

Performance Prediction of Airport Traffic Using LSTM and CNN-LSTM Models

Willy Riyadi, Jasmir

Universitas Dinamika Bangsa, Jambi, Indonesia

Article Info

Article history:

Received May 01, 2023

Revised Juny 15, 2023

Accepted July 27, 2023

Keywords:

Accuracy Prediction

Airport Traffic

CNN-LSTM

COVID-19

LSTM

ABSTRACT

During the COVID-19 pandemic, airports faced a significant drop in passenger numbers, impacting the vital hub of the aircraft transportation industry. This study aimed to evaluate whether Long Short-Term Memory Network (LSTM) and Convolutional Neural Network - Long Short-Term Memory Network (CNN-LSTM) offer more accurate predictions for airport traffic during the COVID-19 pandemic from March to December 2020. The studies involved data filtering, applying min-max scaling, and dividing the dataset into 80% training and 20% testing sets. Parameter adjustment was performed with different optimizers such as RMSProp, Stochastic Gradient Descent (SGD), Adam, Nadam, and Adamax. Performance evaluation uses metrics that include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and R-squared (R²). The best LSTM model achieved an impressive MAPE score of 0.0932, while the CNN-LSTM model had a slightly higher score of 0.0960. In particular, the inclusion of a balanced data set representing a percentage of the base period for each airport had a significant impact on improving prediction accuracy. This research contributes to providing stakeholders with valuable insights into the effectiveness of predicting airport traffic patterns during these unprecedented times.

Copyright ©2022 The Authors.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Willy Riyadi, +6281366054142

Faculty of Computer Science, Department of Computer Engineering,

Universitas Dinamika Bangsa, Jambi, Indonesia,

Email: wriyadi5@gmail.com

How to Cite:

W. Riyadi and J. Jasmir, "Performance Prediction of Airport Traffic Using LSTM and CNN-LSTM Models", Matrik : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 22, no. 3, pp. 627-638, Jul. 2023.

This is an open access article under the CC BY-SA license (<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

Airports serve as crucial terminals for aircraft takeoff, landing, and passenger transfer, playing a pivotal role in the transportation industry [1]. However, the COVID-19 pandemic in 2020 profoundly impacted the aviation sector, resulting in a significant decrease in passenger numbers and the non-operation of an estimated 17,000 aircraft fleets from the peak period from March to June [2]. Such fluctuations in air traffic are a characteristic feature of the aviation industry [3] and are time-dependent, making them suitable for analysis as Time Series data [4]. The air transport industry has been focusing on traffic forecasting methodologies for several decades, while formal studies and academic research on this topic have emerged relatively recently, around three decades ago [5]. Various forecasting techniques have been developed to analyze Time Series data, including statistical methods, computational intelligence, or a combination of both [6]. The primary goal of Time Series data analysis is to utilize historical observations to develop accurate models that reflect the underlying structure of the series. These models enable the prediction and classification of future events [7]. In recent years, deep learning methods and techniques have garnered significant attention in academic research and have been successfully applied to real-world prediction problems, including the analysis of Time Series data. Among these techniques, Long short-term memory (LSTM) and Convolutional Neural Networks (CNN) have emerged as popular and effective deep learning models [8]. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), including LSTM, have demonstrated considerable improvements in handling Time Series data and have shown superior performance, particularly when dealing with raw data [9].

Previous research has already been conducted on analyzing and predicting airports. One study, carried out by [10], utilized a Bayesian Network (BN) approach to predict the level of congestion at airports and the reliability of flight delays. This research achieved a performance rate of 70%. In another study conducted by [11], a machine learning approach was employed, utilizing LightBGM, Multilayer perceptron (MLP), and Random Forests (RF) Classifier. This study aimed to evaluate the accuracy of predicting departure delays, arrival delays, and cancellations. The obtained accuracy score was above 0.794, indicating a high level of accuracy. Furthermore, [12] conducted a study using a deep graph-embedded LSTM (DGLSTM) model, which demonstrated higher prediction accuracy and robustness compared to the existing mainstream methods. Moreover, another study conducted by [13] focused on accurately predicting flight delays using a CNN-LSTM model. This model achieved a classification accuracy of 92.39% and an overall accuracy rate of 84%, surpassing the performance of several benchmark models. Additionally, [14] conducted a study to reduce airport analysis prediction errors. They compared the prediction error of a Conv-LSTM model with a single LSTM and found that the Conv-LSTM model reduced the prediction error by 11.41%. Furthermore, the study introduced an Att-Conv-LSTM model, which further reduced the prediction error by 10.83% compared to the Conv-LSTM model.

