Optimizing the GRU-LSTM Hybrid Model for Air Temperature Prediction in Degraded Wetlands and Climate Change Implications

Yuslena Sari[1], Yudi Firmanul Arifin[2], Novitasari Novitasari[3], Samingun Handoyo[4], Andreyan Rizky Baskara[5], Nurul Fathanah Musatamin[6], Muhammad Tommy Maulidyanto[7], Siti Viona Indah Swari[8], Erika Maulidiya[9]
Department of Information Technology, Universitas Lambung Mangkurat, Banjarmasin, Indonesia[1, 5, 6, 8, 9]
Faculty of Forestry, Universitas Lambung Mangkurat, Banjarmasin, Indonesia[2]
Faculty of Engineering, Universitas Lambung Mangkurat, Banjarmasin, Indonesia[3]
Data Science Study Program, Brawijaya University, Malang, Indonesia[4]
Department of Mining Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia[7]

***Abstract*—Accurate air temperature prediction is critical, particularly for micro air temperatures. The temperature of micro air changes quickly. Micro and macro air temperatures vary, particularly in degraded wetlands. By predicting air temperature, climate change in a degraded wetland environment can be predicted earlier. Furthermore, micro and macro air temperatures are drought index parameters. Knowing the drought index can help you avoid disasters like fires and floods. However, the right indicators for predicting micro or macro temperatures have yet to be found. LSTM excels at tasks requiring complex long-term memory, whereas GRU excels at tasks requiring rapid processing. We proposed a deep learning strategy based on the GRU-LSTM Hybrid model. Both of these deep learning models are excellent for predicting time series. The performance of this hybrid model is affected by changes in model indicators. The preprocessing stage, the number of input parameters, and the presence or absence of a Dropout Layer in the model architecture are among the most influential indicators of model performance. The best macro temperature prediction performance was obtained using 12 monthly average data to predict the next month's temperature, yielding an RMSE of 0.056807, MAE of 0.046592, and R2 of 0.989371. This model also performed well in predicting daily micro temperature, with an RMSE of 0.227086, MAE of 0.190801, and R2 of 0.981802.**

Keywords*—Predictions; temperature; Gated Recurrent Unit (GRU); Long Short-Term Memory (LSTM); performance; indicators

I. Introduction

The wetland ecosystem and biological processes are influenced by air temperature. Temperature knowledge can provide insight into environmental changes and aid in effectively managing and conserving wetlands [1], [2]. Prediction of air temperature is critical in identifying and anticipating potential wetlands disasters. Temperature extremes can cause changes in rain patterns, affecting water discharge and potential flooding. Higher temperatures can cause more evaporation and more intense rain in a shorter time, potentially causing flooding in wetlands [3]-[5]. Extreme or unusual temperatures can harm wetland ecosystems. Overheating, for example, can cause a drop in water levels and dry out wetlands, which can disrupt habitats and threaten the survival of species that live there. In wetlands, hot, dry weather can increase the risk of fire. If extreme temperatures continue, dry vegetation and the risk of wildfires will increase, endangering ecosystems and the surrounding environment [6]-[8]. Because of the numerous effects of temperature changes, it is necessary to forecast temperature to mitigate natural disasters that may occur in the wetland environment.

Various temperature prediction methods have been explored, including ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal ARIMA), LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Unit) models [9–13]. Many studies have employed time series data for predictions, often using Recurrent Neural Network (RNN) architectures. Improved RNN variants, such as GRU and LSTM, have been widely adopted due to their superior performance over traditional RNN models. Long input sequences in RNNs often lead to exploding and vanishing gradients, an issue LSTM effectively mitigates through its gating mechanism, which regulates information flow and retains long-term dependencies in sequential data [10], [14]. LSTM can extract historical patterns more efficiently, outperforming conventional time series forecasting methods. GRU, a simplified LSTM variant, has fewer parameters, making it computationally more efficient while maintaining comparable accuracy [15], [16]. The hybrid GRU-LSTM model was chosen because it leverages the strengths of both architectures: LSTM's ability to capture long-term dependencies and GRU's computational efficiency. The proposed approach balances predictive accuracy and processing time by combining these models, making it well-suited for real-time temperature forecasting in wetland ecosystems. This architecture was optimized by tuning hyperparameters such as learning rate, batch size, and the number of hidden units to minimize prediction errors. The GRU-LSTM model has demonstrated higher accuracy than individual GRU and LSTM models and lower RMSE values than other prediction methods [17], [18].

To address the limitations of conventional models and improve prediction accuracy, this study proposes a novel workflow for air temperature prediction in wetland ecosystems by integrating the GRU-LSTM model. The proposed workflow includes data collection, preprocessing, feature selection, model implementation, and performance evaluation using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R²) metrics. By combining the strengths of GRU and LSTM, this approach enhances the model's ability to capture complex temperature patterns while maintaining computational efficiency. This model is designed for practical implementation in environmental monitoring systems, enabling real-time temperature predictions that can assist policymakers and conservationists in making informed decisions. Additionally, the model's architecture is flexible and can be adapted to different geographical regions by retraining it with local temperature datasets. The GRU-LSTM model also demonstrates scalability, as it can be deployed on cloud-based platforms or edge computing devices, allowing for efficient processing of large-scale temperature data from multiple sensor networks. This model offers a viable solution for long-term wetland monitoring and disaster risk mitigation by balancing predictive accuracy and computational efficiency. Consequently, this study contributes to developing more accurate and efficient temperature prediction methods, supporting disaster risk mitigation and wetland conservation efforts.

II. Material

This study processed and analyzed data using a Lenovo Ideapad 330 15-ARR Laptop. The laptop has an AMD Ryzen 7 2700U processor, 8GB of RAM, a 1TB hard drive, and the Windows 10 operating system, providing sufficient data processing power. Table I presents the data sources used in this study.

TABLE I. SOURCE OF THE DATA

No.	Type of the Data	Source of the Data
1.	Data of Micro Air Temperature	Air temperature data from temperature sensors in the Liang Anggang Protected Forest Block 1
2.	Data of Macro Air Temperature	Daily Average Air Temperature Data from the Meteorology, Climatology and Geophysics Agency (BMKG) Online Data Syamsudin Noor Meteorological Station

TABLE II. DATASET

Date	Tavg
01-01-1996	25.60
01-02-1996	25.50
01-03-1996	26.50
01-04-1996	25.70
….	…
12-31-2021	27.20

This study used air temperature data from Liang Anggang Protected Forest Block 1, which included 42 records with 3-hour intervals and 42 daily average temperature values. The data was collected using IoT sensors and manual measurements to ensure comprehensive and reliable temperature monitoring. Additionally, macro-scale air temperature data was incorporated to provide a broader perspective on temperature variations. This dataset included daily average air temperature from the Meteorology, Climatology, and Geophysics Agency (BMKG) Online Data of the Syamsudin Noor Meteorological Station, covering 26 years (January 1, 1996 – December 31, 2021). Weekly and monthly average temperatures were also used to identify long-term trends and seasonal patterns. Table II presents an example of the collected air temperature data.

III. Proposed Method

The research flow is a series of steps required to conduct research systematically. Fig. 1 shows the research flow of this research.

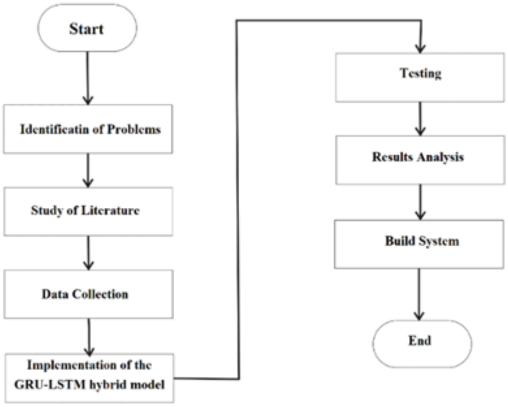


Fig. 1. Flow chart of research process.

The focus of this research shifted from data collection to testing to obtain model performance results that were used in the results analysis process, as shown in Fig. 2.

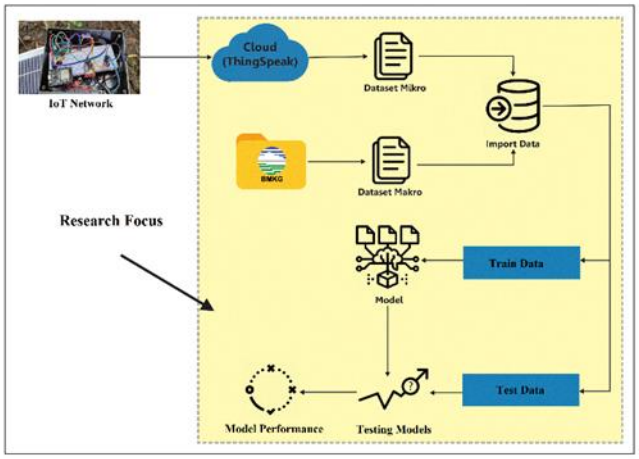


Fig. 2. Research focus.

A. Implementation of GRU-LSTM Hybrid Model

The GRU-LSTM hybrid model implementation process stages are shown in Fig. 3.

