Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 3, September 2024, pp. 580-593
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i3.29659

Comparative Analysis of Optimizer Effectiveness in GRU and CNN-GRU Models for Airport Traffic Prediction

Willy Riyadi, Jasmir
Department of Computer Engineering, Universitas Dinamika Bangsa, Jambi, Indonesia

ARTICLE INFO

Article history:

Received July 30, 2024
Revised August 13, 2024
Published September 16, 2024

Keywords:

Airport Traffic;
GRU;
CNN-GRU;
Covid-19;
Optimizer Effectiveness;
Parameter Tuning

ABSTRACT

The COVID-19 pandemic has posed significant challenges to airport traffic management, necessitating accurate predictive models. This research evaluates the effectiveness of various optimizers in enhancing airport traffic prediction using Deep Learning models, specifically Gated Recurrent Units (GRU) and Convolutional Neural Network-Gated Recurrent Units (CNN-GRU). We compare the performance of optimizers including RMSprop, Adam, Nadam, AdamW, Adamax, and Lion, and analyze the impact of their parameter tuning on model accuracy. Time series data from airports in the United States, Canada, Chile, and Australia were used, with preprocessing steps like filtering, cleaning, and applying a MinMax Scaler. The data was split into 80% for training and 20% for testing. Our findings reveal that the Adam optimizer paired with the GRU model achieved the lowest Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) in the USA. The study underscores the importance of selecting and tuning optimizers, with ReduceLROnPlateau used to adjust the learning rate dynamically, preventing overfitting and improving model convergence. However, limitations include dataset imbalance and region-specific results, which may affect the generalizability of the findings. Future research should address these limitations by developing balanced datasets and exploring optimizer performance across a broader range of regions and conditions. This study lays the groundwork for further investigating sustainable and accurate airport traffic prediction models.

Corresponding Author:

Willy Riyadi, Department of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jalan Jendral Sudirman, Thehok, Jambi Selatan, Jambi, Indonesia, 36138
Email: willyriyadi@gmail.com

1. INTRODUCTION

Airports play a pivotal role in the global transportation network [1], facilitating the movement of millions of passengers and vast amounts of cargo daily [2]. Efficient management of airport traffic is essential for maintaining the smooth operation of the aviation industry [3], ensuring that flights are on time, resources are optimized, and safety is upheld [4]. However, the COVID-19 pandemic in 2020 had profoundly impacted the aviation sector [5]. During the peak period in mid-April 2020, there was a significant decline in passenger numbers, and approximately 17,000, or 64%, of the world's total passenger aircraft fleet were inactive. [6], [7]. Fluctuations in air traffic are common in the aviation industry [8], [9], and these variations over time make the data suitable for time series analysis [10], [11]. Time series data is obtained from observations made sequentially over time [12]. Forecasting airport traffic accurately is crucial for operational efficiency strategic planning and resource allocation, helping airlines, airports, and regulators make informed decisions [13]. This importance has become even more pronounced in recent years, as the aviation industry faces challenges such as changing travel patterns and increasing environmental concerns [14], underscoring the need for robust forecasting models [15], [16]. Research related to Forecasting airports with time series data generally has two main objectives: to understand or model the stochastic mechanisms factors within the data and to forecast future observations based on historical values [17], [18], [19].

Over the past few decades, the air transportation industry has undergone a methodological revolution in forecasting air traffic [20], [21]. Although academic research on this topic has emerged relatively recently, about three decades ago, various forecasting techniques have been diligently studied to analyze time series data [22], [23]. These techniques range from statistical methods to computational intelligence and even combinations to develop accurate models that can precisely predict and classify future events [24], [25]. Previous research typically falls into three main categories regarding prediction methods: flight plan-based algorithms, traffic flow-driven algorithms, and data-driven algorithms [26], [27]. In this context, Gated Recurrent Units (GRU) and Convolutional Neural Networks (CNN) have demonstrated superior performance in time series data prediction, making them particularly suitable for air traffic forecasting [28], [29]. CNN models are effective in capturing spatial dependencies within air traffic flow data, revealing underlying relationships [30]. Meanwhile, GRU models excel in handling the dynamic variations and addressing the vanishing gradient problem often encountered in recurrent neural networks, thus providing more refined and accurate predictions [31]. This study focuses on GRU and CNN-GRU models due to their advanced capabilities in overcoming the limitations of traditional forecasting methods and their potential to improve the accuracy of air traffic predictions.

Several previous studies have focused on analyzing and predicting airport traffic. In the research by [32], the GRU model significantly reduced prediction errors at each time step, particularly in forecasting aircraft vertical speed, which is crucial for enhancing landing efficiency. This highlights the GRU model's strength in handling time series data with complex temporal dependencies. Similarly, the study by [33] emphasized the advantages of combining a Temporal Convolutional Network (TCN) with BiGRU to extract both spatial and temporal features from aircraft trajectories. This approach not only improved feature extraction but also reduced time complexity, demonstrating the potential for hybrid models in air traffic prediction. Meanwhile, [34] explored the use of a GRU model with dual attention gates to learn contextual information from Aviation Safety Reporting System reports. The introduction of an attention mechanism allowed the model to focus on critical information, thereby improving the resolution of long-term dependencies in the data. This study underscored the value of attention mechanisms in enhancing model interpretability and prediction accuracy. Another study [35], conducted experiments using the GRU model on hourly traffic data, as opposed to the more granular 5-minute observations, to lower processing costs. The optimized GRU model achieved a notable performance improvement, with a 4.5% higher average gain value compared to the standard untuned model. This finding suggests that model optimization can lead to significant gains in prediction accuracy, even when using less granular data. Conversely, the study by [36], combined Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) with transfer learning to effectively capture wind speed characteristics across different regions. The results demonstrated the model's ability to make accurate short-term wind speed forecasts, highlighting the effectiveness of combining CNNs and GRUs for complex prediction tasks. These studies collectively illustrate the evolving landscape of forecasting methods in the context of air traffic and other time-sensitive domains. By synthesizing the strengths and limitations of previous research, this study aims to build upon these advancements, specifically focusing on the GRU and CNN-GRU models. The choice of these models is informed by their proven ability to handle temporal and spatial dependencies, address long-term dependency issues, and their adaptability to various optimization techniques, making them well-suited for predicting airport traffic.

