from 63% to 89%. Notably, the Proposed Model demonstrates the highest accuracy at 89%, outperforming all other models including state-of-the-art methods like ST-ResNet.

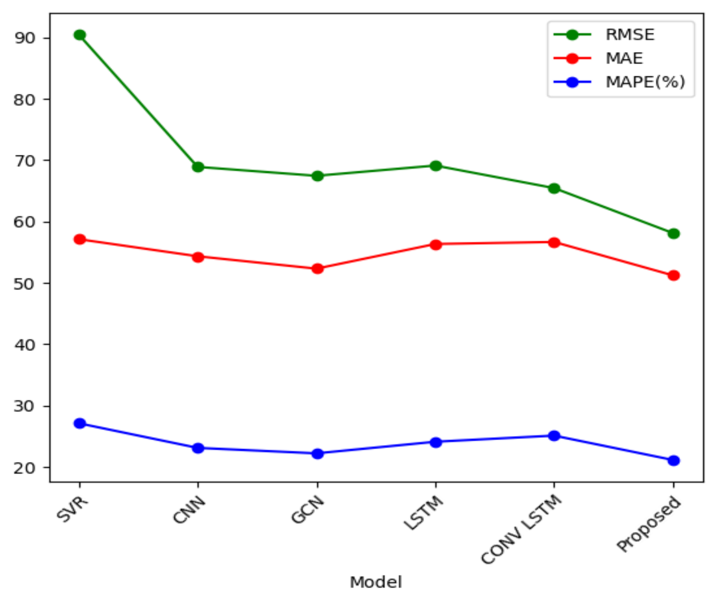

Fig. 6. Evaluation Metrics for Dataset 2- 30 min

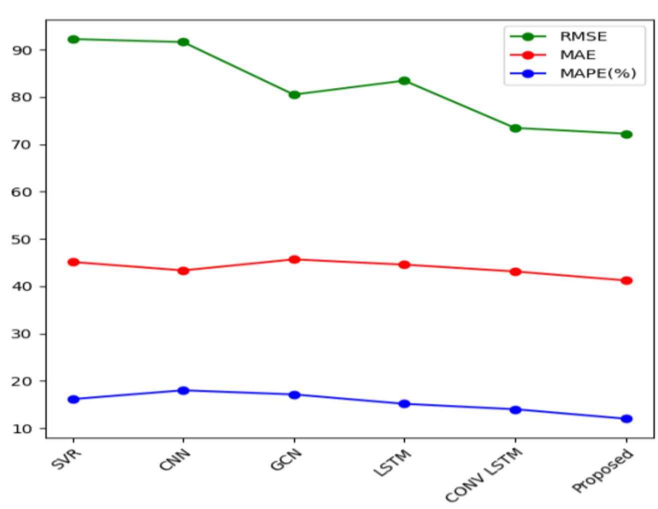

Fig. 7. Evaluation Metrics for Dataset 2- 60 min

Figure 6 and figure 7 compare the performance of SVR, CNN, GCN, LSTM, CONV LSTM, and a Proposed model using RMSE, MAE, and MAPE metrics. The Proposed model consistently has the lowest errors across all metrics, indicating it provides the most accurate predictions among the evaluated models.

TABLE III. TRAINING TIME COMPARISON OF PREDICTIVE MODELS

Models	**Training time(s)**
SVR	90
CNN	112
GCN	134
LSTM	121
CONV LSTM	116
Proposed	134

Table. 3 presents a comparison of the training times for various predictive models. The Support Vector Regression (SVR) model has the shortest training time at 90 seconds, while both the Graph Convolutional Network (GCN) and the proposed model have the longest training times at 134 seconds each. The Convolutional LSTM (CONV LSTM) , Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), models have intermediate training times of 112, 121, and 116 seconds, respectively. This information is crucial for understanding the computational efficiency of each model relative to its performance, as discussed in the broader analysis.

TABLE IV. COMPARATIVE ACCURACY OF PREDICTIVE MODELS

Models	**Accuracy(%)**
SVR	67
CNN	65
GCN	69
LSTM	64
CONV LSTM	72
Proposed	87

Table. 4 compares the accuracy percentages of various predictive models. The proposed model demonstrates a significantly higher accuracy (87%) compared to traditional machine learning models such as Support Vector Regression (SVR) with 67%, CNN with 65%, Graph Convolutional Networks (GCN) with 69%, Long Short-Term Memory (LSTM) with 64%, and Convolutional LSTM (CONV LSTM) with 72%. This indicates that the proposed model offers a substantial improvement in predictive performance over the other benchmark models.