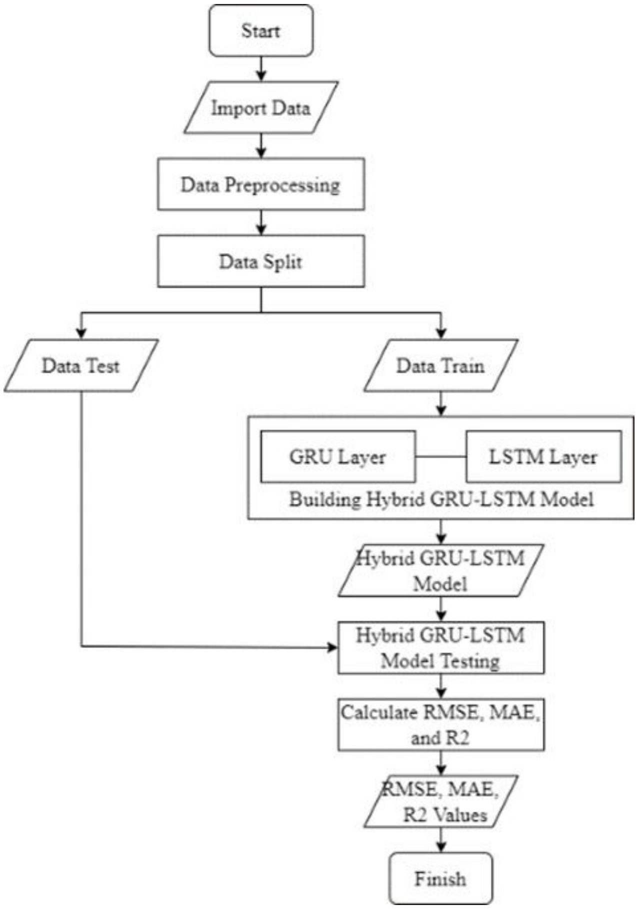


Fig. 3. Workflow model.

B. *Data Preprocessing*

The imported data was used as a data series during the data preprocessing stage, first by testing various input parameters and then using the Exponential (Weighted) Moving Average to fill in NaN values or empty values before being transformed using the Fast Fourier Transform.

1) Exponential (Weighted) Moving Average (EWMA): EWMA is a statistical method for smoothing datasets by decreasing data weight over time, and it is sensitive to process shifts [19]. The standard EWMA calculation is as follows:

$$\mathrm{S}_t = \lambda e_t + (1-\lambda)\mathrm{S}_{t-1} \tag{1}$$

Where S_t is the EWMA result, t is the time stamp, e_t is the average output, and λ is a constant ranging from 0 to 1 that controls the impact of historical data.

2) Fast Fourier Transform (FFT): Time series data is changed using FFT. FFT enhances the Discrete Fourier Transform (DFT) algorithm by leveraging periodicity and symmetry, significantly reducing the number of operations. FFT has the advantages of simplicity and speed. As a result, FFT is used to represent errors in the frequency domain to facilitate error identification [20], [21]. FFT can be calculated with [22]:

$$X_t(\mathrm{k}) = \sum_{t=1}^{\bar{N}} x_{ti} e^{\frac{-j2\pi}{\bar{N}}k(t-1)} \tag{2}$$

Where $X_i = [\, x_{1i}\, x_{2i} \cdots x_{\bar{N}i}]^T$, $\mathrm{i} = 1, 2, \cdots, \underline{\mathrm{y}}$, and $[\cdot]^T$ represents transpose from vector $[\cdot]$, $k = 0, 1, 2, \cdots, \bar{N}-1$.

C. *Data Sharing*

For benchmarking, the datasets must be divided into training and test data. The model is trained using training data and evaluated using test data. This study conducted tests to assess various types of data-sharing ratios.

D. *Model Establishing*

The GRU-LSTM hybrid model was built with the GRU and LSTM layers from library keras.

1) Gated Recurrent Unit (GRU): GRU is a promising algorithm within the Recurrent Neural Network (RNN) [18]. GRU and LSTM cells operate similarly, but GRU cells use a hidden state that combines the forget gate and the input gate to form an update gate. GRU also combines hidden and cell states into a single state [22]. As a result, GRU is known as a simplified variant of LSTM [15]. The GRU equation is as follows [23]:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{(t-1)} + b_z) \tag{3}$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{(t-1)} + b_r) \tag{4}$$

$$\tilde{h}_t = \tanh(W_h \cdot x_t + (r_t \circ U_h \cdot h_{(t-1)}) + b_h) \tag{5}$$

$$h_t = z_t \circ h_{(t-1)} + (1-z_t) \circ \tilde{h}_t \tag{6}$$

Where z_t = update gate, σ = activation of the sigmoid function, W_z = weight of update gate, U_z = weight of hidden state, x_t = input value (input vector x in timestep t), $h_{(t-1)}$= value of prior hidden state cell, b_z = bias of update gate, r_t = reset gate, W_r = weight of reset gate, U_r = weight of hidden state, b_r = bias of reset gate, $\tilde{h}_t$ = Output candidate from cell state vector, W_h = weight of cell state vector, U_h = weight of hidden state, b_h = bias of cell state vector, and h_t = cell state vector.

2) Long-Short Term Memory (LSTM): LSTM is a type of Recurrent Neural Network (RNN) that, when compared to conventional RNNs, allows the network to maintain long-term dependence between data at a specific time from many timesteps because LSTM uses special hidden blocks that remember input data for a long time [24], [25]. A typical LSTM unit consists of a memory cell, forget gate, input gate, and output gate, with the forget gate's purpose being to forget information in the cell state selectively, the input gate deciding what new information is stored in the cell state, and the output gate deciding what value to remove. The cell remembers the value over arbitrary time intervals, and three gates control the data flow into and out of the cell [26]. The LSTM application procedure is as follows [26]:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

$$\tilde{\mathrm{C}}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{9}$$

$$c_t = (f_t * c_{t-1} + i_t * \tilde{\mathrm{C}}_t) \tag{10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad (11)$$

$$h_t = o_t * \tanh(c_t) \qquad (12)$$

Where ft = forget gate, σ = sigmoid activation function, Wf = weight of forget gate, ht−1 = value of prior hidden state cell, xt = input value (input vector x in timestep t), bf = bias of forget gate, it = input gate, Wi = weight of input gate, bi = bias of input gate, C̃t = candidate gate, Tanh = tanh activation function, Wc = weight of candidate gate, bc = bias of candidate gate, ct = cell gate, it = input gate, C̃t = candidate gate, ft= forget gate, ct−1 = value of prior cell state, ot = output gate, Wo = weight of output gate, bo = bias of output gate, ht = hidden state, ct = cell gate. The LSTM architecture is shown in Fig. 4 [15]:

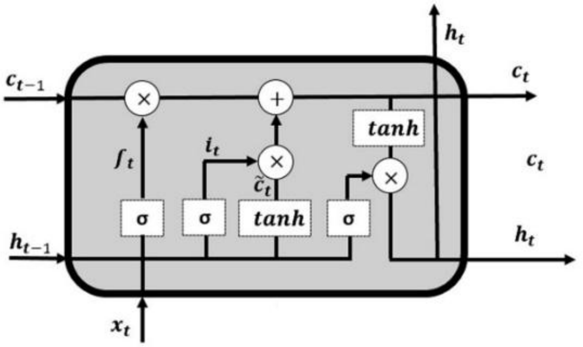


Fig. 4. LSTM architecture.

E. Model Testing

Model testing was carried out using test data to put the previously built GRU-LSTM hybrid model to the test.

F. Calculation of RMSE, MAE and R-Squared

The model test results was used to calculate performance using:

1) Root Mean Squared Error (RMSE): The root of the error value between the predicted and actual values is calculated to test the accuracy of prediction results [23]. The equation for calculating the RMSE value is as follow [27]:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\tilde{y}_i - y_i)^2}{n}} \qquad (13)$$

2) Mean Absolute Error (MAE): MAE measures the absolute difference between predicted and actual data, regardless of whether the difference is positive or negative [28]. The equation for calculating the MAE value is as follow [29]:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\tilde{y}_i - y_i| \qquad (14)$$

3) R-Squared (R^2): R2 or called by coefficient of determination is a statistical measure used to investigate the relationship between actual and predicted data results. R2 can range between $-\infty$ up to 1; the closer the R^2 value is to 1, the better the model fits the dataset [18], [27]. The equation for calculating the value of R^2 is as follows [28]:

$$\bar{\text{Y}} = \frac{1}{n} \sum_{i=1}^{n} y_i \qquad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\tilde{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{\text{Y}} - y_i)^2} \qquad (16)$$

Where n = total of the data, i = (1, 2, 3, 4, 5, …, l), l is the entire data set, $\bar{\text{Y}}$ = the mean value of the actual data, y_{i} = actual value, dan $\tilde{y}_{\text{i}}$ = predicted value.

G. Testing Research Indicators Using Macro Data

In this test scheme, tests was performed to determine the performance of the Hybrid GRU-LSTM model by varying the number of input parameters, data sharing ratios, epochs, batch sizes, neuron sizes, and the arrangement of the model's architectural layers using daily, weekly, and monthly macro air temperature data.

H. Testing the Macro GRU-LSTM Hybrid Model Using Micro Data

In this test scheme, the best model for each use of the data obtained in the previous test scheme was used to predict air temperature on a microscale interval of 3 hours and daily.

I. Comparing the GRU-LSTM Hybrid Model Using GRU and LSTM

In this test scheme, the model performance results of the GRU-LSTM Hybrid Model, the GRU model, and the LSTM Model were compared.

J. Analysis Results

The Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2) were used to calculate the results of model testing. The lower the RMSE and MAE values obtained, the better the performance of the tested model, and the closer to one the R2 value, the better the resulting model. All models' training times were compared to determine which model had the fastest performance.

IV. Results

The GRU-LSTM Hybrid model was created on the Google Collab platform using the Python language and various Python libraries, such as time, numpy, pandas, sklearn, TensorFlow, and Keras. Fig. 5 shows the architecture of the GRU-LSTM Hybrid model [17].