In this paper, we present our main contribution: a comparative performance analysis of the GRU and CNN-GRU models combined with ReduceLROnPlateau for predicting airport traffic. ReduceLROnPlateau is necessary because it helps to dynamically adjust the learning rate during training when the performance metric stops improving, thereby preventing overfitting and ensuring better convergence of the models. We explore 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. We also compare the performance of these optimizers with their default options against those obtained through parameter tuning to determine the most effective configurations. This analysis focuses on time series data of airport traffic obtained from various regions, namely the United States, Canada, Chile, and Australia. For evaluation, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are chosen because they provide clear and interpretable measures of prediction accuracy, with MAE offering a straightforward average error metric and MAPE allowing for easy comparison across different datasets by expressing errors as percentages. Through this evaluation, we aim to determine the effectiveness and accuracy of the best optimizer in GRU and CNN-GRU models in predicting airport traffic, providing valuable insights for future studies in this field while also ensuring long-term sustainability.

2. METHODS

This research involves a series of systematic stages. The process began with acquiring datasets from Kaggle repositories [37], 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 Fig. 1.

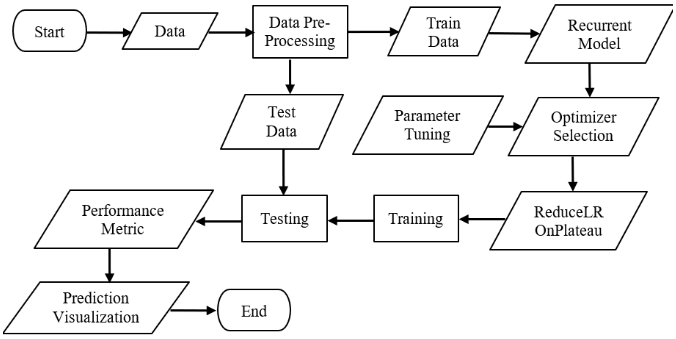

Fig. 1. Flowchart Design of Airport Traffic Prediction

2.1. Data Collection and Exploration

In this study, we used a dataset covering the period from March 16, 2020, to December 12, 2020, which includes data specifically from the early and mid-stages of the COVID-19 pandemic. The dataset consists of a single numerical attribute, "Percent of Baseline," which represents the daily percentage of airport traffic relative to a pre-pandemic baseline. In total, the dataset contains 7,247 data points, with each data point representing daily aggregated data from various airports across the USA, Canada, Chile, and Australia, as shown in Table 1.

Table 1. Airport Name and City

Airport Name	City, Country
Boston Logan International	Boston, USA
Calgary International	Calgary, Canada
Charlotte Douglas International	Charlotte, USA
Chicago O'Hare International	Chicago, USA
Dallas/Fort Worth International	Grapevine, USA
Daniel K. Inouye International	Honolulu, USA
Denver International	Denver, USA
Detroit Metropolitan Wayne County	Romulus, USA
Edmonton International	Leduc County, Canada
Halifax International	Halifax, Canada
Hamilton International	Hamilton, Canada
Hartsfield-Jackson Atlanta International	College Park, USA
John F. Kennedy International	New York, USA
Kingsford Smith	Sydney, Australia
LaGuardia	New York, USA
Los Angeles International	Los Angeles, USA
McCarran International	Paradise, USA
Miami International	Miami Springs, USA
Montreal Mirabel	Mirabel, Canada
Montreal Trudeau	Quebec, Canada
Newark Liberty International	Newark, USA
San Francisco International	South San Francisco, USA
Santiago International Airport	Santiago, Chile
Seattle-Tacoma International	SeaTac, USA
Toronto Pearson	Mississauga, Canada
Vancouver International	Richmond, Canada
Washington Dulles International	Floris, USA
Winnipeg International	Winnipeg, Canada

2.2. Data Pre-Processing

To ensure the accuracy and relevance of our analysis, we implemented several crucial preprocessing steps: filtering, cleaning, and normalization using the MinMaxScaler.

We began with a filtering procedure to remove extraneous data, allowing us to focus on key parameters such as 'Date', 'AirportName', 'PercentOfBaseline', 'City', 'State', and 'Country'. This step is necessary because it eliminates irrelevant or redundant information that could introduce noise or bias into the analysis [46]. By narrowing down the dataset to only the essential attributes, we improve the clarity and focus of our model, ensuring that it processes only the most pertinent data points.

Data cleaning is critical for addressing any inconsistencies, missing values, or errors within the dataset [47]. Cleaning ensures that the data is accurate, complete, and free from anomalies that could otherwise distort the analysis. This step improves data quality by removing or correcting any corrupt data entries, thus providing a solid foundation for reliable modeling and analysis.

After filtering and cleaning, we employed the MinMaxScaler to normalize the 'PercentOfBaseline' values. The MinMaxScaler standardizes the data by scaling and translating each feature individually to a specified range, typically between 0 and 1 [48]. This normalization is crucial for maintaining inter-feature relationships, preventing any single feature from dominating the model due to its scale. By ensuring that all features are on a comparable scale, the MinMaxScaler enhances the model's ability to learn from the data effectively and improves the overall consistency and standardization across the dataset. This process is mathematically represented by (1), where the MinMaxScaler adjusts values accordingly.