In this paper, we present our main contributions, which involve comparing the performance of two powerful models, LSTM (Long Short-Term Memory) and CNN-LSTM (Convolutional Neural Network Long Short-Term Memory), for predicting airport traffic. Our analysis focuses on multivariate time series data, specifically the traffic to and from the airport represented as a percentage of the traffic volume during a reference period. Our evaluation focuses on essential indices such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and R-squared (R²). By contrasting our models with previous approaches that utilized autoregressive moving average (ARMA) models, we aim to assess their effectiveness thoroughly. Based on the research cited in [15], the CNN-LSTM hybrid model proves to be superior to the LSTM model, exhibiting an average reduction of 21.62% in prediction error. Additionally, as mentioned in [16], the CNN-LSTM hybrid model achieves an 18% decrease in Mean Squared Error (MSE) compared to the LSTM model. Furthermore, according to [17], the CNN-LSTM model outperforms other models, achieving an impressive MSE score of 0.37 and demonstrating its excellence when compared to the LSTM model. Moreover, the other research highlights that both the CNN-LSTM and LSTM models outperform the CNN and the back-propagation neural network (BPNN). On the other hand, [18] indicates that the performance of CNN or LSTM models falls short when compared to the traditional SARIMA model.

While previous research has delved into comparing LSTM and CNN-LSTM models, our study introduces a novel aspect by employing distinct datasets for predicting airport traffic patterns in forecasting airport percent of baseline during the COVID-19 period. By using different datasets, we aim to determine which model provides more accurate predictions for this critical application. One of the challenges in this research lies in the imbalanced nature of the airport dataset across different countries. To address this issue, we took the initiative to create the average percentage of airport baseline daily for four representative countries: the USA, Canada, Chile, and Australia. We employed various optimizers, including RMSProp, Stochastic Gradient Descent (SGD), Adam, Nadam, and Adamax, in fine-tuning the model parameters, and they can significantly impact the predictive accuracy. The findings of this research have significant implications for airport management and various stakeholders in making informed choices during these unprecedented times.

2. RESEARCH METHOD

This quantitative research data is analyzed using numerical methods to quantify patterns, encompassing data collection from Kaggle repositories, data preprocessing including data filtering and MinMaxScaler, data partitioning into train 80% and test set 20%, data modeling using LSTM and CNN-LSTM, parameter tuning using RMSProp, Stochastic Gradient Descent (SGD), Adam, Nadam, and Adamax, and performance evaluation using metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and R-squared (R²). The process and outcomes of these stages are illustrated in Figure 1.

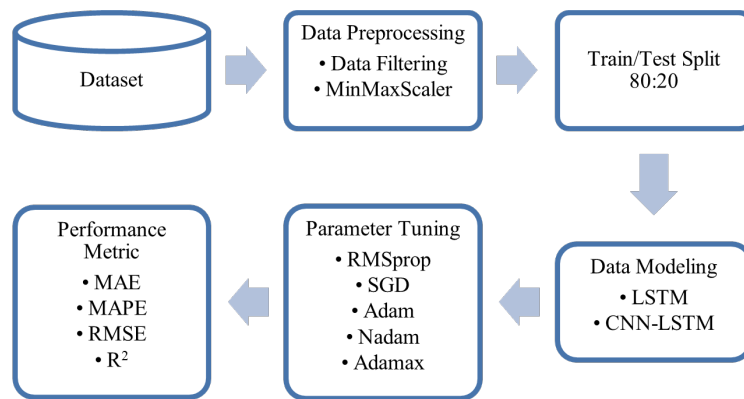


Figure 1. Research Stages

2.1. Dataset

In this study, we utilize a dataset from March 16, 2020, to December 12, 2020, and readily accessible on Kaggle [19]. The dataset comprises a single numerical attribute that can be transposed, and it includes a total of 7247 baseline data points with daily aggregation 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