The GRU model was created using 2 Layer GRU and 2 Layer Dense neurons with a size of 240. ReLU activation was used in the first GRU layer, ELU activation in the second GRU layer, and ReLU activation in the two Dense layers. The LSTM model was created using two layers of LSTM and two layers of Dense with 240 neuron sizes. The first LSTM layer was activated with ReLU, the second GRU layer with ELU, and the two Dense layers with ReLU. Using the add() function from keras.layers, the GRU and LSTM models were combined into a parallel GRU-LSTM model. The model now includes 3 Layer Dense with ELU activation and 240 neuron size, as well as 1 Layer Dense with Linear activation and 240 neuron size. Finally, as the model's output, a Dense Layer with one neuron size was added. The Adam Optimizer was used to build the model, with a learning rate of 0.00015, an L2 Regularizer of 10-4, 20 epochs, and a batch size of 64. This GRU-LSTM hybrid model was used to train each test scheme.

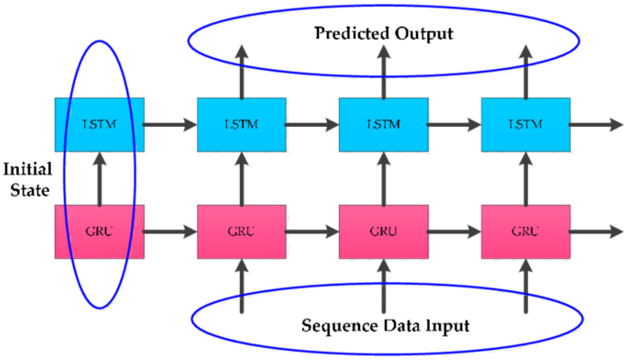


Fig. 5. Architecture of GRU-LSTM hybrid model.

A. Testing Data Preprocessing Techniques

This test was carried out by comparing the use of Min-Max Scaling with the Fast Fourier Transform to 7 NaN Handling techniques. Table III shows the results of testing each data preprocessing scheme using Min-Max Scaling:

TABLE III. COMPARISON OF NAN TECHNIQUES USING MIN-MAX SCALING

Missing Value Techniques	RMSE	MAE	R^2	Duration (s)
Mean	0.6844244536	0.4679480707	-0.1507817381	11.535
Median	0.6739228724	0.4521220334	-0.1240336428	12.755
Last Observation Carried Forward	0.7017514004	0.5292969394	-0.0704251303	11.038
Exponential (Weighted) Moving Average	0.6816600892	0.4810926004	-0.2355169779	11.272
Interpolation Linear	0.7437746843	0.5975289210	0.0155255931	12.202
Interpolation Spline	0.7421904651	0.5940747193	0.0197149326	10.993
Interpolation Time	0.7461391790	0.5987105870	0.0092562629	12.290

Based on this test, it is clear that the various preprocessing stages impact the model's performance. Four of the seven techniques tested with Min-Max Scaling had a negative R^2 value, indicating that the model did not adequately fit the data. This study obtained the best preprocessing stages using the Exponential (Weighted) Moving Average for NaN handling and the Fast Fourier Transform for data transformation, as shown in Table IV.

TABLE IV. COMPARISON OF NAN HANDLING TECHNIQUES USING THE FAST-FOURIER TRANSFORM

Missing Value Techniques	RMSE	MAE	R^2	Duration (s)
Mean	0.1290396570	0.1057177850	0.9590938765	11.822
Median	0.1270965006	0.1043401824	0.9600215361	11.272
Last Observation Carried Forward	0.1391963981	0.1151423968	0.9578841359	12.311
Exponential (Weighted) Moving Average	0.1161321395	0.0949022568	0.9641393763	12.104
Interpolation Linear	0.1638662900	0.1452299168	0.9522139770	11.063
Interpolation Spline	0.1664941613	0.1479380850	0.9506690292	14.679
Interpolation Time	0.1645815491	0.1457774991	0.9517959046	11.428

B. Testing the Total of Data

Table V shows the test results obtained from testing the amount of data using daily macro air temperature data.

TABLE V. RESULTS OF DAILY MACRO DATA TESTING BASED ON TOTAL DATA

Total of the Data	RMSE	MAE	R^2	Duration (s)
50	0.1851642148	0.1498968825	0.9340657953	9.079
100	0.1523455190	0.1254774662	0.9623533541	10.857
200	0.1265524442	0.1079908816	0.9674943384	11.948
300	0.1174715265	0.0959396310	0.9633074235	13.293
500	0.1700496630	0.1323210940	0.9593256173	18.942
1000	0.1802263938	0.1456180500	0.9560157537	22.164
2000	0.1680635190	0.1345518316	0.9591325611	29.965
4000	0.1667981339	0.1313218270	0.9629232147	50.314
8000	0.1735634802	0.1304684863	0.9647927358	86.563

When testing the amount of data with daily average macro air temperature data, 300 data points produced the lowest RMSE and MAE values, 200 data points produced the highest R2 values, and 50 data points produced the shortest model training duration. The test scheme applied 300 data points to obtain the best model performance on this research indicator. The test results obtained from testing the amount of data using weekly macro air temperature data are shown in Table VI.

TABLE VI. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON TOTAL DATA

Total of the Data	RMSE	MAE	R^2	Duration (s)
50	0.0623367201	0.0511048617	0.9455028859	10.697
100	0.1371391608	0.1088339692	0.8295841737	13.122

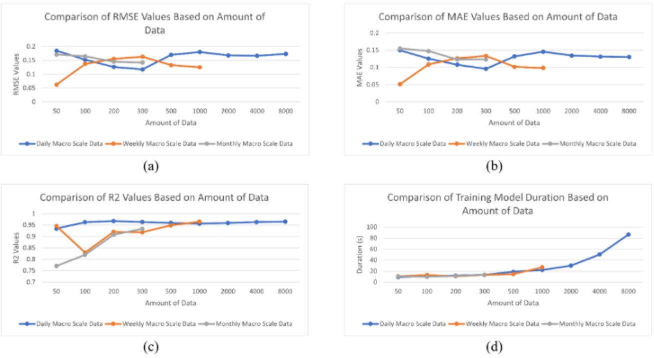
Total of the Data	RMSE	MAE	R^2	Duration (s)
200	0.1554007142	0.1264640826	0.9198697611	10.850
300	0.1634344265	0.1335772436	0.9185153251	13.031
500	0.1330690852	0.1018681449	0.9483418575	14.694
1000	0.1250917579	0.0985749315	0.9648619295	26.889

When testing the amount of data with weekly average macro air temperature data, the lowest RMSE and MAE values were obtained using 50 data, the highest R^2 values were obtained using 1000 data, and the fastest model training duration was obtained using 50 data. The test scheme applied 50 data points to get the best model performance on this research indicator. Table VII shows the results obtained in testing the amount of data using monthly macro air temperature data.

TABLE VII. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON TOTAL DATA

Total of the Data	RMSE	MAE	R^2	Duration (s)
50	0.1712211232	0.1550369370	0.7706810377	10.477
100	0.1649493670	0.1474490374	0.8194394220	9.683
200	0.1452670353	0.1231163900	0.9068988647	11.418
300	0.1420492412	0.1233844163	0.9335440337	13.327

Testing with monthly average macro air temperature data showed that 300 data points yielded the lowest RMSE and highest R², 200 data points had the lowest MAE, and 50 data points resulted in the shortest training time. The test scheme applied 300 data points to achieve the best model performance, as shown in Fig. 6.



(a)

(b)

(c)

(d)

Fig. 6. Analysis of model performance results based on total data (a) RMSE Value, (b) MAE Value, (c) R2 Value, (d) Model training duration.

Increasing the data amount did not continually improve performance; data variations influenced results. For instance, with daily macro temperature data, 300 data points had a lower error rate than 200 or 500, whereas with weekly data, 300 data points had the highest error rate.

C. Parameter Input Testing

Table VIII shows the test results obtained in the input parameter test using daily macro air temperature data.

TABLE VIII. RESULTS OF DAILY MACRO DATA TESTING BASED ON PARAMETER INPUT

Parameter Input	RMSE	MAE	R^2	Duration (s)
1 day	0.4947327263	0.3840819742	0.3491899320	7.326
2 days	0.2478204263	0.1922136994	0.8366996247	9.103
3 days	0.1838271449	0.1453921775	0.9101470651	6.824
4 days	0.1541403522	0.1312984414	0.9368249379	6.975
5 days	0.1393977268	0.1110161938	0.9483316768	7.318
6 days	0.1163984780	0.0952625401	0.9639747016	6.925
7 days	0.1002650901	0.0804510812	0.9732691678	9.645
8 days	0.0902920465	0.0726037929	0.9783223612	7.895
9 days	0.0782327439	0.0637689138	0.9837261575	6.967
10 days	0.0744849001	0.0633215273	0.9852480489	7.635
11 days	0.0731881517	0.0594071249	0.9857572273	7.047
12 days	0.0765548540	0.0585217093	0.9844167352	7.538

The lowest RMSE value was in the use of 11 daily average data to predict the next day's average, the lowest MAE value was in the use of 12 daily average data to predict the next day's average, the highest R^2 value was in the use of 11 daily average data to predict the next day's average, and the fastest model training duration was in the use of 3 daily average data to predict the next day's average. To achieve the best model performance on this research indicator, the test scheme applied 11 Input Parameters that use 11 daily average data to predict the next day's average. The test results obtained in the input parameter test using weekly macro air temperature data is shown in Table IX.