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

2.3. Gated Recurrent Unit (GRU)

In this study, we utilized the GRU model because it effectively addresses the vanishing gradient problem commonly found in standard RNNs by combining the cell state and hidden state [38]. GRUs are particularly well-suited for time series prediction due to their simplified architecture, consisting of only two gating mechanisms: the update gate and the reset gate. These gates are crucial for solving long-interval and long-delay time series prediction problems [39]. The update gate controls how much information from the previous time step is carried forward to the current step, ensuring that relevant data is retained over long sequences [40]. The reset gate, on the other hand, controls how much of the previous information is forgotten, allowing the model to reset its memory when necessary. This functionality is illustrated in Fig. 2.

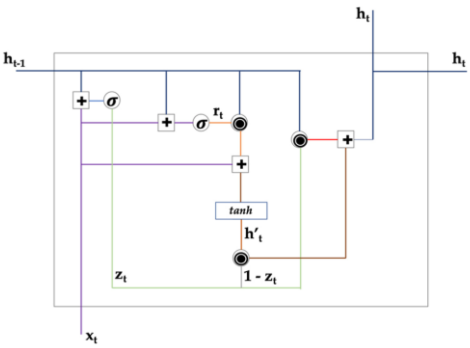

Fig. 2. GRU Architecture [41]

In (1), the variables x_t and h_t represent the current input and the output at step t, while r_t and z_t denote the reset and update gates. These two gates are key structures of the GRU, each being a simple neural network. The candidate activation for the output $\tilde{h}_t$ is h_t. Intuitively, the reset gate r_t and the update gate z_t measure the correlation between the previous state information and the prediction for the next step.

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t &= \tanh(W \cdot [r_t \cdot h_{t-1}, x_t]) \\ h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot h_t \end{aligned} \tag{2}$$

2.4. Convolutional Neural Network (CNN)

In this study, we also employed a Convolutional Neural Network (CNN) model. A CNN is essentially a neural network that uses convolution operations, instead of fully connected layers, as one of its layers [42]. CNNs are highly successful technologies, particularly well-suited for problems where the input data has a grid-like topology, such as time series (1-D grid) or images (2-D grid) [43]. These networks consist of an input layer, an output layer, and multiple hidden layers. The hidden layers typically include convolutional layers, pooling layers, fully connected layers, and various normalization layers, which together enable the model to automatically and adaptively learn spatial hierarchies of features [44], illustrated in Fig. 3.

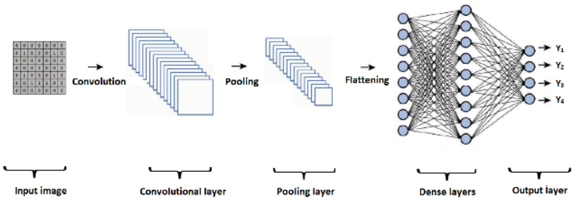

Fig. 3. CNN Structure [45]

2.5. Train/Test Split

In addressing the challenge posed by the imbalanced distribution of airport data across different countries, we adopted a methodological approach aimed at enhancing data representativeness. Specifically, we computed a daily average airport baseline for each country using the Pandas dataframe.groupby().mean() function. This function allowed us to group the dataset by country and calculate the daily mean of airport data. By integrating these baseline data points, we aimed to create a more balanced dataset that better reflects the overall airport activity within each country.

The decision to average the data was driven by the need to mitigate the dominance of data from countries with a higher volume of airport records, such as the USA, which could skew the overall analysis. By averaging the baseline data, we ensured that each country contributed equally to the model, thereby reducing the risk of bias and improving the fairness of comparisons across countries.

However, this approach has implications for model performance and generalizability. Averaging may smooth out extreme values and reduce variability, potentially leading to a loss of granular information that could be significant in some contexts. This could impact the model's ability to capture nuanced patterns within each country's data. On the other hand, this method enhances the generalizability of the model by preventing it from being overly influenced by data from a few countries with more records.

To further validate this approach, we partitioned the dataset into an 80% training and 20% testing split, applying the mean function to data from the USA, Canada, Chile, and Australia, as detailed in Table 2. This helped in assessing the effectiveness of the method in diverse regional contexts, providing insights into its impact on model robustness and predictive accuracy:

Table 2. Train/Test split result

Country	Train (80%)	Test (20%)	Total (100%)
USA	210	52	262
Chile	191	47	238
Canada	210	52	262
Australia	206	51	257

2.6. Recurrent Layer Model

This study employs two GRU layers, as detailed in Table 3. GRU Layer, which is responsible for capturing temporal dependenddcies in the data. The ReLU activation function is used to introduce non-linearity. The return_sequences setting ensures that the layer outputs the full sequence of data for the next layer. Dropout Layer randomly drops 20% of the neurons during training to prevent overfitting and improve the model's generalization ability. The Dense Layer consists of a single neuron, which generates the final prediction output, typically representing a single value in a regression task like time series forecasting.

In Table 4, In this CNN-GRU architecture, the layers work together to extract features and learn temporal dependencies from the input data. The Conv1D Layer performs convolution operations along the time axis to capture local patterns in the input data. It uses filters to extract features from the input sequences while maintaining the original sequence length due to 'same' padding. The GRU Layer processes the sequential data to capture temporal dependencies and patterns. It returns the full sequence of outputs, which is necessary for further processing by subsequent layers. The Dropout layer randomly drops 20% of the neurons during training to prevent overfitting and improve generalization. A Dense Layer with a single unit produces the final output, typically for tasks like time series forecasting.