2.2. Data Preprocessing

In this stage of the research, the focus is on cleaning irrelevant data using data filtering and performing normalization using techniques such as feature scaling or min-max scaler. Normalization aims to adjust the range of independent variables or features within the data so that they fall between 0 and 1 [20]. This process helps ensure that the data values for each attribute are in a standardized and consistent range, as in Equation (1):

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

2.3. Train/Test Split

The dataset used in this study is divided into training data and testing data through the Train/Test Split process. In this case, the dataset is split in an 80/20 ratio. Due to the imbalanced nature of the airport dataset across different countries, we took the initiative to create the average airport baseline daily on the basis for USA, Canada, Chile, and Australia.

2.4. Data Modeling

The LSTM component of the model consists of two layers. The dropout layer plays a crucial role in mitigating overfitting. In the final stage, the fully connected layer or dense layer utilizes the spatial correlation patterns extracted from the previous layers to predict future values. By leveraging these patterns, the model can make accurate estimations about the forthcoming values, facilitating its ability to forecast future outcomes as illustrated in Figure 2 :

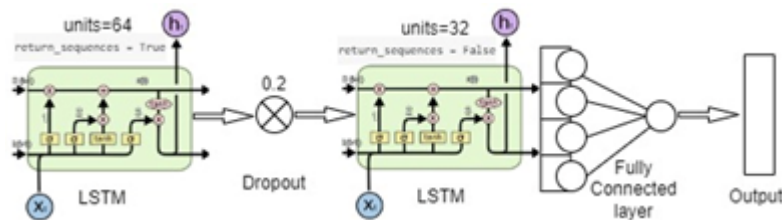


Figure 2. Flow diagram CNN-LSTM model

The CNN-LSTM model comprises a Conv1D layer and an AveragePooling1D layer. The purpose of these layers is to capture relevant patterns and features in the input data. Additionally, to prevent overfitting, a Dropout layer is included. The LSTM component of the model consists of two layers. In the final stage, the fully connected layer or dense layer utilizes the spatial correlation patterns extracted from the previous layers to predict future values. By considering these patterns, the model can make informed estimations regarding the future values of the time series, as illustrated in Figure 3.

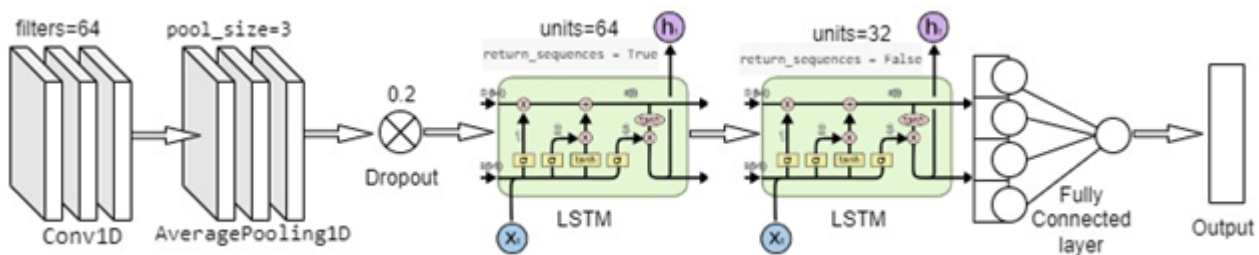


Figure 3. Flow diagram CNN-LSTM model

2.5. Parameter Tuning

We utilize various optimization algorithms to enhance performance in deep learning models, RMSProp (Root Mean Square Propagation) is an optimization algorithm that aims to adjust the learning rate based on the magnitude of recent gradients, Stochastic Gradient Descent (SGD) is a fundamental optimization algorithm which updates the model parameters by taking small steps proportional to the negative gradient of the loss function, while Adam (Adaptive Moment Estimation) maintains a running average of past gradients and their squares, and uses this information to update the model parameters, Nadam is an extension of Adam that incorporates Nesterov accelerated gradient (NAG) into its update rule and calculates the gradient using an intermediate point ahead of the current parameters, which helps the algorithm to converge faster, and Adamax is another variant of Adam that uses the infinity norm (maximum absolute value) of the gradients and uses the exponentially weighted infinity norm to update the parameters.