TABLE IX. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON PARAMETER INPUT

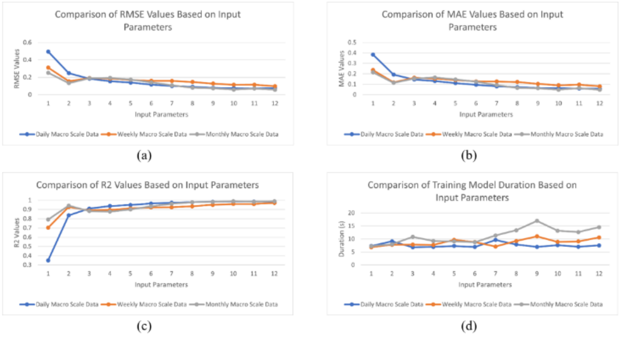
Parameter Input	RMSE	MAE	R^2	Time (s)
1 week	0.3119153610	0.2365223692	0.7032010388	6.849
2 weeks	0.1542141168	0.1180309918	0.9274500509	7.787
3 weeks	0.1909112509	0.1624875251	0.8888135417	7.846
4 weeks	0.1851261596	0.1540472429	0.8954499045	7.708
5 weeks	0.1701543518	0.1398497052	0.9116767640	9.665
6 weeks	0.1594423340	0.1269611427	0.9224474403	8.832
7 weeks	0.1582014176	0.1261986688	0.9236499033	7.102
8 weeks	0.1462549959	0.1217502412	0.9347455294	9.227
9 weeks	0.1272247907	0.1041676132	0.9506221313	11.002
10 weeks	0.1145790877	0.0905048298	0.9599502721	8.880
11 weeks	0.1155339656	0.0965910716	0.9592799586	9.063
12 weeks	0.0976961789	0.0799839595	0.9708831757	10.617

When the input parameters were tested using weekly average macro air temperature data, the lowest RMSE and MAE values were obtained using 12 weekly average data to predict the following week's average, the highest R^2 value was obtained using 12 weekly average data to predict the following week's average, and the shortest model training duration was obtained by using one weekly average data to predict the following week's average. To achieve the best model performance on this research indicator, the test scheme applied 12 Input Parameters that used 12 weekly average data to predict the following week's average. The test results obtained in the input parameter test using monthly macro air temperature data are shown in Table X.

TABLE X. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON INPUT DATA

Parameter Input	RMSE	MAE	R^2	Duration (s)
1 month	0.2520620441	0.2141007492	0.7907473106	7.227
2 months	0.1328624270	0.1141378974	0.9418619443	7.976
3 months	0.1890981064	0.1546116300	0.8822311320	10.817
4 months	0.1929430235	0.1650484307	0.8773932727	9.214
5 months	0.1738967881	0.1457596050	0.9004046042	9.102
6 months	0.1413953839	0.1238171764	0.9341544236	8.743
7 months	0.1078759913	0.0922371984	0.9616729194	11.342
8 months	0.0801424696	0.0654195946	0.9788465317	13.374
9 months	0.0738925268	0.0626033554	0.9820172057	16.962
10 months	0.0588448555	0.0470539990	0.9885955816	13.186
11 months	0.0712976266	0.0640248735	0.9832580405	12.678
12 months	0.0568075286	0.0465924708	0.9893715990	14.567

When the input parameters were tested using monthly average macro air temperature data, the lowest RMSE and MAE values were obtained using 12 monthly average data to predict the next month's average, the highest R^2 value was obtained using 12 monthly average data to predict the next month's average, and the shortest model training duration was obtained using one monthly average data to predict the next month's average. To achieve the best model performance on this research indicator, the test scheme applied 12 Input Parameters that used 12 monthly average data to predict the next month's average. The analysis of the effect of the number of input parameters on the performance of the Hybrid GRU-LSTM model is shown in Fig. 7.



(a)

(b)

(c)

(d)

Fig. 7. Analysis of model performance results based on input parameters (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model training duration.

Increasing the number of input parameters affected the model's performance. Each data use almost always resulted in a non-significant decrease in error rate from 3 input parameters to 12 input parameters. Increasing the number of input parameters from one to two significantly improved model performance, so using a GRU-LSTM hybrid model with one input parameter was not recommended.

D. Testing of Data Sharing Ratio

Table XI shows the test results obtained when testing the data sharing ratio using daily macro air temperature data.

TABLE XI. RESULTS OF DAILY MACRO DATA TESTING BASED ON DATA SHARING RATIO

Data Sharing Ratio	RMSE	MAE	R^2	Duration (s)
50 : 50	0.1746709404	0.1380718406	0.9607895299	10.457
60 : 40	0.1704924270	0.1366307721	0.9618051593	13.888
70 : 30	0.1854317031	0.1515861214	0.9537578721	13.381
80 : 20	0.1651222128	0.1313805477	0.9569454454	11.056
90 : 10	0.1164872255	0.0956705397	0.9639197460	12.978

Testing the data-sharing ratio using daily average macro air temperature data revealed that a 90:10 ratio provided the best model performance, with the lowest RMSE and MAE values and the highest R² value. In contrast, a 50:50 ratio significantly reduced the training time but at the cost of lower model accuracy. These findings indicate that a more extensive training dataset (90%) enhances model learning, leading to better predictive performance. The test scheme applied a 90:10 data-sharing ratio to optimize performance based on these results. Further analysis of the impact of data-sharing ratios using weekly average macro air temperature data is presented in Table XII.

TABLE XII. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON DATA SHARING RATIO

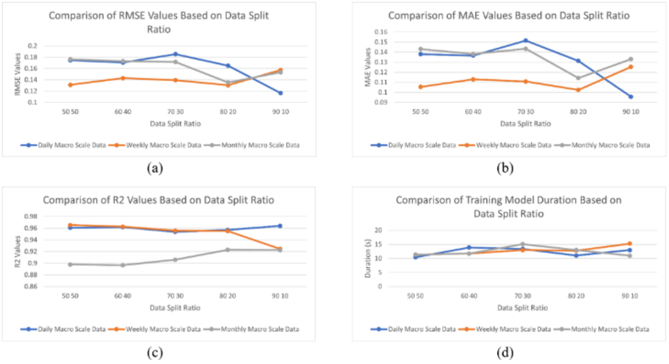
Data Sharing Ratio	RMSE	MAE	R^2	Duration (s)
50 : 50	0.1310978587	0.1054439719	0.9654619062	11.361
60 : 40	0.1430146436	0.1128988945	0.9627798949	11.728
70 : 30	0.1393393249	0.1107925050	0.9557609879	12.986
80 : 20	0.1304721654	0.1024591358	0.9551738898	12.774
90 : 10	0.1573456424	0.1253423013	0.9244736860	15.262

When testing the data sharing ratio with weekly average macro air temperature data, the 80:20 ratio had the lowest RMSE and MAE values, the 50:50 ratio had the highest R^2 value, and the 50:50 ratio had the shortest model training duration. Table XIII shows the results obtained when testing the data-sharing ratio using monthly macro air temperature data.

TABLE XIII. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON DATA SHARING RATIO

Data Sharing Ratio	RMSE	MAE	R^2	Duration (s)
50 : 50	0.1763222753	0.1431057810	0.8977571251	11.401
60 : 40	0.1730447912	0.1381097616	0.8966686071	11.724
70 : 30	0.1717020795	0.1433728574	0.9059654873	15.100
80 : 20	0.1353351706	0.1142427518	0.9230716267	13.004
90 : 10	0.1533416137	0.1331341740	0.9225580644	10.967

When testing the data sharing ratio with monthly average macro air temperature data, the 80:20 ratio produced the lowest RMSE and MAE values, the 80:20 ratio produced the highest R^2, and the 90:10 ratio produced the shortest model training duration. To achieve the best model performance on this research indicator, the test scheme applied an 80:20 Data Sharing Ratio. Fig. 8 shows the analysis of the effect of the data-sharing ratio on the performance of the Hybrid GRU-LSTM model.



(a)

(b)

(c)

(d)

Fig. 8. Analysis of model performance results based on data sharing ratio (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model training duration.

Changes in the data share ratio affected model performance, but data also affected it. Based on this test, the best data-sharing ratio for each data set was not found.

E. Epoch Testing

The epoch test results obtained using daily macro air temperature data are shown in Table XIV.

TABLE XIV. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON EPOCH

Epochs	RMSE	MAE	R^2	Duration (s)
10	0.1412090605	0.1098501957	0.9469801969	10.081
20	0.1178582018	0.0964495148	0.9630654677	12.043
30	0.1159206586	0.0952576068	0.9642698645	17.625
40	0.1158310883	0.0951873015	0.9643250595	18.672
50	0.1162878559	0.0955324685	0.9640431441	18.745
60	0.1159376773	0.0948069858	0.9642593724	21.236
70	0.1173887879	0.0965781303	0.9633590926	22.936
80	0.1162496231	0.0956332868	0.9640667838	27.657
90	0.1163170192	0.0956855566	0.9640251069	27.304
100	0.1161146882	0.0951103535	0.9641501532	30.338

On this research indicator, the test scheme with 40 Epoch yielded the best model performance. The epoch test results obtained using weekly macro air temperature data is shown in Table XV:

TABLE XV. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON EPOCH

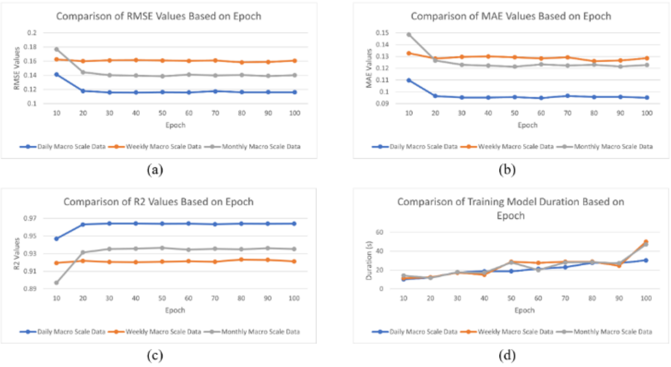
Epochs	RMSE	MAE	R^2	Duration (s)
10	0.1625175407	0.1327365961	0.9194270372	11.416
20	0.1600507480	0.1281993821	0.9218544474	12.325
30	0.1612495635	0.1296542749	0.9206794082	17.127
40	0.1615903931	0.1300313808	0.9203437375	15.127
50	0.1609623787	0.1293785623	0.9209616959	28.846
60	0.1603825770	0.1283413292	0.9215300773	27.610
70	0.1610848606	0.1292554419	0.9208413641	28.765
80	0.1584860572	0.1260031276	0.9233749144	28.937
90	0.1589206752	0.1266394984	0.9229540786	24.613
100	0.1607210281	0.1285202697	0.9211985418	50.008

The test scheme with 80 Epoch yielded the best model performance on this research indicator. The epoch test results obtained using monthly macro air temperature data is shown in Table XVI.