Table 3. GRU Parameter Model

Parameter	Value
GRU unit	64
GRU activation	ReLu
GRU return_sequences	True
Dropout	0.2
GRU unit	32
GRU activation	ReLu
GRU return_sequences	False
Dense unit	1

Table 4. CNN-GRU Parameter Model

Parameter	Value
Conv1d filters	64
Conv1d kernel_size	3
Conv1d strides	1
Conv1d activation function	ReLu
Conv1d padding	same
GRU unit	64
GRU activation	ReLu
GRU return_sequences	True
Dropout	0.2
Conv1d filters	32
Conv1d kernel_size	3
Conv1d strides	1
Conv1d activation function	ReLu
Conv1d padding	same
GRU unit	32
GRU activation	ReLu
GRU return_sequences	False
Dense unit	1

2.7. Optimizer Selection

We chose the specific optimizers RMSProp, Adam, Nadam, AdamW, Adamax, and Lion due to their proven effectiveness in handling various optimization challenges. RMSProp is selected for its ability to adapt the learning rate based on recent gradient magnitudes, making it effective for non-stationary problems [49]. Adam combines the advantages of RMSProp and momentum, offering robust performance across a wide range of deep-learning tasks [49]. Nadam enhances Adam by incorporating Nesterov momentum, which often leads to faster convergence [50]. AdamW is an improved version of Adam that decouples weight decay from the gradient update, resulting in better generalization [51]. Adamax, a variant of Adam, uses the infinity norm and is well-suited for models with large parameter spaces [52]. Lastly, Lion is a newer optimizer designed to handle large-scale and complex datasets efficiently, offering a balance between convergence speed and computational efficiency [53]. These optimizers were selected to ensure a comprehensive evaluation of their impact on the GRU and CNN-GRU models' predictive capabilities.

2.8. Performance Metric

To measure and compare the performance of each model, we used Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) were chosen because they provide clear and interpretable measures of prediction accuracy, with MAE offering a straightforward average error metric and MAPE allowing for easy comparison across different datasets by expressing errors as percentages [54]. Both metrics are computed using the following formulas:

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

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

3. RESULTS AND DISCUSSION

In this study, the choice of 60 epochs ensures sufficient training iterations for the models to converge without overfitting. A batch size of 32 balances computational efficiency and model performance. The `ReduceLROnPlateau` with `patience=3` and `factor=0.2` was chosen to dynamically reduce the learning rate when the validation loss stops improving, helping to fine-tune the model during training. The `min_delta=0.00001` ensures that only significant improvements are considered, and the `min_lr=0.00000001` prevents the learning rate from becoming too small to make further progress. Fig. 4 provides a visual representation of the GRU model structure for the USA. In the CNN-GRU model, we incorporated Conv1D layers into the previous GRU model. Fig. 5 provides a visual representation of the CNN-GRU model structure for the USA.

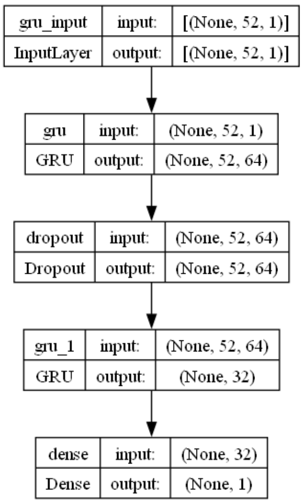

Fig. 4. GRU Model Structure

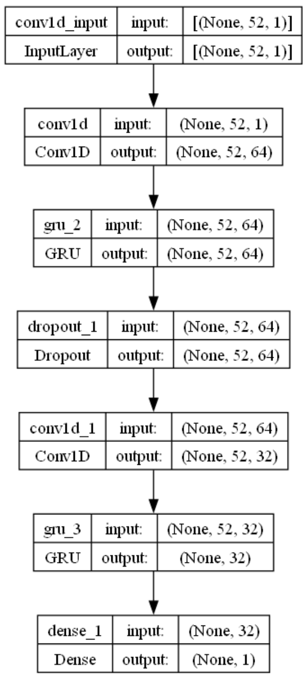

Fig. 5. CNN-GRU Model Structure

Table 5 provides a comprehensive summary of the performance metric scores for several countries, including the USA, Canada, Chile, and Australia. In our analysis, first, we utilized various optimization techniques, such as RMSProp, Adam, Nadam, AdamW, Adamax, and Lion on default parameters to train GRU and CNN-GRU models. These models were trained using Visual Studio Code with a Python 3 runtime. By comparing the scores obtained with these optimization techniques, we evaluate the effectiveness of the models in predicting airport baseline percentages for each country.

Tabel 5. Default Optimizer Performance

Country	Model	Optimizer	MAE	MAPE
USA	GRU	RMSprop	0.0706	0.1182
		Lion	0.0700	0.1187
		Adam	0.0683	0.1163
		Nadam	0.0686	0.1165
		AdamW	0.0729	0.1201
		Adamax	0.0686	0.1168
	CNN-GRU	RMSprop	0.0697	0.1184
		Lion	0.0655	0.1145
		Adam	0.0677	0.1163
		Nadam	0.0697	0.1184
		AdamW	0.0667	0.1152
		Adamax	0.0658	0.1141
Australia	GRU	RMSprop	0.0721	0.2874
		Lion	0.0715	0.2875
		Adam	0.0722	0.2985
		Nadam	0.0727	0.2864
		AdamW	0.0721	0.2865
		Adamax	0.0715	0.2930
	CNN-GRU	RMSprop	0.0724	0.2938
		Lion	0.0713	0.2917
		Adam	0.0702	0.2885
		Nadam	0.0708	0.2899
		AdamW	0.0711	0.2906
		Adamax	0.0723	0.2938
Chile	GRU	RMSprop	0.0877	0.3514
		Lion	0.0948	0.3897
		Adam	0.0858	0.3430
		Nadam	0.0871	0.3527
		AdamW	0.0914	0.3624
		Adamax	0.0842	0.3420
	CNN-GRU	RMSprop	0.0852	0.3380
		Lion	0.0913	0.3834
		Adam	0.0865	0.3599
		Nadam	0.0860	0.3396
		AdamW	0.0890	0.3602
		Adamax	0.0856	0.3416
Canada	GRU	RMSprop	0.1097	0.1633
		Lion	0.1104	0.1643
		Adam	0.1041	0.1591
		Nadam	0.1158	0.1675
		AdamW	0.1020	0.1566
		Adamax	0.1028	0.1566
	CNN-GRU	RMSprop	0.1124	0.1656
		Lion	0.1080	0.1632
		Adam	0.1087	0.1607
		Nadam	0.1088	0.1626
		AdamW	0.1107	0.1624
		Adamax	0.1039	0.1579