2.6. Performance Metric

To evaluate and compare the performances of the implemented methods, Equation (2) is used to calculate MAE from a sample of N data points. Assume y_i and \hat{y}_i are variables of paired observations that express the same phenomenon.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2)$$

Equation (3) is used to calculate the MAPE. MAPE is a measure prediction accuracy of a CNN-LSTM method in statistics, for example, in trend prediction. Where y_i is the actual airport baseline value and \hat{y}_i is the predicted value N.

$$MAPE = \frac{100\%}{N} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{3}$$

Equation (4) is used to calculate the RMSE. If \hat{y}_i is a vector of n predictions generated from a sample of N data points on all airport baseline variables, and y_i is the vector of the observed variable of the airport baseline being predicted.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \tag{4}$$

Equation (5) is used to calculate the R-squared represents the coefficient of how well the predicted values fit \hat{y} compared to the original values y_i . The value from 0 to 1 is interpreted as percentages. The higher the value is, the better the model is.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} \tag{5}$$

3. RESULT AND ANALYSIS

3.1. Dataset

Based on the 7247 datasets, we have summarized the number of data entries for different countries. In the United States of America (USA), we have 4441 data entries. For Australia, the dataset contains 257 data entries. In Chile, we have 238 data entries; in Canada, there are 2311. These numbers can be visualized in Figure 4.

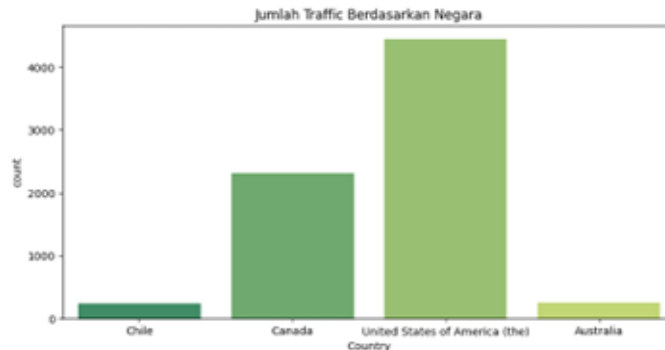


Figure 4. Traffic volume by country

3.2. Data Preprocessing

To ensure the relevance and accuracy of the analysis, a filtering process was applied to eliminate irrelevant data. The results of this filtering process are presented in Table 2, which contains only the data deemed relevant for the analysis.

Table 2. Data Filtering

Before	After
'AggregationMethod', 'Date', 'Version', 'Airport-Name', 'PercentOfBaseline', 'Centroid', 'City', 'State', 'ISO_3166_2', 'country', 'Geography'	'Date', 'AirportName', 'PercentOfBaseline', 'City', 'State', 'country'

The MinMaxScaler rescales the data features, such as attributes or variables, to a specified range, typically between 0 and 1. This scaling process maintains the relative relationships between the values of different features, ensuring that the data retains its original distribution and structure, presented in Table 3.

Table 3. Data Scaling Example

Original Value	Scaling Value
88.058824	1.0
87.823529	0.99732038
81.705882	0.92765031
73.764706	0.8372132

3.3. Train/Test Split

Considering the imbalanced distribution of data entries across various countries in the airport dataset, we took the initiative to address this issue by creating an average airport baseline on a daily basis. This process involved using the Pandas dataframe.groupby().mean() function. The dataset was divided into an 80/20 ratio, and the mean values were calculated for the USA, Canada, Chile, and Australia, resulting in average baseline values, presented in Table 4.

Table 4. Train/Test Split

Country	Data Training (80%)	Data Testing (20%)	Total (100%)
United States of America	210	52	262
Chile	190	48	238
Canada	210	52	262
Australia	206	51	257