TABLE XVI. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON EPOCH

Epochs	RMSE	MAE	R^2	Duration (s)
10	0.1768219120	0.1485940880	0.8970258287	14.240
20	0.1443285145	0.1265315739	0.9313942646	11.598
30	0.1401524357	0.1229330155	0.9353069778	17.838
40	0.1396633099	0.1222866899	0.9357577414	17.241
50	0.1388041207	0.1213296601	0.9365457289	28.092
60	0.1409545112	0.1233228180	0.9345643983	20.015
70	0.1397881914	0.1223504527	0.9356428042	28.075
80	0.1404582682	0.1230280480	0.9350243311	28.311
90	0.1391538159	0.1214379346	0.9362256001	27.242
100	0.1401636231	0.1227904198	0.9352966494	47.246

The test scheme with 50 epochs yielded the best model performance on this research indicator. Fig. 9 analyzes the effect of the number of epochs on the performance of the Hybrid GRU-LSTM model.



(a) (b) (c) (d)

Fig. 9. Analysis of model performance results based on the number of Epoch (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model training duration.

The number of epochs did not significantly affect the model's performance; only when the number of epochs was increased from 10 to 20, the error rate decreased by 0.02 - 0.04, and the MAE value decreased by 0.03 - 0.02. Other tests tended to have the same error rate, with the RMSE value being 0.12 for daily macro data, 0.16 for weekly macro data, and 0.14 for monthly macro data.

F. Batch Size Testing

Table XVII presents the results of the batch size test using daily macro air temperature data, comparing model performance across different batch sizes. The table shows how batch size affects error values (RMSE, MAE, and R^2) and training stability. Smaller batch sizes may improve generalization but require more iterations, while larger batch sizes can speed up training but may lead to overfitting. This analysis helps determine the optimal batch size for the Hybrid GRU-LSTM model.

TABLE XVII. RESULTS OF DAILY DATA TESTING BASED ON BATCH SIZE

Batch Size	RMSE	MAE	R^2	Duration (s)
32	0.1165713735	0.0960287397	0.9638675999	16.838
64	0.1161702623	0.0956351955	0.9641158285	12.916
128	0.1182457040	0.0959229463	0.9628221966	10.385

The test scheme with 64 Batch Sizes yielded the best model performance on this research indicator. The results of the batch size test using weekly macro air temperature data is shown in Table XVIII:

TABLE XVIII. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON BATCH SIZE

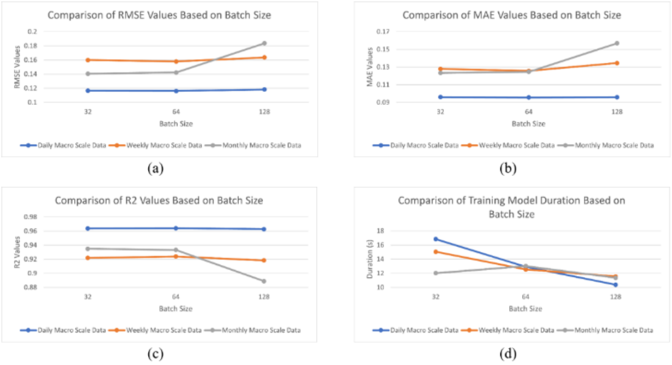
Batch Size	RMSE	MAE	R^2	Duration (s)
32	0.1599404943	0.1278730529	0.9219620741	15.048
64	0.1579052215	0.1257131268	0.9239355319	12.542
128	0.1635250788	0.1345633975	0.9184249057	11.566

The test scheme with 64 Batch Sizes yielded the best model performance on this research indicator. The results of the batch size test using monthly macro air temperature data are shown in Table XIX:

TABLE XIX. RESULTS OF MONTHLY DATA TESTING BASED ON BATCH SIZE

Batch Size	RMSE	MAE	R^2	Duration (s)
32	0.1405752504	0.1233503645	0.9349160546	12.019
64	0.1423798040	0.1245906945	0.9332343744	13.026
128	0.1836717030	0.1569849137	0.8888931982	11.357

The test scheme with 32 batch sizes yielded the best model performance for this research indicator. Fig. 10 shows the analysis of the effect of batch size on the performance of the Hybrid GRU-LSTM model.



(a) (b) (c) (d)

Fig. 10. Analysis of model performance results based on total batch size (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model training duration.

Increasing the number of batch sizes did not affect the model's performance. Except for using 128 batch sizes, the model performance was consistent across tests, with the RMSE value at 0.12 for daily macro data, 0.16 for weekly macro data, and 0.14 for monthly macro data.

G. Neuron Size Testing

The neuron size test results obtained using daily macro air temperature data are shown in Table XX.

TABLE XX. RESULTS OF DAILY MACRO DATA TESTING BASED ON NEURON SIZE

Neuron Size	RMSE	MAE	R^2	Duration (s)
10	0.1474003564	0.1178712074	0.9422289780	8.242
20	0.1516744342	0.1224722308	0.9388301028	11.471
30	0.1472699419	0.1163316259	0.9423311602	9.093
60	0.1228602713	0.0997256246	0.9598638304	10.605
120	0.1157847143	0.0940748270	0.9643536193	10.974
240	0.1165024016	0.0957993578	0.9639103442	12.659

Based on this research indicator, the test scheme with a neuron size of 120 produced the best model performance. The results of the neuron size test using weekly macro air temperature data are presented in Table XXI.

TABLE XXI. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON NEURON SIZE

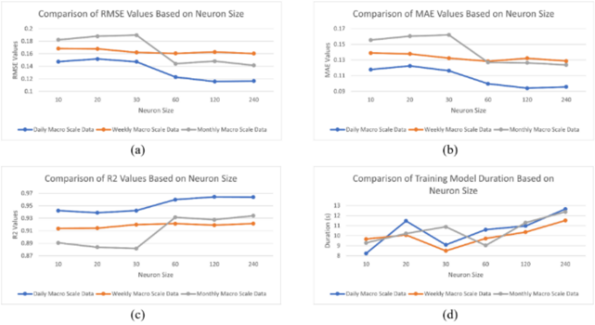
Neuron Size	RMSE	MAE	R^2	Duration (s)
10	0.1682801592	0.1389587059	0.9136117495	9.663
20	0.1677301907	0.1378873890	0.9141754902	10.070
30	0.1620301639	0.1322839767	0.9199095760	8.496
60	0.1605895360	0.1284312306	0.9213274301	9.727
120	0.1627621243	0.1322071073	0.9191843353	10.361
240	0.1603681788	0.1286783949	0.9215441658	11.524

The test scheme with 240 Neuron Size produced the best model performance on this research indicator. The neuron size test results obtained using monthly macro air temperature data are shown in Table XXII.

TABLE XXII. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON NEURON SIZE

Neuron Size	RMSE	MAE	R^2	Duration (s)
10	0.1822280375	0.1554668319	0.8906329402	9.285
20	0.1878851786	0.1603905481	0.8837370911	10.220
30	0.1896031054	0.1620292374	0.8816012730	10.887
60	0.1441418736	0.1270096016	0.9315715872	9.028
120	0.1481143123	0.1263619519	0.9277479467	11.299
240	0.1413978662	0.1237826017	0.9341521117	12.373

The test scheme with 240 neuron sizes produced the best model performance for this research indicator. Fig. 11 examines the effect of neuron size on the performance of the Hybrid GRU-LSTM model. Increasing the number of neuron sizes affected the model's performance. Increasing the number of neuron sizes from 30 to 60 significantly improved model performance, and the error rate decreased by around 0.03 - 0.05 in tests using daily macro data and monthly macro data.



(a)

(b)

(c)

(d)

Fig. 11. Analysis of model performance based on number of neuron sizes (a) RMSE value, (b) MAE value, (c) R^2 value, (d) Training duration.

H. Layer Testing

Layer testing was conducted by modifying the model's layer arrangement. The scheme used to test the layer is shown in Table XXIII.

