JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 10 Number 1 | June 2025: 12-21
DOI: 10.15575/join.v10i1.1362

Comparison Airport Traffic Prediction Performance Using BiGRU and CNN-BiGRU Models

Willy Riyadi[1], Jasmir[2], Xaverius Sika[3]
[1,2]Department of Computer Engineering, Universitas Dinamika Bangsa Jambi, Indonesia
[3]Department of Information Systems, Universitas Dinamika Bangsa Jambi, Indonesia

Article Info

Article history:

Received May 31, 2024
Revised December 17, 2024
Accepted December 17, 2024
Published April 01, 2025

Keywords:

Accuracy Prediction
Airport Traffic
BiGRU
CNN-BiGRU
COVID-19

ABSTRACT

COVID-19 pandemic has significantly disrupted the aviation industry, highlighting the critical need for accurate airport traffic predictions. This study compares the performance of BiGRU and CNN-BiGRU models to enhance airport traffic forecasting accuracy models from March to December 2020. Data preprocessing was performed using Python's Pandas library. This involved filtering, scaling using min-max normalization, and splitting the data into 80:20 training-testing split using Python's Pandas library. Various optimization techniques—RMSProp, Adam, Nadam, Adamax, AdamW, and Lion—were applied, along with ReduceLROnPlateau, to optimize model performance. The models were evaluated using Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). The best predictive performance was observed in the United States using the CNN-BiGRU model with the Adam optimizer, achieving the lowest MAE of 0.0580, MSE of 0.0097, and MAPE of 0.0979. The use of a balanced dataset, representing each airport's traffic as a percentage of a baseline period, significantly improved prediction accuracy. This research provides valuable insights for stakeholders seeking effective airport traffic prediction methods during unprecedented times.

Corresponding Author:

Willy Riyadi,
Computer Engineering Department, Faculty of Computer Science, Universitas Dinamika Bangsa Jambi
Jl. Jendral Sudirman, The Hok, Jambi, Indonesia.
Email: wriyadi5@gmail.com

1. INTRODUCTION

Airports significantly contribute to global connectivity, enabling seamless travel for millions of passengers and facilitating the transportation of goods [1]. The COVID-19 pandemic of 2020 had a devastating impact on the aviation industry, leading to significant disruptions and economic losses, leading to a sharp decline in air travel and the grounding of a substantial portion of the world's passenger aircraft fleet [2]. Recent advancements in air traffic prediction have led to a surge in interest in sophisticated techniques. Researchers have delved into a diverse array of methods, including statistical models, computational intelligence algorithms, and hybrid approaches, to analyze time series data and forecast future trends [3],[4]. By analyzing historical data, time series analysis enables us to uncover hidden patterns and make informed predictions about future events [5]. Among the models evaluated, the Gated Recurrent Unit (GRU) and the Long Short-Term Memory (LSTM) networks have demonstrated superior performance [6]. Although the LSTM model offers slightly better predictive accuracy than the GRU, it comes at the cost of significantly higher space complexity and longer training times [7]. To enhance predictive capabilities further, the Bidirectional Gated Recurrent Unit (Bi-GRU) is employed, capturing bidirectional information within time series data more effectively [8].

Numerous studies have explored techniques for predicting and analyzing airport operations. For instance, a recent study [9] proposed a hybrid model combining time-domain convolutional networks (TCNs) and bidirectional gated recurrent units (BiGRUs) to predict aircraft trajectories. This approach effectively captures both spatial and temporal dependencies in the data. Another study [10], investigated the use of CNN-BiGRU-AAM models for predicting traffic flow under various weather conditions. The results demonstrated the model's ability to accurately capture periodic patterns and distinguish between peak hours, even in adverse weather conditions. Additionally, research has been conducted on using TCN-DAGRU models [11] for predicting civil aircraft risks and BiGRU models[12], for real-time wave height prediction. These studies have shown the effectiveness of these models in capturing complex temporal dependencies and making accurate predictions. Finally, a hybrid 1D-CNN and attention-based Bi-GRU model [13], has been proposed for moisture content detection, demonstrating the potential of combining multiple techniques to improve prediction accuracy.