In Table 6, we applied parameter tuning to various optimization techniques to enhance model performance. For the RMSProp optimizer, we used specific hyperparameters, including rho=0.0001, which controls the moving average of squared gradients, weight_decay=0.0001 to help prevent overfitting by adding a regularization term to the loss function, and enabled exponential moving average (EMA) with ema_momentum=0.0001, 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=0.0001, controlling the exponential decay rate for the second-moment estimates (i.e., the variance of gradients). EMA was also utilized with ema_momentum=0.0001, contributing to more stable training. These hyperparameters were selected and validated using grid search, which systematically explored different combinations to identify the settings that yielded the best predictive performance for the GRU and CNN-GRU models. This iterative process involved testing and validating various combinations with the validation set to find the most effective configurations. The results from this tuning process are reflected in the model performance metrics, highlighting the importance of careful optimizer selection and parameter tuning in achieving optimal results.

Tabel 6. Optimizer Performance with Variable Tuning

Country	Model	Optimizer	MAE	MAPE
USA	GRU	RMSprop	0.0694	0.1162
		Lion	0.0689	0.1169
		Adam	0.0663	0.1130
		Nadam	0.0670	0.1137
		AdamW	0.0710	0.1169
		Adamax	0.0672	0.1144
	CNN-GRU	RMSprop	0.0678	0.1151
		Lion	0.0644	0.1120
		Adam	0.0661	0.1135
		Nadam	0.0682	0.1159
		AdamW	0.0650	0.1123
		Adamax	0.0646	0.1126
Australia	GRU	RMSprop	0.0705	0.2811
		Lion	0.0697	0.2796
		Adam	0.0701	0.2858
		Nadam	0.0708	0.2899
		AdamW	0.0710	0.2822
		Adamax	0.0695	0.2790
	CNN-GRU	RMSprop	0.0709	0.2876
		Lion	0.0696	0.2848
		Adam	0.0690	0.2824
		Nadam	0.0691	0.2828
		AdamW	0.0698	0.2854
		Adamax	0.0708	0.2876
Chile	GRU	RMSprop	0.0862	0.3453
		Lion	0.0926	0.3805
		Adam	0.0840	0.3360
		Nadam	0.0846	0.3426
		AdamW	0.0896	0.3553
		Adamax	0.0824	0.3349
	CNN-GRU	RMSprop	0.0831	0.3316
		Lion	0.0894	0.3755
		Adam	0.0850	0.3538
		Nadam	0.0845	0.3338
		AdamW	0.0865	0.3502
		Adamax	0.0836	0.3317
Canada	GRU	RMSprop	0.1066	0.1587
		Lion	0.1084	0.1614
		Adam	0.1020	0.1558
		Nadam	0.1136	0.1643
		AdamW	0.0993	0.1525
		Adamax	0.1011	0.1539
	CNN-GRU	RMSprop	0.1106	0.1630
		Lion	0.1061	0.1603
		Adam	0.1067	0.1577
		Nadam	0.1068	0.1596
		AdamW	0.1084	0.1590
		Adamax	0.1008	0.1532

Table 6 presents a comparison of prediction performance using various optimizers for the GRU and CNN-GRU models across several countries. The results suggest that different optimizers affect model performance depending on the country and model architecture. The Adam optimizer's good performance with the GRU model in the USA might be due to its ability to adapt learning rates effectively in diverse data. The Lion optimizer's success with the CNN-GRU model could be due to its enhanced convergence properties with complex architectures. In Australia, the Adamax optimizer's stability helped the GRU model, while the Adam optimizer's general adaptability benefited the CNN-GRU model. In Chile, Adamax's blend of Adam and infinity norm properties suited the GRU model, whereas RMSprop's adaptive learning rate worked well for the

CNN-GRU model. In Canada, AdamW's weight decay regularization improved the GRU model's performance, while the Adamax optimizer effectively balanced learning rate adjustments for the CNN-GRU model.

To provide a visual representation of these results, Fig. 6 – Fig. 9 present the prediction outcomes of the best optimizers with provided parameter tuning for the GRU and CNN-GRU models in estimating the baseline airport percentage. The red line represents the actual baseline airport percentage data, while the blue line depicts the predictions generated by the GRU model with the best optimization. The green line illustrates the predictions made by the CNN-GRU model with the best optimization. By examining these graphical representations, we can evaluate the accuracy and effectiveness of the selected optimizers in predicting the baseline airport percentage.