TABLE XXIII. LAYER TESTING SCHEME

Scheme	Description
Scheme 1	In the GRU-LSTM parallel model, each has 1 GRU/LSTM Layer and 1 Dense Layer
Scheme 2	In the GRU-LSTM parallel model, each has 2 GRU/LSTM Layer and 1 Dense Layer
Scheme 3	In the GRU-LSTM parallel model, each has 3 GRU/LSTM Layer and 1 Dense Layer
Scheme 4	In the parallel GRU-LSTM model, each has 1 GRU/LSTM Layer and 2 Dense Layers
Scheme 5	In the parallel GRU-LSTM model, each has 2 GRU/LSTM Layer and 2 Dense Layers
Scheme 6	In the parallel GRU-LSTM model, each has 3 GRU/LSTM Layer and 2 Dense Layers
Scheme 7	In the parallel GRU-LSTM model, each has 1 GRU/LSTM Layer and 3 Dense Layers
Scheme 8	In the parallel GRU-LSTM model, each has 2 GRU/LSTM Layer and 3 Dense Layers
Scheme 9	In the parallel GRU-LSTM model, each has 3 GRU/LSTM Layer and 3 Dense Layers
Scheme 10	In the parallel GRU-LSTM model, each has 2 GRU/LSTM Layers, 1 Dropout Layer, and 2 Dense Layers
Scheme 11	In the parallel GRU-LSTM model, each has 1 Layer GRU/LSTM, 1 Layer Dropout, 1 Layer GRU/LSTM, 1 Layer Dropout, and 2 Layers Dense
Scheme 12	Using 1 Layer Dense after implementing the GRU-LSTM parallel model
Scheme 13	Using 2 Layer Dense after implementing the GRU-LSTM parallel model
Scheme 14	Using 3 Layer Dense after implementing the GRU-LSTM parallel model
Scheme 15	Using 4 Layer Dense after implementing the GRU-LSTM parallel model

The layer test results obtained using daily macro air temperature data are shown in Table XXIV:

TABLE XXIV. RESULTS OF DAILY MACRO DATA TESTING BASED ON LAYER

Layers	RMSE	MAE	R^2	Duration (s)
Scheme 1	0.1207798570	0.0960987393	0.9612115875	9.724
Scheme 2	0.1170422441	0.0950615363	0.9635751088	13.741
Scheme 3	0.1164223853	0.0956228881	0.9639599015	16.659
Scheme 4	0.1193676096	0.0970861849	0.9621133700	11.356
Scheme 5	0.1171691442	0.0944435854	0.9634960804	12.952
Scheme 6	0.1166691458	0.0958401778	0.9638069636	21.968
Scheme 7	0.1168505315	0.0937090292	0.9636943374	12.021
Scheme 8	0.1156994266	0.0948360588	0.9644061147	13.936

Layers	RMSE	MAE	R^2	Duration (s)
Scheme 9	0.1160412787	0.0953250178	0.9641954685	23.327
Scheme 10	0.1205027634	0.0988276975	0.9613893604	14.647
Scheme 11	0.1491179453	0.1198267348	0.9408747752	13.310
Scheme 12	0.1152844223	0.0943347440	0.9646610014	13.840
Scheme 13	0.1160242160	0.0946575514	0.9642059972	12.546
Scheme 14	0.1163461744	0.0952354286	0.9640070702	13.246
Scheme 15	0.1166766606	0.0955005634	0.9638023009	15.742

After implementing the GRU-LSTM parallel model, test scheme 12 achieved the best model performance on this research indicator. The layer test results obtained using weekly macro air temperature data are shown in Table XXV:

TABLE XXV. RESULTS OF WEEKLY MACRO DATA TESTING BASED ON LAYER

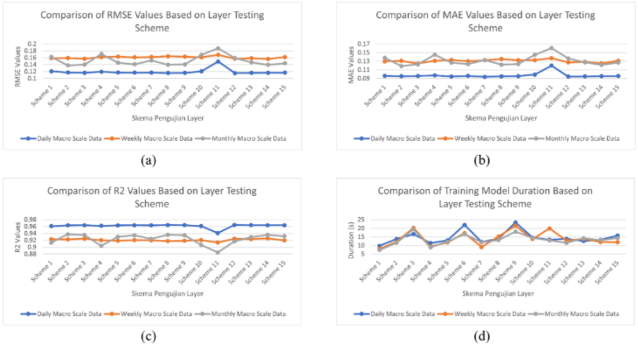
Layers	RMSE	MAE	R^2	Duration (s)
Scheme 1	0.1585801594	0.1294481339	0.9232838940	8.031
Scheme 2	0.1592887200	0.1307524235	0.9225968037	12.092
Scheme 3	0.1570656604	0.1245989907	0.9247422310	20.229
Scheme 4	0.1617638061	0.1308678964	0.9201726773	9.351
Scheme 5	0.1632904734	0.1330755769	0.9186588054	11.745
Scheme 6	0.1612453088	0.1299866103	0.9206835940	17.198
Scheme 7	0.1621873110	0.1322764476	0.9197541471	9.056
Scheme 8	0.1642158546	0.1350723155	0.9177342578	15.201
Scheme 9	0.1632028674	0.1321671039	0.9187460617	21.451
Scheme 10	0.1612673131	0.1325025413	0.9206619447	13.687
Scheme 11	0.1683629685	0.1372377593	0.9135267067	19.908
Scheme 12	0.1573133076	0.1271241494	0.9245047244	12.781
Scheme 13	0.1586990147	0.1287871174	0.9231688540	13.876
Scheme 14	0.1563978327	0.1247707114	0.9253808476	11.979
Scheme 15	0.1619698941	0.1309035883	0.9199691468	11.850

The best model performance on this research indicator was obtained using Test Scheme 14 after implementing the parallel GRU-LSTM model. The layer test results obtained using monthly macro air temperature data are shown in Table XXVI.

TABLE XXVI. RESULTS OF MONTHLY MACRO DATA TESTING BASED ON LAYER

Layers	RMSE	MAE	R^2	Duration (s)
Scheme 1	0.1624325311	0.1378718658	0.9131035240	7.353
Scheme 2	0.1377480782	0.1186656237	0.9375075951	11.547
Scheme 3	0.1401070555	0.1226267907	0.9353488651	19.437
Scheme 4	0.1708900003	0.1450752292	0.9038189659	8.928
Scheme 5	0.1455686171	0.1262893909	0.9302102483	12.174
Scheme 6	0.1410352850	0.1234592980	0.9344893812	16.611
Scheme 7	0.1519643950	0.1326691130	0.9239428879	11.777
Scheme 8	0.1394894457	0.1219289526	0.9359175898	13.114
Scheme 9	0.1407462996	0.1233650677	0.9347575725	18.069
Scheme 10	0.1684723507	0.1447563020	0.9065211390	14.408
Scheme 11	0.1872637291	0.1604907882	0.8845049222	12.991
Scheme 12	0.1590729268	0.1363708035	0.9166609234	11.487
Scheme 13	0.1464591550	0.1272421046	0.9293537377	14.149
Scheme 14	0.1395472120	0.1213907857	0.9358645023	13.124
Scheme 15	0.1439520390	0.1267799049	0.9317517087	14.256

Test Scheme 2 achieved the best model performance on this research indicator using a parallel GRU-LSTM model with 2 Layers of GRU/LSTM and one dense layer. Fig. 12 shows the analysis of the effect of layers on the performance of the Hybrid GRU-LSTM model.



(a)

(b)

(c)

(d)

Fig. 12. Analysis of model performance results based on layer testing scheme (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model Training Duration.

Layer changes had no discernible effect on model performance. Increasing or decreasing the number of layers did not affect the model error rate; the model error rate only increased significantly when the Dropout Layer was added in Schemes 10 and 11. As a result, using the GRU-LSTM hybrid model with a dropout layer in testing with 300 data points was not recommended.

I. Testing the GRU-LSTM Hybrid Model for Macro Data for Micro Data

This test applied a model trained on macro air temperature data to test micro air temperature data. The model used is as follows:

1) Model 1: This GRU-LSTM hybrid model was built with Daily Macro Data, incorporating 11 daily averages to predict the average of the following days.

2) Model 2: The GRU-LSTM LSTM hybrid model was built using Weekly Macro Data and 50 weekly average data points.

3) Model 3: This GRU-LSTM hybrid model was built using monthly macro data by combining 12 monthly averages to predict the next month’s average.

Table XXVII shows the test results obtained in the GRU-LSTM Hybrid Model t2st, which was built using macro data to predict micro data every 3 hours:

TABLE XXVII. RESULTS OF MACRO DATA MODEL USING MICRO DATA PER 3 HOURS

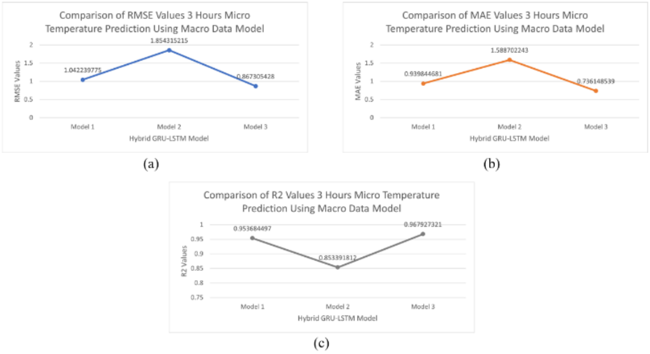
GRU-LSTM Hybrid Models	RMSE	MAE	R^2
Model 1	1.0422397749	0.9398446809	0.9536844965
Model 2	1.8543152148	1.5887022426	0.8533918118
Model 3	0.8673054280	0.7361485392	0.9679273214

A test that used Model 3 to predict micro air temperature every 3 hours had the best model performance in this test. The temperature changes during the day were significant, reaching 3 - 14 °C, so micro air temperature data with 3-hour intervals had several outlier data. These significant temperature changes affected the model’s performance. As a result, using models trained on daily, weekly, and monthly macro data still resulted in a relatively high error rate. The test results obtained in the GRU-LSTM Hybrid Model test, which was built using macro data to predict daily microdata, are shown in Table XXVIII:

TABLE XXVIII. TESTING RESULTS OF THE MACRO DATA MODEL USING DAILY MICRO DATA

GRU-LSTM Hybrid Models	RMSE	MAE	R^2
Model 1	0.2307456696	0.1950775580	0.9812115284
Model 2	0.3606277068	0.2694757832	0.9541074158
Model 3	0.2270860682	0.1908012237	0.9818027687

The best model performance in this test was in a test that used Model 3 to predict micro air temperature every 3 hours. The graph of model performance results is shown in Fig. 13.