5. CONCLUSION AND FUTURE WORK

Bus passenger flow prediction is challenging due to the dynamic and irregular nature of transportation systems. Variables like traffic conditions, weather changes, and fluctuating passenger demands complicate the development of accurate predictive models. This research utilizes a Bi-LSTM network to tackle these challenges. Bi-LSTM captures sequential dependencies by processing information in both ahead and behind directions, providing a comprehensive understanding of temporal patterns and improving prediction accuracy. To further enhance the model, Dynamic Feature Attention is employed, dynamically adjusting attention weights to emphasize the most relevant features at each time step. The model achieves superior accuracy in RMSE, MAE, and MAPE%, demonstrating its effectiveness in transit forecasting and potential for optimizing urban mobility. Future research could focus on enhancing computational efficiency and integrating real-time data streams to further refine predictions. Exploring hybrid approaches that combine deep learning with statistical methods could also improve prediction accuracy and robustness. These advancements aim to support more efficient public transportation systems, benefiting urban communities and stakeholders.

Conflicts Of Interest

None.

Funding

The authors explicitly states that the research project did not receive any funding from institutions or sponsors.

Acknowledgment

The authors would like to thank the administrative staff at the institution for their assistance and logistical support throughout the duration of this research.

References

[1] P. Kozlov, "Optimizing public transport services using AI to reduce congestion in metropolitan area," International Journal of Intelligent Automation and Computing, vol. 5, no. 2, pp. 1-14, 2022.

[2] M. Nama, A. Nath, N. Bechra, J. Bhatia, S. Tanwar, M. Chaturvedi, and B. Sadoun, "Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges," International Journal of Communication Systems, vol. 34, no. 9, e4814, 2021.

[3] A. Boukerche, Y. Tao, and P. Sun, "Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems," Computer Networks, vol. 182, 107484, 2020.

[4] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," Sustainability, vol. 12, no. 11, 4660, 2020.

[5] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1840-1852, 2020.

[6] B. He, R. Ma, W. Zhang, J. Zhu, and X. Zhang, "An improved generating energy prediction method based on bi-LSTM and attention mechanism," Electronics, vol. 11, no. 12, 1885, 2022.

[7] M. Montalvo-Martel, A. Ochoa-Zezzatti, E. Carrum, and D. Barzaga, "Design of an Urban Transport Network for the Optimal Location of Bus Stops in a Smart City Based on a Big Data Model and Spider Monkey Optimization Algorithm," in Technological and Industrial Applications Associated with Intelligent Logistics, 2021, pp. 167-201.

[8] X. Yang, Q. Xue, X. Yang, H. Yin, Y. Qu, X. Li, and J. Wu, "A novel prediction model for the inbound passenger flow of urban rail transit," Information Sciences, vol. 566, pp. 347-363, 2021.

[9] Z. Cheng, M. Trépanier, and L. Sun, "Incorporating travel behavior regularity into passenger flow forecasting," Transportation Research Part C: Emerging Technologies, vol. 128, 103200, 2021.

[10] A. AghaKouchak, B. Pan, O. Mazdiyasni, M. Sadegh, S. Jiwa, W. Zhang, ... and S. Sorooshian, "Status and prospects for drought forecasting: Opportunities in artificial intelligence and hybrid physical–statistical forecasting," Philosophical Transactions of the Royal Society A, vol. 380, no. 2238, 20210288, 2022.

[11] M. A. Fadhel, A. M. Duhaim, A. Saihood, A. Sewify, M. N. Al-Hamadani, A. S. Albahri, ... and Y. Gu, "Comprehensive systematic review of information fusion methods in smart cities and urban environments," Information Fusion, 102317, 2024.

[12] Q. Xu, "Incorporating CNN-LSTM and SVM with wavelet transform methods for tourist passenger flow prediction," Soft Computing, vol. 28, no. 3, pp. 2719-2736, 2024.

[13] M. Alizadeh and J. Ma, "High-dimensional time series analysis and anomaly detection: A case study of vehicle behavior modeling and unhealthy state detection," Advanced Engineering Informatics, vol. 57, 102041, 2023.

[14] J. Mulerikkal, S. Thandassery, V. Rejathalal, and D. M. D. Kunnamkody, "Performance improvement for metro passenger flow forecast using spatio-temporal deep neural network," Neural Computing and Applications, vol. 34, no. 2, pp. 983-994, 2022.

[15] J. Obregon and J. Y. Jung, "Explanation of ensemble models," in Human-Centered Artificial Intelligence, Academic Press, 2022, pp. 51-72