3.4. Data Modeling

In this LSTM model, we implemented a 2-layer architecture. The initial layer consisted of 64 units with return sequences set to True, allowing it to produce outputs for each time step. To mitigate overfitting, we incorporated a dropout rate of 0.2, which randomly set a portion of the input units to 0 during training. The second layer of the LSTM model consisted of 32 units with return sequences set to False. During training, we optimized the models using the Rectified Linear Unit (ReLU) activation function, which helps the model capture non-linear relationships in the data. We trained the models for a total of 60 epochs, with a batch size of 64. For a visual representation of the USA LSTM model architecture, please refer to Figure 5.

```
# Initialising the RNN using LSTM
model_USA_LSTM = Sequential()
model_USA_LSTM.add(LSTM(64, activation='relu', return_sequences = True, input_shape = (X_train1.shape[1], 1)))
model_USA_LSTM.add(Dropout(0.2))
model_USA_LSTM.add(LSTM(32, activation='relu', return_sequences = False))
model_USA_LSTM.add(Dense(1))
model_USA_LSTM.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 52, 64)	16896
dropout (Dropout)	(None, 52, 64)	0
lstm_1 (LSTM)	(None, 32)	12416
dense (Dense)	(None, 1)	33

Total params: 29,345
Trainable params: 29,345
Non-trainable params: 0

Figure 5. LSTM model

In the CNN-LSTM model, we utilized a Conv1d layer with 64 filters. The kernel size was set to 3, the padding was set to same, and the strides were set to 1. This convolutional layer helped extract relevant features from the input data. Furthermore, we included an AveragePooling1D layer with a pool size of 32, strides of 1, and padding set to same. This pooling layer helped down-sample the data and capture important temporal patterns. To prevent overfitting, we incorporated a dropout rate of 0.2. During training, the

dropout layer randomly sets input units to 0. In the LSTM layer, we utilized 64 units with return sequences set to True. Additionally, we added another LSTM layer with 32 units, where return sequences were set to False. We employed the Rectified Linear Unit (ReLU) activation function throughout the training process. We trained the models for a total of 60 epochs, with a batch size of 64. For a visual representation of the USA CNN-LSTM model models for a total of 60 epochs, with a batch size of 64. For a visual representation of the USA CNN-LSTM model architecture, please refer to Figure 6.

```
# Initialising the CNN using LSTM
model_USA_ConvLSTM = Sequential()
model_USA_ConvLSTM.add(Conv1D(filters=64, kernel_size=3, strides=1, activation='relu', padding='same', input_shape=(X_train2.shape[1],1)))
model_USA_ConvLSTM.add(AveragePooling1D(pool_size=3, strides=1, padding='same'))
model_USA_ConvLSTM.add(Dropout(0.2))
model_USA_ConvLSTM.add(LSTM(64, activation='relu', return_sequences = True))
model_USA_ConvLSTM.add(LSTM(32, activation='relu', return_sequences = False))
model_USA_ConvLSTM.add(Dense(1))
model_USA_ConvLSTM.summary()