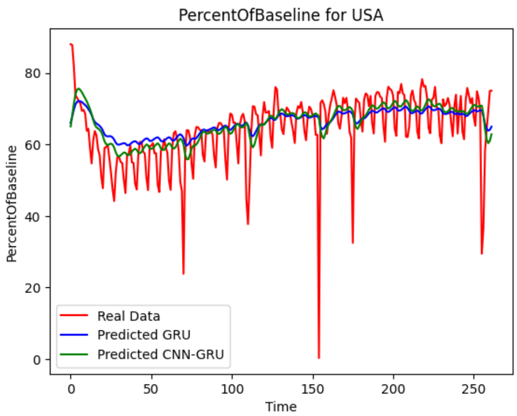

Fig. 6. Prediction Accuracy with Parameter Tuning for USA

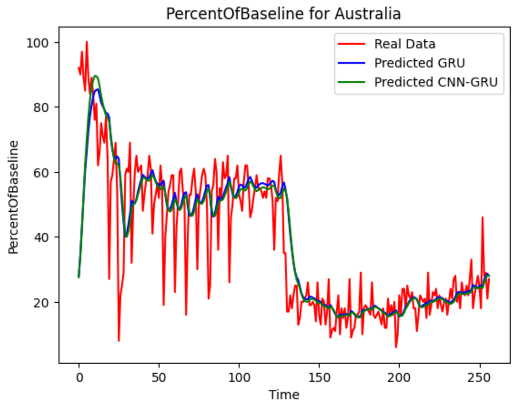

Fig. 7. Prediction Accuracy with Parameter Tuning for Australia

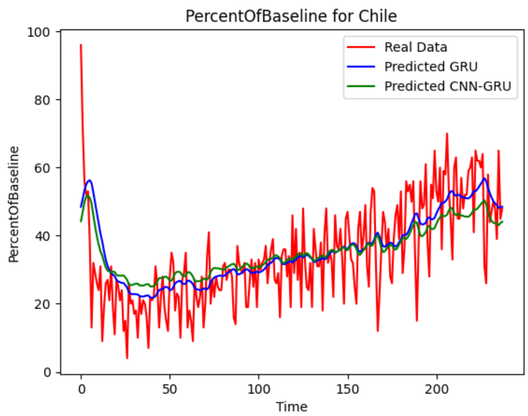

Fig. 8. Prediction Accuracy with Parameter Tuning for Chile

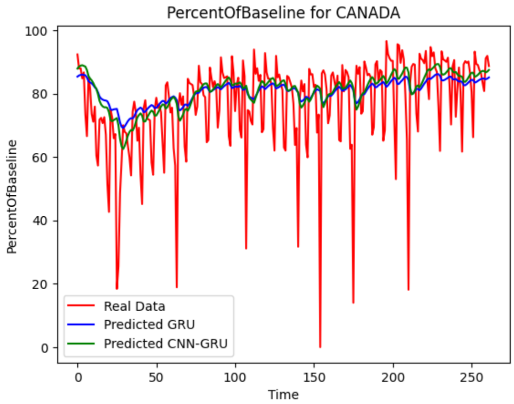

Fig. 9. Prediction Accuracy with Parameter Tuning for Canada

4. CONCLUSION

This study highlights the critical role of selecting and fine-tuning optimizers, alongside the use of ReduceLROnPlateau for dynamically adjusting the learning rate, in preventing overfitting and enhancing model convergence. However, limitations such as dataset imbalance and region-specific outcomes may affect the generalizability of these findings. Prediction results using GRU and CNN-GRU models on airport baseline data varied significantly across different countries, primarily due to the imbalanced dataset, which spanned approximately 10 months and averaged 1.019 data points per country. This led to noticeable performance variations in the USA, Canada, Chile, and Australia.

The optimal optimizers varied by region, emphasizing the need for careful selection. In the USA, the best performance for the GRU model was achieved with the Adam optimizer, resulting in an MAE of 0.0663 and a MAPE of 0.1130. For the CNN-GRU model, the Lion optimizer performed best with an MAE of 0.0644 and a MAPE of 0.1120. In Australia, the Adamax optimizer showed the best results for the GRU model, with an MAE of 0.0695 and a MAPE of 0.2790, while the Adam optimizer was most effective for the CNN-GRU model, yielding an MAE of 0.0690 and a MAPE of 0.2824. In Chile, the Adamax optimizer led to the best performance for the GRU model, with an MAE of 0.0824 and a MAPE of 0.3349, whereas the RMSprop optimizer was optimal for the CNN-GRU model, with an MAE of 0.0831 and a MAPE of 0.3316. In Canada, the AdamW optimizer provided the best results for the GRU model, achieving an MAE of 0.0993 and a MAPE of 0.1525, while the Adamax optimizer was most effective for the CNN-GRU model, with an MAE of 0.1008 and a MAPE of 0.1532.

These variations underscore the significant impact of dataset imbalance on prediction accuracy, reinforcing the need for balanced datasets in future research. Future studies should prioritize the creation of balanced datasets tailored to each region to mitigate the effects of data imbalance and improve prediction outcomes. Additionally, further investigation into optimizer parameter tuning is recommended, as it has been shown to significantly enhance prediction accuracy compared to default settings.

REFERENCES

[1] D. Dimitriou and A. Karagkouni, "Assortment of Airports' Sustainability Strategy: A Comprehensiveness Analysis Framework," *Sustainability*, vol. 14, no. 7, p. 4217, Apr. 2022, https://doi.org/ 10.3390/su14074217.

[2] A. Dixit and S. K. Jakhar, "Airport capacity management: A review and bibliometric analysis," *J Air Transp Manag*, vol. 91, p. 102010, Mar. 2021, https://doi.org/ 10.1016/j.jairtraman.2020.102010.

[3] 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, https://doi.org/ 10.1155/2020/6309272.

[4] S. Sreenath, K. Sudhakar, and A. Yusop, "Sustainability at airports: Technologies and best practices from ASEAN countries," *J Environ Manage*, vol. 299, p. 113639, Dec. 2021, https://doi.org/ 10.1016/j.jenvman.2021.113639.

[5] 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, https://doi.org/10.1016/j.jairtraman.2020.102007.

[6] H. Nakamura and S. Managi, "Airport risk of importation and exportation of the COVID-19 pandemic," *Transp Policy (Oxf)*, vol. 96, pp. 40–47, Sep. 2020, https://doi.org/ 10.1016/j.tranpol.2020.06.018.