In this paper, we conduct our main contribution on comparative analysis of two advanced models for forecasting airport traffic: the Bidirectional Gated Recurrent Unit (BiGRU) and the CNN-BiGRU (Convolutional Neural Network – Bidirectional Gated Recurrent Unit). Our study utilizes multivariate time series data, focusing on airport traffic volumes expressed as a percentage relative to a reference period. We assess the models using key performance metrics, Mean Absolute Percentage Error (MAPE), including Mean Absolute Error (MAE), and Mean Squared Error (MSE). To ensure data quality and consistency, preprocessing steps are applied before model implementation. The preprocessed data is first fed into the CNN model to extract local features and sequential relations through CNN. While CNNs excel at capturing spatial information, they fall short in recognizing sequential correlations within the data [14]. To overcome this limitation, we incorporate bidirectional neural networks (BiGRU), which provide enhanced performance over unidirectional GRU networks by processing information in both forward and backward directions [15]. Our bidirectional GRU network is designed to extract critical features from data sequences in both temporal directions. Additionally, we introduce an attention mechanism to assign varying weights to each hidden layer, thereby improving prediction accuracy. This dual-directional approach enhances the network's ability to account for temporal dependencies within the sequence, leading to more accurate forecasting results [16].

Additionally, we investigate the use of various optimizers, such as Root Mean Squared Propagation (RMSProp), Adam, Nadam, Adamax, AdamW, and Lion, combined with ReduceLROnPlateau, for predicting airport traffic. ReduceLROnPlateau is employed to dynamically adjust the learning rate during training when the performance metric stops improving, thereby preventing overfitting and enhancing the predictive performance of the models. This study examines the performance of BiGRU and CNN-BiGRU models in forecasting airport traffic using data from the USA, Canada, Chile, and Australia. The findings of this research can provide valuable insights for improving airport operations and resource allocation.

2. METHOD

This study contributes to the field of airport traffic forecasting by systematically evaluating the performance of BiGRU and CNN-BiGRU models. We collected and preprocessed airport traffic data from Kaggle [17], followed by data preprocessing steps, including filtering, cleaning, and applying a MinMax Scaler. The data was then split into training (80%) and testing (20%) sets. We implemented recurrent layer models using GRU and CNN-GRU and explored the use of various optimizers, including Root Mean Square Propagation (RMSProp), Adam, Nadam, AdamW, Adamax, and Lion, to assess their impact on the model's predictive capabilities, followed by parameter tuning and the application of ReduceLROnPlateau to dynamically adjust the learning rate. The prediction results with default optimizers were compared to those obtained after parameter tuning, using performance metrics such as MAE and MAPE. Additionally, we provided prediction graphs for GRU and CNN-GRU results compared to the actual data. The processes and outcomes of these stages are illustrated in Figure 1 below.