(a)

(b)

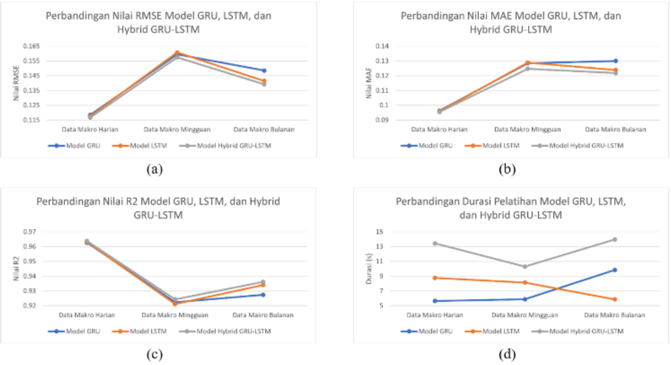
(c)

Fig. 13. Analysis of macro data model performance results for prediction of microtemperature per 3 hours (a) RMSE value, (b) MAE value, (c) R^2 value.

The model's performance was tested in predicting daily micro air temperature using a model trained using daily, weekly, and monthly macro data. This model showed a lower error rate than testing using data with 3-hour intervals.

J. GRU, LSTM and Hybrid GRU-LSTM Model Testing

This evaluation compares the performance of the GRU, LSTM, and Hybrid GRU-LSTM models in handling sequential data. Fig. 14 presents the results for each test scheme, highlighting their accuracy and efficiency. This provides insights into the strengths and suitability of each model for forecasting tasks.



(a)

(b)

(c)

(d)

Fig. 14. Analysis of performance model results of GRU Model, LSTM Model, and GRU-LSTM Hybrid Model (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model Training Duration.

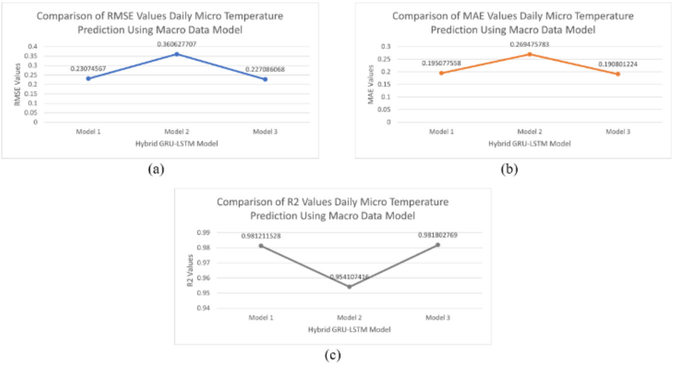
Table XXIX compares the Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Hybrid GRU-LSTM models. This table highlights their predictive accuracy, efficiency, and overall performance in handling sequential data.

In every use of the data tested, the GRU-LSTM Hybrid Model outperformed the GRU-only and LSTM-only Models. Fig. 15 shows a performance comparison analysis of the GRU, LSTM, and Hybrid GRU-LSTM Modes. The prediction results of the GRU-only model, the LSTM-only model, and the GRU-LSTM Hybrid model on daily, weekly, and monthly macro data did not differ significantly. The model performance results showed significant results. If larger quantities were predicted, this has not been investigated further.

TABLE XXIX. COMPARISON RESULTS OF GRU, LSTM, AND GRU-LSTM HYBRID MODELS

Data Using	Model Using	RMSE	MAE	R^2	Duration (s)
Daily Macro Data	GRU model	0.1184584205	0.0964402422	0.9626883153	5.650
	LSTM models	0.1176304218	0.0959534831	0.9632080936	8.774
	GRU-LSTM Hybrid models	0.1166597038	0.0955331357	0.9638128215	13.437
Weekly Macro Data	GRU model	0.1594879534	0.1285021928	0.9224030555	5.875
	LSTM models	0.1607783534	0.1289006750	0.9211423186	8.146
	GRU-LSTM Hybrid models	0.1575065981	0.1247597571	0.9243190886	10.293

Data Using	Model Using	RMSE	MAE	R^2	Duration (s)
Monthly Macro Data	GRU model	0.1485245039	0.1300583339	0.9273471991	9.837
	LSTM models	0.1415222437	0.1239307644	0.9340362175	5.858
	GRU-LSTM Hybrid models	0.1392682847	0.1217903526	0.9361206344	13.975



(a)

(b)

(c)

Fig. 15. Analysis of performance model results of GRU Model, LSTM Model, and GRU-LSTM Hybrid Model (a) RMSE Value, (b) MAE Value, (c) R^2 Value, (d) Model training duration.

V. Discussion

The choice of data preprocessing techniques significantly influenced the performance of the GRU-LSTM hybrid model. Using Min-Max Scaling with various NaN handling methods resulted in suboptimal performance, with four out of seven techniques showing negative R² values, indicating a poor model fit. However, applying the Fast Fourier Transform (FFT) improved model accuracy across all NaN handling techniques. The Exponential (Weighted) Moving Average technique produced the best results with the lowest RMSE and MAE values and the highest R² value. These results indicate that FFT effectively transforms data so that the model can better recognize patterns.

Increasing the amount of training data did not continuously improve model performance. In daily macro temperature data, 300 data points produced the lowest RMSE and MAE values, while 200 data points yielded the highest R² value. However, the error rate also increased when the data size increased to 500 or more. A similar pattern was found in the weekly and monthly data tests, where the optimal amount of data varied. These results suggest that excessive data can lead to information overload or noise accumulation, which disrupts model accuracy. This finding is significant for real-world applications where computational resources are limited. This study provides a practical guideline for selecting training data without unnecessary computational costs by identifying the optimal data size for different timescales.

The number of input parameters played a crucial role in improving prediction accuracy. For daily macro temperature predictions, the model achieved the best performance using 11 days of historical data, while for weekly and monthly data, 12 weeks and 12 months of historical data provided the most optimal results. Increasing input parameters beyond a certain threshold did not significantly enhance model performance. This indicates that only a certain amount of historical data provides relevant information for the model to predict future air temperatures. In practical applications, this insight helps optimize data storage and processing requirements by preventing excessive use of historical data that does not contribute to better predictions.

The ratio of training to testing data affected the model's effectiveness. A 90:10 split produced the best daily macro temperature data accuracy, while an 80:20 split yielded optimal results in weekly and monthly datasets. This difference suggests that the optimal ratio depends on the scale of the data used. An extreme split, such as 50:50, speeds up training time but reduces the model's ability to generalize new data. Conversely, a 90:10 ratio risks overfitting because the training data dominates too much. These findings are helpful for practitioners who need to balance training time and model generalization, particularly in operational forecasting systems where real-time updates are essential.

The number of epochs used in training did not significantly affect model accuracy. The most notable improvement occurred between 10 and 20 epochs, where RMSE and MAE values decreased significantly. However, after exceeding 20 epochs, performance gains became insignificant, indicating a saturation point in model training. The best results were obtained with 40 epochs for daily data, 80 epochs for weekly data, and 50 epochs for monthly data. These differences suggest that the model requires a varying number of epochs depending on the scale of the data used. Training time can be optimized for practical deployment by selecting the appropriate number of epochs, reducing computational overhead without sacrificing prediction accuracy.

Batch size had minimal impact on model accuracy. A batch size of 64 yielded the best daily and weekly data performance, while a batch size of 32 was more optimal for monthly data. However, increasing the batch size to 128 slightly decreased model performance. These results indicate that while batch size optimization can affect training stability, changes in batch size do not significantly impact prediction accuracy. This suggests practitioners can choose moderate batch sizes to balance memory usage and model performance. This makes the model scalable for different computing environments, from high-performance servers to edge devices.

Increasing the number of neurons positively affected model accuracy, mainly when the number of neurons increased from 30 to 60, where RMSE values significantly decreased. The best results were obtained with 120 neurons for daily data and 240 neurons for weekly and monthly data. This suggests that more neurons allow the model to capture more complex patterns, but an excessive number of neurons risks overfitting. In practical applications, understanding the optimal number of neurons helps in designing efficient models that maximize accuracy without unnecessary computational complexity, making the model more scalable for large-scale implementation.

Changes in the number of GRU/LSTM and Dense layers did not significantly impact prediction accuracy. The best configuration for daily data was obtained using one Dense Layer after GRU-LSTM, weekly data using three Dense Layers after GRU-LSTM, and monthly data using two GRU/LSTM Layers and one Dense Layer. Adding Dropout Layers worsened model performance, indicating that dropout is unsuitable for this air temperature prediction scenario. These insights provide practical recommendations for model architecture design, allowing practitioners to avoid unnecessary layers that do not contribute to accuracy improvements, thereby reducing training time and computational costs.

Applying a model trained on macro temperature data to predict micro temperature data every three hours resulted in a high error rate. The large temperature fluctuations within this timeframe made it difficult for the model to recognize temperature change patterns. Among the three models tested, the model trained with monthly data performed the best. This indicates that longer-term macro data provides more stable patterns for micro-temperature predictions. However, the still relatively high error rate suggests that further adjustments are needed for the model to predict micro temperatures accurately. This finding is particularly relevant for real-world applications where localized micro-temperature forecasts are required, such as in precision agriculture or climate-sensitive industries. Future work should explore additional feature engineering techniques or hybrid models incorporating real-time weather variables.