[7] A. Barczak, I. Dembińska, D. Rozmus, and K. Szopik-Depczyńska, "The Impact of COVID-19 Pandemic on Air Transport Passenger Markets-Implications for Selected EU Airports Based on Time Series Models Analysis," *Sustainability*, vol. 14, no. 7, p. 4345, Apr. 2022, https://doi.org/ 10.3390/su14074345.

[8] 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, https://doi.org/10.1016/j.physa.2019.121790.
[9] A. Kanavos, F. Kounelis, L. Iliadis, and C. Makris, "Deep learning models for forecasting aviation demand time series," *Neural Comput Appl*, vol. 33, no. 23, pp. 16329–16343, Dec. 2021, https://doi.org/ 10.1007/s00521-021-06232-y.
[10] 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, https://doi.org/10.1016/j.physa.2019.123585.
[11] Monika, S. Verma, and P. Kumar, "Generic Deep-Learning-Based Time Series Models for Aviation Accident Analysis and Forecasting," *SN Comput Sci*, vol. 5, no. 1, p. 32, Nov. 2023, https://doi.org/ 10.1007/s42979-023-02353-4.
[12] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021, https://doi.org/ 10.1109/ACCESS.2021.3107975.
[13] Z. Wang and W.-K. Song, "Sustainable airport development with performance evaluation forecasts: A case study of 12 Asian airports," *J Air Transp Manag*, vol. 89, p. 101925, Oct. 2020, https://doi.org/ 10.1016/j.jairtraman.2020.101925.
[14] H. Tang *et al.*, "Airport terminal passenger forecast under the impact of COVID-19 outbreaks: A case study from China," *Journal of Building Engineering*, vol. 65, p. 105740, Apr. 2023, https://doi.org/ 10.1016/j.jobe.2022.105740.
[15] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study," *Chaos Solitons Fractals*, vol. 140, p. 110121, Nov. 2020, https://doi.org/ 10.1016/j.chaos.2020.110121.
[16] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study," *Chaos Solitons Fractals*, vol. 140, p. 110227, Nov. 2020, https://doi.org/ 10.1016/j.chaos.2020.110227.
[17] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*, 2nd ed. New Jersey: Wiley, 2015, https://books.google.co.id/books?id=Xeh8CAAAQBAJ.
[18] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction," *IEEE Access*, vol. 9, pp. 83105–83123, 2021, https://doi.org/ 10.1109/ACCESS.2021.3085085.
[19] Z. Shen, Y. Zhang, J. Lu, J. Xu, and G. Xiao, "A novel time series forecasting model with deep learning," *Neurocomputing*, vol. 396, pp. 302–313, Jul. 2020, https://doi.org/ 10.1016/j.neucom.2018.12.084.
[20] 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, https://doi.org/10.1016/j.jairtraman.2020.101946.
[21] G. Gui, Z. Zhou, J. Wang, F. Liu, and J. Sun, "Machine Learning Aided Air Traffic Flow Analysis Based on Aviation Big Data," *IEEE Trans Veh Technol*, vol. 69, no. 5, pp. 4817–4826, May 2020, https://doi.org/ 10.1109/TVT.2020.2981959.
[22] S. Yassine and A. Stanulov, "A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR THE PURPOSE OF PREDICTING NORWEGIAN AIR PASSENGER TRAFFIC," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 28–43, Jul. 2023, https://doi.org/ 10.59543/ijmscs.v2i.7851.
[23] K. Cai, Y. Li, Y.-P. Fang, and Y. Zhu, "A Deep Learning Approach for Flight Delay Prediction Through Time-Evolving Graphs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11397–11407, Aug. 2022, https://doi.org/ 10.1109/TITS.2021.3103502.
[24] 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.
[25] W. Shao, A. Prabowo, S. Zhao, P. Koniusz, and F. D. Salim, "Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map," *Neurocomputing*, vol. 472, pp. 280–293, Feb. 2022, https://doi.org/ 10.1016/j.neucom.2021.04.136.
[26] Z. Yan, H. Yang, F. Li, and Y. Lin, "A Deep Learning Approach for Short-Term Airport Traffic Flow Prediction," *Aerospace*, vol. 9, no. 1, 2022, https://doi.org/ 10.3390/aerospace9010011.
[27] S. Choi and Y. J. Kim, "Artificial neural network models for airport capacity prediction," *J Air Transp Manag*, vol. 97, p. 102146, Oct. 2021, https://doi.org/ 10.1016/j.jairtraman.2021.102146.
[28] L. Yuan, J. Liu, H. Chen, D. Fang, and W. Chen, "A CNN-GRU Hybrid Model for Predicting Airport Departure Taxiing Time," *Aerospace*, vol. 11, no. 4, p. 261, Mar. 2024, https://doi.org/ 10.3390/aerospace11040261.
[29] J. Yu, "Short-term Airline Passenger Flow Prediction Based on the Attention Mechanism and Gated Recurrent Unit Model," *Cognit Comput*, vol. 14, no. 2, pp. 693–701, Mar. 2022, https://doi.org/ 10.1007/s12559-021-09991-x.
[30] H. Shafienya and A. C. Regan, "4D flight trajectory prediction using a hybrid Deep Learning prediction method based on ADS-B technology: A case study of Hartsfield–Jackson Atlanta International Airport (ATL)," *Transp Res Part C Emerg Technol*, vol. 144, p. 103878, Nov. 2022, https://doi.org/ 10.1016/j.trc.2022.103878.
[31] L. Yuan, J. Liu, H. Chen, D. Fang, and W. Chen, "A CNN-GRU Hybrid Model for Predicting Airport Departure Taxiing Time," *Aerospace*, vol. 11, no. 4, p. 261, Mar. 2024, https://doi.org/ 10.3390/aerospace11040261.
[32] S. Pavitpok, P. Phasukkit, and C. Pradabpet, "Vertical Speed Prediction for the Efficient Landing of Aircraft Using GRU," in *2020 5th International STEM Education Conference (iSTEM-Ed)*, 2020, pp. 47–50. https://doi.org/ 10.1109/iSTEM-Ed50324.2020.9332739.