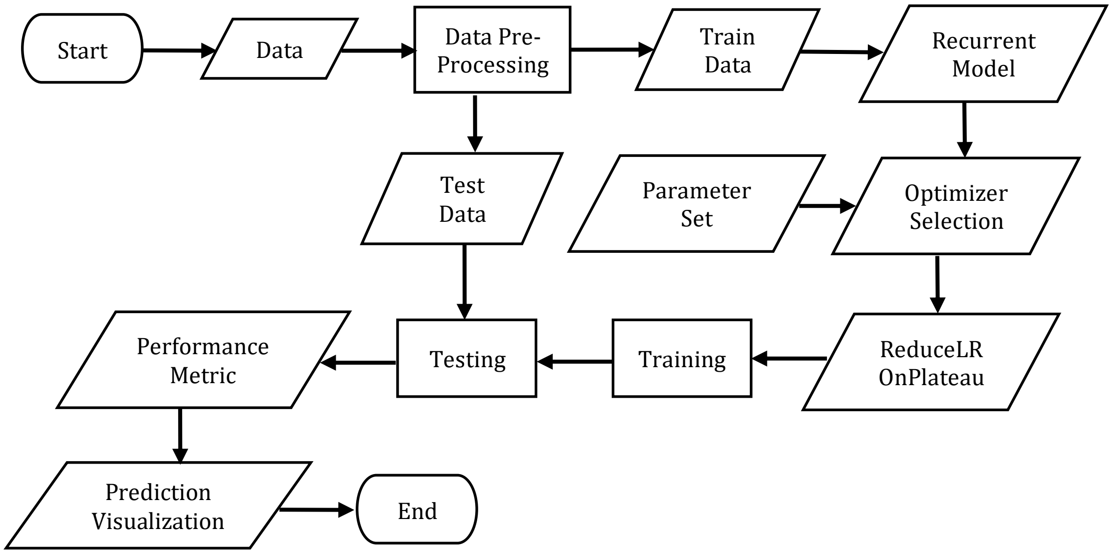

Figure 1. Flowchart Designof Airport Traffic

2.1. Dataset

In this study, we utilized a publicly available Kaggle dataset covering March 16, 2020, to December 12, 2020 [17]. The dataset, which includes 7,247 daily aggregated data points, features a single numerical attribute, 'PercentOfBaseline', that can be adjusted or reorganized. As detailed in Table 1.

Table 1. Dataset Describe Transpose

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
PercentOfBaseline	7247.0	66.651442	22.134433	0.0	53.0	67.0	84.0	100.0

The data was sourced from 27 airports across four countries: the United States (17 airports), Canada (9 airports), Chile (1 airport), and Australia (1 airport), as detailed in Table 2.

Table 2. Airport count

Country	Count
USA	17
Canada	9
Australia	1
Chile	1

2.2. Data Preprocessing

To ensure the accuracy of our analysis, we implemented key preprocessing steps: filtering, cleaning, and normalization with MinMaxScaler. Filtering removed extraneous data, focusing on essential parameters like 'Date', 'AirportName', and 'PercentOfBaseline', which reduced noise and bias. Data cleaning addressed inconsistencies, missing values, and errors, enhancing data quality for reliable analysis [18]. Finally, we normalized the 'PercentOfBaseline' values using MinMaxScaler, which scaled the data to a range between 0 and 1, ensuring consistent feature relationships and improving the model's learning efficiency. This process is mathematically represented by equation (1), where the MinMaxScaler adjusts values accordingly:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

2.3. Train/Test Split

To address the imbalance in airport data across countries, we used a methodological approach to enhance data representativeness. We calculated a daily average airport baseline for each country using the Pandas groupby().mean() function, ensuring a more balanced dataset that reflects overall airport activity within each country. This approach mitigates the dominance of data from countries like the USA, reducing bias and improving fairness in cross-country comparisons. While averaging may smooth out extreme values and reduce variability, potentially losing some detailed information, it enhances the model's generalizability by preventing over-reliance on data from countries with higher record volumes. We validated this approach by splitting the dataset into 80% training and 20% testing, applying the method to data from the USA, Canada, Chile, and Australia, as shown in Table 3, to assess its impact on model robustness and predictive accuracy.

Table 3. Train/Test split result

Country	Training (80%)	Testing (20%)	Total (100%)
United States of America	210	52	262
Chile	191	47	238
Canada	210	52	262
Australia	206	51	257

2.4. Algorithms

The Bidirectional Gated Recurrent Unit (BiGRU) is a powerful neural network architecture designed to address the challenges of long-term dependency and vanishing gradients in sequential data. By processing sequences in both forward and backward directions, BiGRU effectively captures long-range temporal dependencies, leading to improved model performance [19]. In this context, $\vec{h}_t$ and $\overleftarrow{h}_t$ represent the hidden-layer states of the forward and backward computations, respectively, as mathematically described in equation (2).

$$y_t = \sigma(\vec{h}_t, \overleftarrow{h}_t) \tag{2}$$

Our BiGRU Model Parameter settings are presented in Table 4:

Table 4. BiGRU Model Parameter Settings

Layer	Parameter	Value
BiGRU Layer 1	Units	64
	Activation Function	ELU
	Return Sequences	TRUE
Dropout Layer	Dropout Rate	0.2
BiGRU Layer 2	Units	32
	Activation Function	ELU
	Return Sequences	FALSE
Dense Layer	Units	1

This research employs a two-layer Bidirectional Gated Recurrent Unit (BiGRU) network. The first BiGRU layer is configured with return_sequences=True to capture long-range dependencies, while the second layer with return_sequences=False outputs the final sequence. A dropout layer is included to mitigate overfitting. Finally, a fully connected layer is used to leverage the extracted features and make the final prediction. Figure 2 illustrates the BiGRU model architecture used in this study, based on our parameter settings :