Model: "sequential_1"
-----
Layer (type)                Output Shape              Param #
-----
conv1d (Conv1D)              (None, 52, 64)           256
average_pooling1d (AverageP (None, 52, 64)           0
ooling1D)
lstm_2 (LSTM)                (None, 52, 64)           33024
lstm_3 (LSTM)                (None, 32)               12416
dense_1 (Dense)              (None, 1)                 33
-----
Total params: 45,729
Trainable params: 45,729
Non-trainable params: 0
```

Figure 6. LSTM model

3.5. Performance Metric

The performance of the models was evaluated using various metrics, including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and R-Squared. Table 5 presents the performance metric scores for different countries: the USA, Canada, Chile, and Australia. The scores were calculated using different optimizers, including RMSProp, Stochastic Gradient Descent (SGD), Adam, Nadam, and Adamax. By analyzing these scores, we can assess the effectiveness of the models in predicting the airport percentage of baseline for each country.

Table 5. Performance of ReLu Activation in Different Optimizers

Country	Model	Optimizer	MAE	MAPE	RMSE	R2
United States of America (USA)	LSTM	RMSprop	0.0505	0.0927	0.0822	0.4404
		SGD	0.0716	0.1247	0.1021	0.1350
		Adam	0.0532	0.0968	0.0842	0.4126
		Nadam	0.0516	0.0926	0.0806	0.4617
		Adamax	0.0524	0.0932	0.0804	0.4635
	CNN-LSTM	RMSprop	0.0516	0.0961	0.0857	0.3910
		SGD	0.0711	0.1243	0.1023	0.1316
		Adam	0.0507	0.0947	0.0849	0.4018
		Nadam	0.0513	0.0936	0.0827	0.4324
		Adamax	0.0561	0.0960	0.0801	0.4675
Australia	LSTM	RMSprop	0.0713	0.3303	0.1169	0.5771
		SGD	0.1393	0.4608	0.1828	-0.0341
		Adam	0.0730	0.3370	0.1179	0.5696
		Nadam	0.0767	0.3535	0.1213	0.5451
		Adamax	0.0738	0.3351	0.1185	0.5658
	CNN-LSTM	RMSprop	0.0742	0.3221	0.1136	0.5639
		SGD	0.1168	0.4271	0.1552	0.1858
		Adam	0.0750	0.3181	0.1147	0.5549
		Nadam	0.0763	0.3355	0.1165	0.5408
		Adamax	0.0759	0.3452	0.1152	0.5511

Country	Model	Optimizer	MAE	MAPE	RMSE	R2
Chile	LSTM	RMSprop	0.0886	0.4456	0.1146	0.4364
		SGD	0.1176	0.5387	0.1501	0.0321
		Adam	0.0880	0.4440	0.1119	0.4623
		Nadam	0.0897	0.4401	0.1144	0.4378
		Adamax	0.0890	0.4588	0.1149	0.4334
	CNN-LSTM	RMSprop	0.0881	0.4302	0.1125	0.4571
		SGD	0.1165	0.5344	0.1486	0.0514
		Adam	0.0874	0.4422	0.1130	0.4522
		Nadam	0.0877	0.4325	0.1122	0.4599
		Adamax	0.0872	0.4382	0.1119	0.4623
Canada	LSTM	RMSprop	0.0963	0.1505	0.1391	-0.0316
		SGD	0.0989	0.1555	0.1413	-0.0647
		Adam	0.0989	0.1523	0.1382	-0.0182
		Nadam	0.0979	0.1505	0.1368	0.0023
		Adamax	0.0954	0.1491	0.1375	-0.0079
	CNN-LSTM	RMSprop	0.1043	0.1553	0.1347	0.0324
		SGD	0.1005	0.1565	0.1403	-0.0496
		Adam	0.1032	0.1554	0.1400	-0.0452
		Nadam	0.1098	0.1583	0.1352	0.0260
		Adamax	0.1080	0.1582	0.1379	-0.0130

Table 5 displays the performance of different optimizers for LSTM and CNN-LSTM models in various countries. Evaluating the models based on the R-squared metric, we observed the following trends: For the USA LSTM and CNN-LSTM models, the Adamax optimizer achieved better performance. For the Australia LSTM model and CNN-LSTM model, the RMSprop optimizer showed superior results. In the case of the Chile LSTM model, the Adam optimizer outperformed other optimizers. For the Chile CNN-LSTM model, the Nadam optimizer demonstrated better performance. The Canada LSTM model achieved better results with the Nadam optimizer. Finally, for the Canada CNN-LSTM model, the RMSprop optimizer yielded superior performance. Finally, for the Canada CNN-LSTM model, the RMSprop optimizer yielded superior performance.