[33] 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, https://doi.org/ 10.1155/2022/2086904.
[34] 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, https://doi.org/ 10.1016/j.eswa.2023.121423.
[35] B. Hussain, M. K. Afzal, S. Ahmad, and A. M. Mostafa, "Intelligent Traffic Flow Prediction Using Optimized GRU Model," *IEEE Access*, vol. 9, pp. 100736–100746, 2021, https://doi.org/ 10.1109/ACCESS.2021.3097141.
[36] L. Ji, C. Fu, Z. Ju, Y. Shi, S. Wu, and L. Tao, "Short-Term Canyon Wind Speed Prediction Based on CNN—GRU Transfer Learning," *Atmosphere (Basel)*, vol. 13, no. 5, p. 813, May 2022, https://doi.org/ 10.3390/atmos13050813.
[37] 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.
[38] M.-C. Chiu, H.-W. Hsu, K.-S. Chen, and C.-Y. Wen, "A hybrid CNN-GRU based probabilistic model for load forecasting from individual household to commercial building," *Energy Reports*, vol. 9, pp. 94–105, Oct. 2023, https://doi.org/ 10.1016/j.egyr.2023.05.090.
[39] W. Li, H. Wu, N. Zhu, Y. Jiang, J. Tan, and Y. Guo, "Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU)," *Information Processing in Agriculture*, vol. 8, no. 1, pp. 185–193, Mar. 2021, https://doi.org/ 10.1016/j.inpa.2020.02.002.
[40] J. Wang, P. Wang, H. Tian, K. Tansey, J. Liu, and W. Quan, "A deep learning framework combining CNN and GRU for improving wheat yield estimates using time series remotely sensed multi-variables," *Comput Electron Agric*, vol. 206, p. 107705, Mar. 2023, https://doi.org/ 10.1016/j.compag.2023.107705.
[41] A. Dutta, S. Kumar, and M. Basu, "A Gated Recurrent Unit Approach to Bitcoin Price Prediction," *Journal of Risk and Financial Management*, vol. 13, no. 2, p. 23, Feb. 2020, https://doi.org/ 10.3390/jrfm13020023.
[42] M. Pan *et al.*, "Water Level Prediction Model Based on GRU and CNN," *IEEE Access*, vol. 8, pp. 60090–60100, 2020, https://doi.org/ 10.1109/ACCESS.2020.2982433.
[43] N. Ketkar and J. Moolayil, *Deep Learning with Python*. Berkeley, CA: Apress, 2021. https://doi.org/ 10.1007/978-1-4842-5364-9.
[44] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, Jun. 2020, https://doi.org/ 10.1007/s13748-019-00203-0.
[45] V. Maeda-Gutiérrez *et al.*, "Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases," *Applied Sciences*, vol. 10, no. 4, p. 1245, Feb. 2020, https://doi.org/ 10.3390/app10041245.
[46] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, https://doi.org/ 10.1016/j.gltp.2022.04.020.
[47] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "Big Data Preprocessing," Cham: Springer International Publishing, 2020. https://doi.org/ 10.1007/978-3-030-39105-8.
[48] 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, https://doi.org/ 10.3390/fi12110194.
[49] V.-H. Nhu *et al.*, "Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area," *Catena (Amst)*, vol. 188, p. 104458, May 2020, https://doi.org/ 10.1016/j.catena.2020.104458.
[50] Q. Zhang *et al.*, "Boosting Adversarial Attacks with Nadam Optimizer," *Electronics (Basel)*, vol. 12, no. 6, p. 1464, Mar. 2023, https://doi.org/ 10.3390/electronics12061464.
[51] R. Castro, I. Pineda, and M. E. Morocho-Cayamcela, "Hyperparameter Tuning over an Attention Model for Image Captioning," 2021, pp. 172–183. https://doi.org/ 10.1007/978-3-030-89941-7_13.
[52] C. Arora, G. Raj, A. Ajit, and A. Saxena, "ADAMAX-Based Optimization of Efficient Net V2 for NSFW Content Detection," in *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, IEEE, Apr. 2023, pp. 1–6. https://doi.org/ 10.1109/InC457730.2023.10263203.
[53] B. Kishiyama, Y. Lee, and J. Yang, "Improving VulRepair's Perfect Prediction by Leveraging the LION Optimizer," *Applied Sciences*, vol. 14, no. 13, p. 5750, Jul. 2024, https://doi.org/ 10.3390/app14135750.
[54] P. Cinaglia and M. Cannataro, "Forecasting COVID-19 Epidemic Trends by Combining a Neural Network with Rt Estimation," *Entropy*, vol. 24, no. 7, p. 929, Jul. 2022, https://doi.org/ 10.3390/e24070929.

BIOGRAPHY OF AUTHORS

Willy Riyadi is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Computer Engineering in STIKOM Dinamika Bangsa Jambi in 2011 and Master Degree in Information Systems in Universitas Dinamika Bangsa Jambi in 2014. His research interests are Deep Learning and Information Systems. He can be contacted at email: wriyadi5@gmail.com. ORCID: https://orcid.org/0000-0002-7091-1876.

Jasmir is a senior lecturer at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Computer Engineering in 1995 and a Master's degree in Information Technology in 2006 from Universitas Putra Indonesia YPTK Padang, Indonesia. He receives a Doctorate in Informatics Engineering at Universitas Sriwijaya Palembang, Indonesia in 2022. His research interest is data mining, machine learning, and deep learning for natural language processing and its application. He can be contacted at email: ijay_jasmir@yahoo.com. ORCID: https://orcid.org/0000-0003-0059-2512.