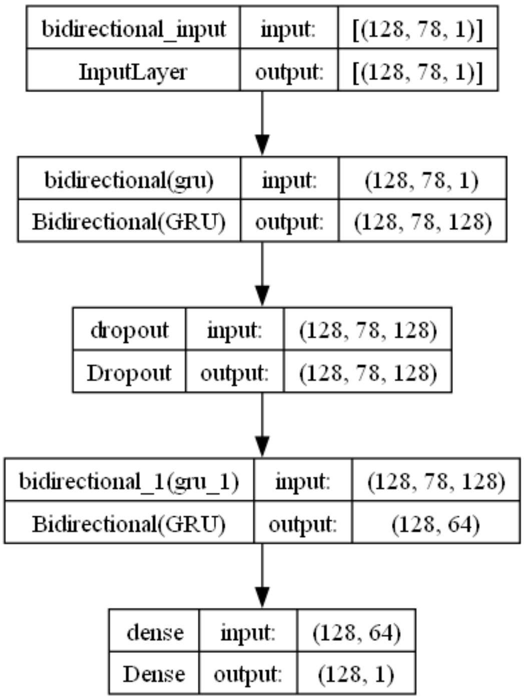

Figure 2. BiGRU Plot Model

This research employs one-dimensional Convolutional Neural Networks (1D CNNs) to process time series and text data, effectively capturing sequential contextual information [20]. In this study, 1D CNNs are utilized to extract temporal features. To optimize the performance of the deep CNN and GRU layers, careful hyperparameter tuning was conducted based on prior research and validated through performance evaluation. The specific parameter settings for our CNN-BiGRU model are detailed in Table 5.

Table 5. CNN-BiGRU Model Parameter Settings

Layer	Parameter	Value
Convolutional Layer 1	Filters	64
	Kernel Size	2
	Stride	1
	Activation Function	ELU
	Padding	Same
BiLSTM Layer 1	Units	64
	Activation Function	ELU
	Return Sequences	TRUE
Dropout Layer	Dropout Rate	0.2
Convolutional Layer 2	Filters	32
	Kernel Size	2
	Stride	1
	Activation Function	ELU
	Padding	Same
BiLSTM Layer 2	Units	32
	Activation Function	ELU
	Return Sequences	FALSE
Dense Layer	Units	1

This research employs a one-layer Conv1D network with a dropout layer to mitigate overfitting. This is followed by a two-layer BiGRU network, where the first layer is configured with return_sequences=True to capture temporal dependencies, and the second layer with return_sequences=False to output the final sequence. Finally, a fully connected layer is utilized to leverage the spatial correlation patterns extracted from the previous layers. Figure 3 illustrates the BiGRU model architecture used in this study, based on our parameter settings.