The results of the LSTM and CNN-LSTM models, utilizing the optimizers with the best performance based on the R-squared score tests, are visually depicted in Figure 7 to Figure 10. These figures provide a graphical representation of the model predictions and actual airport percentage of baseline. In these visualizations, the red line represents the actual values of the airport percentage of baseline. The blue line corresponds to the predictions obtained from the LSTM model, while the green line represents the predictions derived from the CNN-LSTM model. By analyzing these graphical representations, we can assess the accuracy and effectiveness of the chosen optimizers in predicting the airport percentages of baseline. Comparing the predicted lines to the actual values allows us to evaluate how well the models capture the underlying patterns and trends in the data.

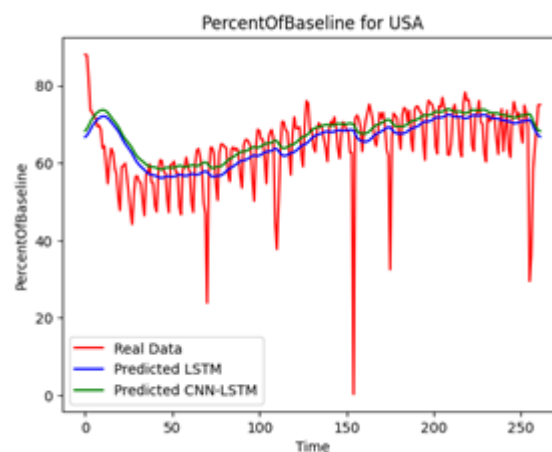


Figure 7. United states of America (USA) prediction result

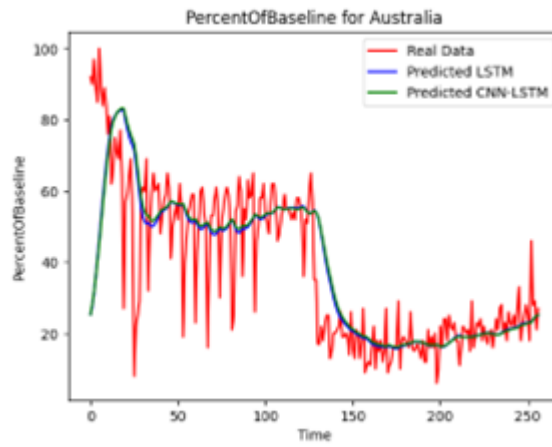


Figure 8. Australia prediction result

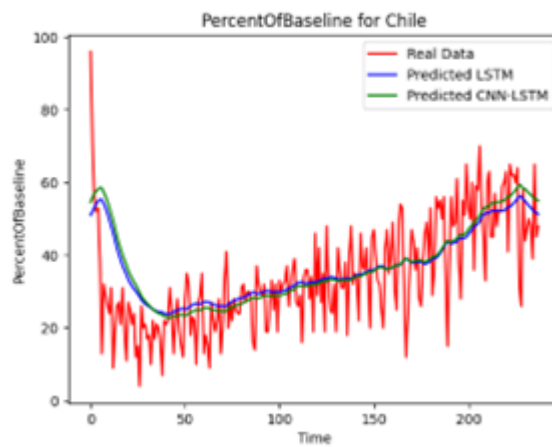


Figure 9. Chile prediction result

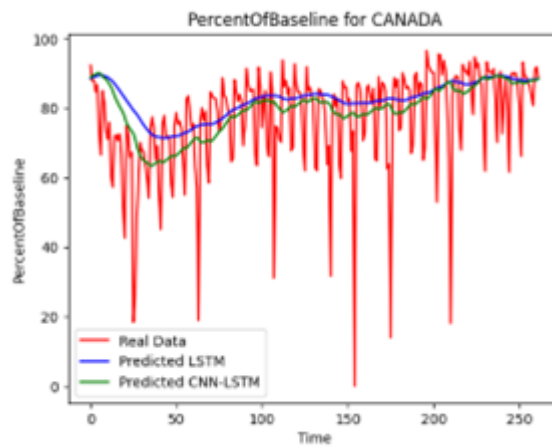


Figure 10. Canada prediction result

The USA dataset gained our highest-performing LSTM model in this research and achieved an impressive MAPE score of 0.0932. The CNN-LSTM model for the USA had a slightly higher MAPE score of 0.0960. On the other hand, the Chile dataset gained the lowest MAPE score achieved by the CNN-LSTM model, which achieved an impressive score of 0.4382. Conversely, the LSTM model for Chile had a slightly higher MAPE score of 0.4440. Table 6 compares our results with previous research, utilizing the Mean Absolute Percentage Error (MAPE) metric. A lower MAPE score suggests more accurate predictions.

Table 6. MAPE Score Result Comparison with Previous Research

Model	MAPE Score
LSTM [17]	35.78
CNN-LSTM [17]	31.84
LSTM [21]	20.2824
CNN-LSTM [21]	20.8055
LSTM [22]	44.06
CNN-LSTM [22]	40.38
LSTM (USA)	0.0932
CNN-LSTM (USA)	0.096

4. CONCLUSION

After analyzing a total of 7,247 recorded baseline data points, we calculated the average data for each country on the corresponding dates, resulting in 1,019 baseline data points. However, it is important to acknowledge that the prediction performance varied across different countries: the USA, Canada, Chile, and Australia. This variation can be attributed to the imbalanced dataset and the limited data availability, with only around ten months of data for each airport. The USA demonstrated the best prediction performance among the countries analyzed, while Chile demonstrated the lowest prediction performance. Our best LSTM model achieved a MAPE score of 0.0932, while the CNN-LSTM model achieved a slightly higher MAPE score of 0.0960. These scores indicate a relatively low average percentage error in the predictions. To enhance the accuracy of future predictions, we recommend conducting further research addressing the data imbalance issue. Incorporating a more balanced dataset for each country, with a more comprehensive and representative sample of data, can potentially lead to more reliable and precise prediction results.

5. ACKNOWLEDGEMENTS

The authors express sincere gratitude to Universitas Dinamika Bangsa for their support in this research. The authors also like to extend their appreciation to the anonymous reviewers and editors for providing insightful ideas and suggestions to enhance the quality of this research.