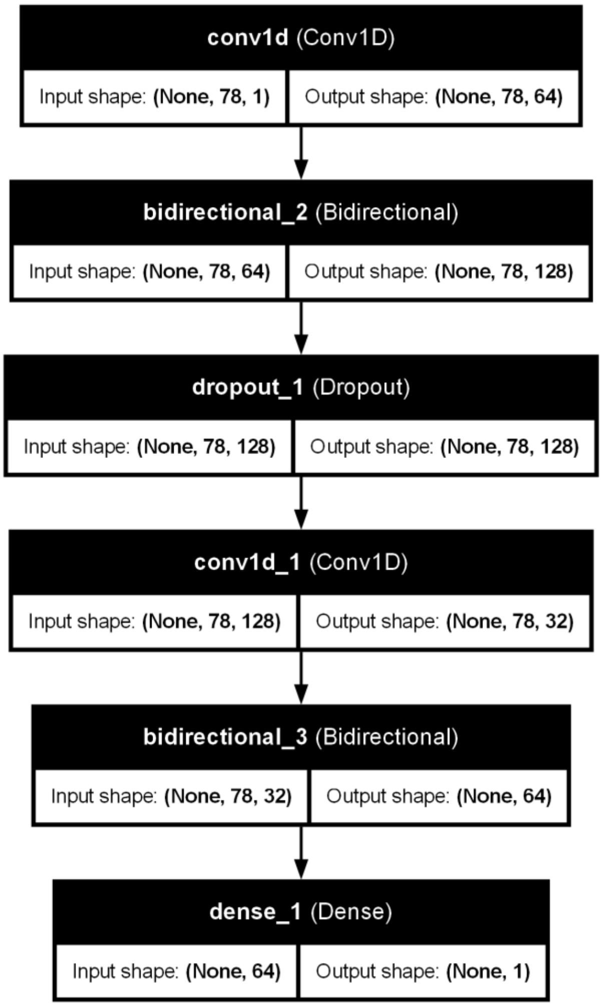

Figure 3. CNN-BiGRU Plot Model

2.4. Performance Metrics

To evaluate and compare the performance of the implemented methods, we calculate the Mean Absolute Error (MAE) using Equation (3). MAE measures the average magnitude of errors between actual and predicted airport baseline values, regardless of their direction. The actual airport baseline value is denoted as P_i, and the predicted value is denoted as $\hat{P}_i$.

$$MAE = \sum_{i=1}^{N} \frac{|P_i - \hat{P}_i|}{N} \quad (3)$$

Equation (4) is used to calculate the Mean Absolute Percentage Error (MAPE), a metric that assesses the relative accuracy of predictions. MAPE measures the average percentage difference between actual and predicted airport baseline values. Here, P_i represents the actual airport baseline value, $\hat{P}_i$ denotes the predicted value, and N is the total number of observations.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \left|\frac{P_i - \hat{P}_i}{P_i}\right| \times 100 \quad (4)$$

Equation (5) is used to calculate the Mean Squared Error (MSE) from a sample of N data points. It measures the average squared difference between the actual airport baseline values, P_i, and the predicted values, $\hat{P}_i$.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \left(P_i - \widehat{P_i}\right)^2 \quad (5)$$

3. RESULT AND DISCUSSION

We chose the optimizers RMSProp, Adam, Nadam, AdamW, Adamax, and Lion for their proven effectiveness in optimization tasks. RMSProp adapts learning rates for non-stationary problems, Adam combines RMSProp with momentum for robustness, Nadam adds Nesterov momentum for faster convergence, AdamW improves generalization by separating weight decay, Adamax handles large parameter spaces, and Lion efficiently manages large datasets. For model training, we used 60 epochs and a batch size of 32 to balance efficiency and performance. The ReduceLROnPlateau with patience=3 and factor=0.2 adjusted the learning rate to fine-tune the model, with min_delta=0.00001 and min_lr=0.00000001 ensuring significant improvements and preventing excessively small learning rates. We evaluated model performance using MAE, MSE, and MAPE. beta_1=0.00009, beta_2=0.00009, weight_decay=0.00001

We applied parameter tuning to optimize performance, setting specific hyperparameters for the RMSProp optimizer, we used specific hyperparameters, including rho=0.000001, which controls the moving average of squared gradients, weight_decay=0.000001 to help prevent overfitting by adding a regularization term to the loss function, and enabled exponential moving average (EMA) with ema_momentum=0.000001, which influences the smoothing factor, leading to more stable training. Similarly, for the Adam, Nadam, AdamW, Adamax, and Lion optimizers, we set beta_1=0.0001, determining the exponential decay rate for the first moment estimates (i.e., the mean of gradients), and beta_2=0.000001, weight_decay=0.000001, controlling the exponential decay rate for the second-moment estimates (i.e., the variance of gradients). EMA was also utilized with ema_momentum=0.000001, contributing to more stable training and validating them through grid search. Table 6 summarizes the performance metrics for various countries, with training conducted in a Python environment on Visual Studio Code. This process allowed us to assess the effectiveness of the optimization techniques in predicting airport traffic percentages.

Table 6. Performance in different optimizer

Country	Model	Optimizer	MAE	MSE	MAPE
USA	BiGRU	RMSprop	0.0598	0.0101	0.1007
		Adam	0.0614	0.0108	0.1037
		Nadam	0.0882	0.0129	0.1256
		Adamax	0.0593	0.0104	0.1006
		AdamW	0.0752	0.0105	0.1115
		Lion	0.0784	0.0108	0.1151
	CNN-BiGRU	RMSprop	0.0588	0.0106	0.1003
		Adam	0.058	0.0097	0.0979
		Nadam	0.1391	0.0291	0.209
		Adamax	0.0578	0.0105	0.0988
		AdamW	0.1382	0.029	0.208
		Lion	0.1427	0.0305	0.2145
Australia	BiGRU	RMSprop	0.1021	0.0137	0.3108
		Adam	0.0377	0.003	0.2302
		Nadam	0.0448	0.0038	0.2514
		Adamax	0.0373	0.0029	0.2532
		AdamW	0.0399	0.0032	0.2356
		Lion	0.0461	0.0039	0.2544
	CNN-BiGRU	RMSprop	0.0467	0.0037	0.2808
		Adam	0.0649	0.0067	0.275
		Nadam	0.0399	0.0033	0.237
		Adamax	0.0678	0.0072	0.2822
		AdamW	0.0383	0.0031	0.236
		Lion	0.0405	0.0033	0.239
Chile	BiGRU	RMSprop	0.1268	0.0225	0.2817
		Adam	0.1229	0.0214	0.2774
		Nadam	0.1302	0.0235	0.2873
		Adamax	0.1077	0.0173	0.268

Country	Model	Optimizer	MAE	MSE	MAPE
	CNN-BiGRU	AdamW	0.1251	0.0217	0.2828
		Lion	0.1222	0.0211	0.2774
		RMSprop	0.1261	0.0226	0.2789
		Adam	0.1235	0.0216	0.277
		Nadam	0.1271	0.0227	0.2795
		Adamax	0.1312	0.0239	0.2862
		AdamW	0.1275	0.0229	0.2822
		Lion	0.1128	0.0189	0.2626
Canada	BiGRU	RMSprop	0.0799	0.0156	0.1452
		Adam	0.076	0.0154	0.1401
		Nadam	0.1035	0.0232	0.1816
		Adamax	0.0746	0.0154	0.1388
		AdamW	0.0929	0.0206	0.1678
		Lion	0.0767	0.0153	0.1407
	CNN-BiGRU	RMSprop	0.0747	0.0154	0.1406
		Adam	0.0737	0.0155	0.1396
		Nadam	0.0716	0.0159	0.1375
		Adamax	0.1242	0.0285	0.2077
		AdamW	0.0751	0.0152	0.1394
		Lion	0.074	0.0155	0.1393

Figures 4 visually represent these results, showing prediction outcomes with the top optimizers for both models. The red line indicates actual baseline data, the blue line shows predictions by the GRU model, and the green line represents CNN-GRU predictions, allowing us to assess optimizer performance in predicting airport traffic percentages