6. DECLARATIONS

AUTHOR CONTRIBUTION

The first authors performed data collection and data forecasting. The second authors performed data analysis and preprocessing. All authors carried out revision and nalization of the manuscript.

FUNDING STATEMENT

This research was supported by Universitas Dinamika Bangsa.

COMPETING INTEREST

The Authors state that this research has no conict of interest.

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, pp. 1–11, 2020.

- [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," *Journal of Air Transport Management*, vol. 91, pp. 1–8, 2021.
- [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, pp. 1–16, 2019.
- [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, pp. 1–50, 2020.
- [5] D. C. Tascón and O. Díaz Olariaga, "Air traffic forecast and its impact on runway capacity. A System Dynamics approach," *Journal of Air Transport Management*, vol. 90, pp. 1–10, 2021.
- [6] H. Abbasimehr and R. Paki, "Improving time series forecasting using LSTM and attention models," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 673–691, 2022.
- [7] 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.
- [8] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNNLSTM model for gold price time-series forecasting," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 351–17 360, 2020.
- [9] J. Yu, A. de Antonio, and E. Villalba-Mora, "Deep Learning (CNN, RNN) Applications for Smart Homes: A Systematic Review," *Computers*, vol. 11, no. 2, 2022.
- [10] Á. Rodríguez-Sanz, F. G. Comendador, R. A. Valdés, J. Pérez-Castán, R. B. Montes, and S. C. Serrano, "Assessment of airport arrival congestion and delay: Prediction and reliability," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 255–283, 2019.
- [11] M. Lambelho, M. Mitici, S. Pickup, and A. Marsden, "Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions," *Journal of Air Transport Management*, vol. 82, pp. 1–11, 2020.
- [12] W. Zeng, J. Li, Z. Quan, and X. Lu, "A Deep Graph-Embedded LSTM Neural Network Approach for Airport Delay Prediction," *Journal of Advanced Transportation*, vol. 2021, pp. 1–15, 2021.
- [13] Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction," *Expert Systems with Applications*, vol. 227, 2023.
- [14] J. Qu, M. Xiao, L. Yang, and W. Xie, "Flight Delay Regression Prediction Model Based on Att-Conv-LSTM," 2023.
- [15] L. Ma and S. Tian, "A Hybrid CNN-LSTM Model for Aircraft 4D Trajectory Prediction," *IEEE Access*, vol. 8, pp. 134 668–134 680, 2020.
- [16] A. Vidal and W. Kristjanpoller, "Gold volatility prediction using a CNN-LSTM approach," *Expert Systems with Applications*, vol. 157, pp. 1–9, 2020.
- [17] T. Y. Kim and S. B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy*, vol. 182, pp. 72–81, 2019.
- [18] K. He, L. Ji, C. W. D. Wu, and K. F. G. Tso, "Using SARIMACNNLSTM approach to forecast daily tourism demand," *Journal of Hospitality and Tourism Management*, vol. 49, pp. 25–33, 2021.
- [19] T. Shin, "COVID-19's Impact on Airport Traffic," 2021.
- [20] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020.
- [21] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, "Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models," *Renewable Energy*, vol. 177, pp. 101–112, 2021.
- [22] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," *IEEE Access*, vol. 8, pp. 180 544–180 557, 2020.

[This page intentionally left blank.]