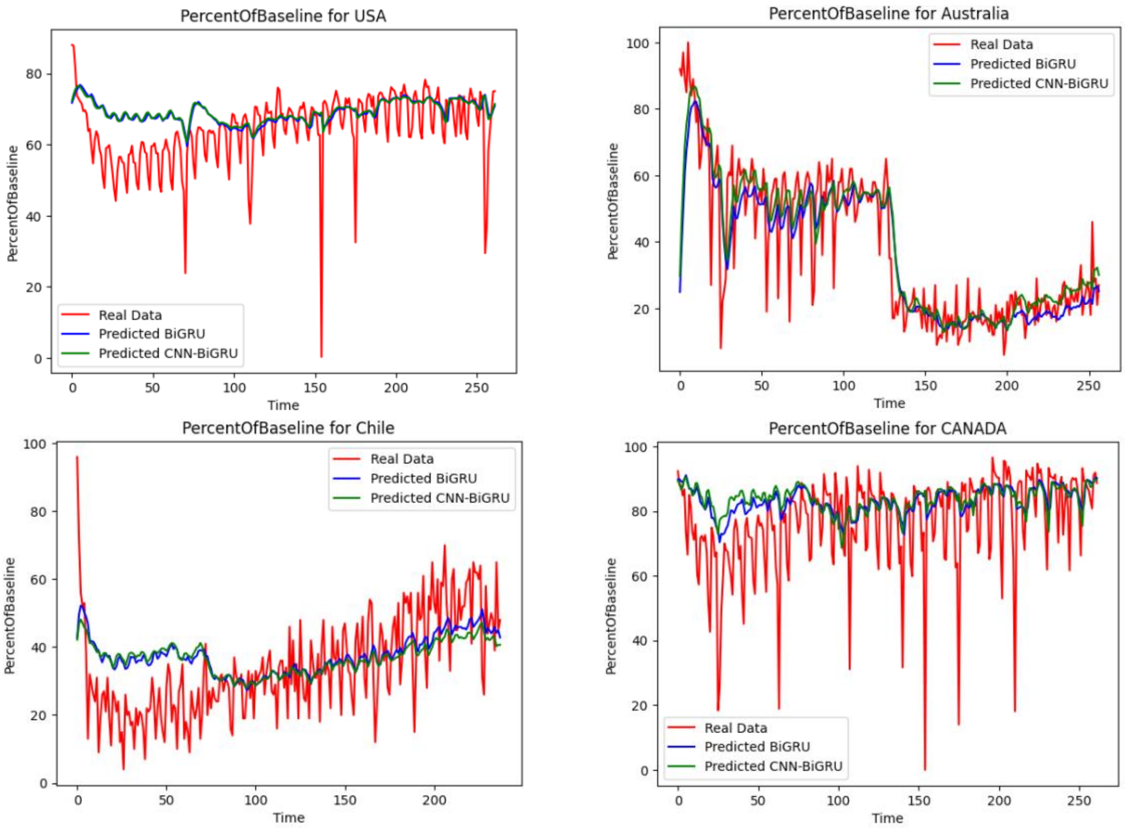

Figure 4. Best prediction result for each country

4. CONCLUSION

This study provides valuable insights into the application of deep learning models for airport baseline forecasting. While the BiGRU and CNN-BiGRU models demonstrated potential, their performance was constrained by data limitations due to the limited and imbalanced dataset. In terms of optimizer performance across countries, the CNN-BiGRU model generally outperformed the BiGRU model. USA: CNN-BiGRU with Adam (MAE: 0.0580, MSE: 0.0097, MAPE: 0.0979) ranked better than BiGRU with Adamax. Australia: BiGRU with Adamax (MAE: 0.0373, MSE: 0.0029) slightly outperformed CNN-BiGRU with AdamW, though the latter had a lower MAPE. Chile: CNN-BiGRU with Lion (MAPE: 0.2626) achieved a lower MAPE, but BiGRU with Adamax had better MAE and MSE. Canada: CNN-BiGRU with Nadam (MAE: 0.0716, MAPE: 0.1375) performed better overall than BiGRU with Adamax. Future research should prioritize the collection of high-quality, large-scale datasets to enable the development of more robust and accurate forecasting models. Additionally, exploring innovative techniques such as transfer learning and domain adaptation can help mitigate the impact of data scarcity in specific regions. By addressing these challenges, we can advance the field of airport baseline forecasting and contribute to improved operational efficiency and passenger experience.

ACKNOWLEDGEMENTS

This research was a collaborative effort, supported by LPPM Universitas Dinamika Bangsa. The authors would like to thank their colleagues at Universitas Dinamika Bangsa for their contributions, including providing data, sharing expertise, and offering critical feedback. While the interpretations and conclusions presented in this paper are those of the authors, we acknowledge the valuable input of our collaborators.

REFERENCES

[1] Z. Yang, Y. Wang, J. Li, L. Liu, J. Ma, and Y. Zhong, "Airport Arrival Flow Prediction considering Meteorological Factors Based on Deep-Learning Methods," *Complexity*, vol. 2020, p. 6309272, 2020, doi: 10.1155/2020/6309272.

[2] S. V Gudmundsson, M. Cattaneo, and R. Redondi, "Forecasting temporal world recovery in air transport markets in the presence of large economic shocks: The case of COVID-19," *J Air Transp Manag*, vol. 91, p. 102007, 2021, doi: https://doi.org/10.1016/j.jairtraman.2020.102007.

[3] X. Zhang, H. Liu, Y. Zhao, and X. Zhang, "Multifractal detrended fluctuation analysis on air traffic flow time series: A single airport case," *Physica A: Statistical Mechanics and its Applications*, vol. 531, p. 121790, 2019, doi: https://doi.org/10.1016/j.physa.2019.121790.

[4] H. Liu, X. Zhang, and X. Zhang, "Multiscale multifractal analysis on air traffic flow time series: A single airport departure flight case," *Physica A: Statistical Mechanics and its Applications*, vol. 545, p. 123585, 2020, doi: https://doi.org/10.1016/j.physa.2019.123585.

[5] D. C. Tascón and O. Díaz Olariaga, "Air traffic forecast and its impact on runway capacity. A System Dynamics approach," *J Air Transp Manag*, vol. 90, p. 101946, 2021, doi: https://doi.org/10.1016/j.jairtraman.2020.101946.

[6] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, 2021, doi: https://doi.org/10.1016/j.neucom.2021.02.046.

[7] R. Tang, Z. Yang, J. Lu, H. Liu, and H. Zhang, "Real-time Trajectory Prediction of Unmanned Aircraft Vehicles Based on Gated Recurrent Unit," 2022, pp. 585–596. doi: 10.1007/978-981-16-5429-9_45.

[8] X. Li, X. Ma, F. Xiao, C. Xiao, F. Wang, and S. Zhang, "Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA)," *J Pet Sci Eng*, vol. 208, p. 109309, Jan. 2022, doi: 10.1016/j.petrol.2021.109309.

[9] J. Huang and W. Ding, "Aircraft Trajectory Prediction Based on Bayesian Optimised Temporal Convolutional Network–Bidirectional Gated Recurrent Unit Hybrid Neural Network," *International Journal of Aerospace Engineering*, vol. 2022, pp. 1–19, Dec. 2022, doi: 10.1155/2022/2086904.

[10] S. Liu, W. Lin, Y. Wang, D. Z. Yu, Y. Peng, and X. Ma, "Convolutional Neural Network-Based Bidirectional Gated Recurrent Unit–Additive Attention Mechanism Hybrid Deep Neural Networks for Short-Term Traffic Flow Prediction," *Sustainability*, vol. 16, no. 5, p. 1986, Feb. 2024, doi: 10.3390/su16051986.

[11] D. Zhou, X. Zhuang, J. Cai, H. Zuo, X. Zhao, and J. Xiang, "An ensemble model using temporal convolution and dual attention gated recurrent unit to analyze risk of civil aircraft," *Expert Syst Appl*, vol. 236, p. 121423, Feb. 2024, doi: 10.1016/j.eswa.2023.121423.

[12] F. Meng, T. Song, D. Xu, P. Xie, and Y. Li, "Forecasting tropical cyclones wave height using bidirectional gated recurrent unit," *Ocean Engineering*, vol. 234, p. 108795, Aug. 2021, doi: 10.1016/j.oceaneng.2021.108795.

[13] Q. Yuan, J. Wang, M. Zheng, and X. Wang, "Hybrid 1D-CNN and attention-based Bi-GRU neural networks for predicting moisture content of sand gravel using NIR spectroscopy," *Constr Build Mater*, vol. 350, p. 128799, Oct. 2022, doi: 10.1016/j.conbuildmat.2022.128799.

[14] N. Zhao, H. Gao, X. Wen, and H. Li, "Combination of Convolutional Neural Network and Gated Recurrent Unit for Aspect-Based Sentiment Analysis," *IEEE Access*, vol. 9, pp. 15561–15569, 2021, doi: 10.1109/ACCESS.2021.3052937.

[15] L. Zeng, W. Ren, and L. Shan, "Attention-based bidirectional gated recurrent unit neural networks for well logs prediction and lithology identification," *Neurocomputing*, vol. 414, pp. 153–171, Nov. 2020, doi: 10.1016/j.neucom.2020.07.026.

[16] P. Li *et al.*, "Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation," *ISPRS Int J Geoinf*, vol. 9, no. 11, p. 635, Oct. 2020, doi: 10.3390/ijgi9110635.

[17] T. Shin, “COVID-19’s Impact on Airport Traffic,” kaggle. Accessed: Jan. 10, 2023. [Online]. Available: https://www.kaggle.com/datasets/terenceshin/covid19s-impact-on-airport-traffic

[18] I. M. Pires, F. Hussain, N. M. M. Garcia, P. Lameski, and E. Zdravevski, “Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification,” *Future Internet*, vol. 12, no. 11, p. 194, Nov. 2020, doi: 10.3390/fi12110194.

[19] Z. Zhang *et al.*, “An Improved Bidirectional Gated Recurrent Unit Method for Accurate State-of-Charge Estimation,” *IEEE Access*, vol. 9, pp. 11252–11263, 2021, doi: 10.1109/ACCESS.2021.3049944.

[20] W. Choi, M.-J. Kim, M.-S. Yum, and D.-H. Jeong, “Deep Convolutional Gated Recurrent Unit Combined with Attention Mechanism to Classify Pre-Ictal from Interictal EEG with Minimized Number of Channels,” *J Pers Med*, vol. 12, no. 5, p. 763, May 2022, doi: 10.3390/jpm12050763.