The GRU-LSTM hybrid model consistently performed better than the standalone GRU or LSTM models in all test scenarios. Although the differences in RMSE, MAE, and R^2 values between the three models were insignificant, the hybrid model still showed slightly higher accuracy. However, this advantage came at the cost of longer training times. These results indicate that using only the GRU or LSTM model is sufficient for applications requiring fast predictions. In contrast, the GRU-LSTM hybrid model is recommended for more precise forecasting needs. From a scalability perspective, this suggests that the choice of model should depend on computational resources and accuracy requirements. The standalone GRU or LSTM model is more suitable for real-time applications with limited processing power. In contrast, the GRU-LSTM hybrid model is ideal for high-accuracy forecasting in research and large-scale monitoring systems.

VI. Conclusion

The objective of this study was to see how the number of input parameters, number of data sharing ratios, number of epochs, number of batch sizes, number of neuron sizes, and layer arrangement of the model architecture affected the performance results of the GRU-LSTM hybrid model. The best model performance was obtained from all tests using Macro Air Temperature Data when testing with monthly Macro Air Temperature Data and a 12 Input Parameter scheme. The obtained RMSE is 0.056807, the MAE is 0.046592, and the R^2 is 0.989371. Because the error rate derived from the RMSE and MAE values is relatively low for prediction, the GRU-LSTM Hybrid Model is appropriate for predicting macro air temperature. The data preprocessing stages, the number of input parameters used, and the presence or absence of a Dropout Layer in the model architecture are the indicators that have the most significant influence on model performance. Furthermore, this study investigates whether a model trained on macro data can be used to predict microdata. The GRU-LSTM Hybrid Model, which was built using monthly macro data and 12 Parameter Inputs, produced the best results in predicting Micro Air Temperatures every 3 Hours and Daily Micro Air Temperatures. In microdata tests at daily intervals, the best RMSE, MAE, and R^2 values were 0.227086, 0.190801, and 0.981802, respectively. Because the error rate obtained in testing using daily microdata is relatively low, it can be concluded that micro air temperature predictions can be performed using the GRU-LSTM Hybrid Model, which was trained using monthly macro temperature data. Because of their high accuracy, the results of the 12 input parameters can be used to build a time series air temperature prediction system. The number of inputs indicates the impact on the model's performance results. The Hybrid GRU-LSTM model with 12 inputs can be used to design a temperature prediction application in a wetland environment.

Acknowledgment

This research was funded by the DRTPM Higher Education Excellence Research scheme with Agreement/Contract Number 026/E5/PG.02.00.PL/2023 and Internal PNBP Funds of Universitas Lambung Mangkurat with contract Number 615.82/UN8.2/PL/2023

References

[1] A. Ibrahem Ahmed Osman, A. Najah Ahmed, M. F. Chow, Y. Feng Huang, and A. El-Shafie, “Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia,” Ain Shams Eng. J., vol. 12, no. 2, 2021, doi: 10.1016/j.asej.2020.11.011.

[2] Z. Yahya Dewangga and S. Koesuma, “Development of forest fire early warning system based on the wireless sensor network in Lawu Mountain.,” J. Phys. Conf. Ser., vol. 1153, no. 1, 2019, doi: 10.1088/1742-6596/1153/1/012025.

[3] A. Ashok, H. P. Rani, and K. V. Jayakumar, “Monitoring of dynamic wetland changes using NDVI and NDWI based landsat imagery,” Remote Sens. Appl. Soc. Environ., vol. 23, no. May, p. 100547, 2021, doi: 10.1016/j.rsase.2021.100547.

[4] S. J. Lite, K. J. Bagstad, and J. C. Stromberg, “Riparian plant species richness along lateral and longitudinal gradients of water stress and flood disturbance, San Pedro River, Arizona, USA,” J. Arid Environ., vol. 63, no. 4, pp. 785–813, 2005, doi: 10.1016/j.jaridenv.2005.03.026.

[5] R. Tawalbeh, F. Alasali, Z. Ghanem, M. Alghazzawi, and A. Abu-raideh, “Innovative Characterization and Comparative Analysis of Water Level Sensors for Enhanced Early Detection and Warning of Floods,” 2023.

[6] N. Novitasari, Y. Sari, Y. F. Arifin, N. F. Mustamin, and E. Maulidiya, “Use Of UAV Images For Peatland Cover Classification Using The Convolutional Neural Network Method,” J. Southwest Jiaotong Univ., vol. 6, no. 3, pp. 1–13, 2023.

[7] A. F. Zulkarnain, Y. Sari, and R. Rakhmadani, “Monitoring System for Early Detection of Fire in Wetlands based Internet of Things (IoT) using Fuzzy Methods,” IOP Conf. Ser. Mater. Sci. Eng., vol. 1115, no. 1, p. 012007, 2021, doi: 10.1088/1757-899x/1115/1/012007.

[8] A. R. Baskara, Y. Sari, A. A. Anugerah, E. S. Wijaya, and R. A. Pramunendar, “Fire Detection In Wetland Using YOLOv4 And Deep Learning Architecture,” in 2022 Seventh International Conference on Informatics and Computing (ICIC), 2022, pp. 1–6. doi: 10.1109/ICIC56845.2022.10006963.

[9] B. Guha and G. Bandyopadhyay, “Gold Price Forecasting Using ARIMA Model,” J. Adv. Manag. Sci., vol. 4, no. 2, pp. 117–121, 2016, doi: 10.12720/joams.4.2.117-121.

[10] L. R. Jácome-Galarza, M. A. Realpe-Robalino, J. Paillacho-Corredores, and J. L. Benavides-Maldonado, “Time Series in Sensor Data Using State-of-the-Art Deep Learning Approaches: A Systematic Literature Review,”

in Smart Innovation, Systems and Technologies, 2022, vol. 252. doi: 10.1007/978-981-16-4126-8_45.

[11] H. Kim et al., “Exsolution of Ru Nanoparticles on BaCe0.9Y0.1O3-δ Modifying Geometry and Electronic Structure of Ru for Ammonia Synthesis Reaction Under Mild Conditions,” Small, vol. 19, no. 6, 2023, doi: 10.1002/smll.202205424.

[12] M. Alkaff, H. Khatimi, W. Puspita, and Y. Sari, “Modelling and predicting wetland rice production using support vector regression,” Telkomnika (Telecommunication Comput. Electron. Control., vol. 17, no. 2, pp. 819–825, 2019, doi: 10.12928/TELKOMNIKA.V17I2.10145.

[13] T. Bernatin, G. Sundari, S. A. Nisha, and M. S. Godwin Premi, “Optimized inter prediction for h.264 video codec,” Int. J. Electron. Telecommun., vol. 67, no. 2, 2021, doi: 10.24425/ijet.2021.135981.

[14] V. S. Suruthhi, V. Smita, J. Rolant Gini, and K. I. Ramachandran, “Detection and localization of audio event for home surveillance using CRNN,” Int. J. Electron. Telecommun., vol. 67, no. 4, 2021, doi: 10.24425/ijet.2021.139771.

[15] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, and T. M. Brenza, “Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells,” Chaos, Solitons and Fractals, vol. 146, 2021, doi: 10.1016/j.chaos.2021.110861.

[16] W. Wu, W. Liao, J. Miao, and G. Du, “Using gated recurrent unit network to forecast short-term load considering impact of electricity price,” in Energy Procedia, 2019, vol. 158. doi: 10.1016/j.egypro.2019.01.950.

[17] E. Haque, S. Tabassum, and E. Hossain, “A Comparative Analysis of Deep Neural Networks for Hourly Temperature Forecasting,” IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3131533.

[18] M. S. Islam and E. Hossain, “Foreign exchange currency rate prediction using a GRU-LSTM hybrid network,” Soft Comput. Lett., vol. 3, 2021, doi: 10.1016/j.socl.2020.100009.

[19] Z. Kong, B. Tang, L. Deng, W. Liu, and Y. Han, “Condition monitoring of wind turbines based on spatio-temporal fusion of SCADA data by convolutional neural networks and gated recurrent units,” Renew. Energy, vol. 146, 2020, doi: 10.1016/j.renene.2019.07.033.

[20] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, “Machine Learning-Based Approach for Hardware Faults Prediction,” IEEE Trans. Circuits Syst. I Regul. Pap., vol. 67, no. 11, 2020, doi: 10.1109/TCSI.2020.3010743.

[21] S. Mishra, C. Bordin, K. Taharaguchi, and I. Palu, “Comparison of deep learning models for multivariate prediction of time series wind power generation and temperature,” Energy Reports, vol. 6, 2020, doi: 10.1016/j.egyr.2019.11.009.

[22] Y. Fu, Z. Gao, Y. Liu, A. Zhang, and X. Yin, “Actuator and sensor fault classification for wind turbine systems based on fast fourier transform and uncorrelated multi-linear principal component analysis techniques,” Processes, vol. 8, no. 9, 2020, doi: 10.3390/pr8091066.

[23] X. Liu, Z. Lin, and Z. Feng, “Short-term offshore wind speed forecast by seasonal ARIMA - A comparison against GRU and LSTM,” Energy, vol. 227, 2021, doi: 10.1016/j.energy.2021.120492.

[24] D. Tukymbekov, A. Saymbetov, M. Nurgaliyev, N. Kuttybay, G. Dosymbetova, and Y. Svanbayev, “Intelligent autonomous street lighting system based on weather forecast using LSTM,” Energy, vol. 231, 2021, doi: 10.1016/j.energy.2021.120902.

[25] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, “Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management,” Int. J. Inf. Manage., vol. 57, 2021, doi: 10.1016/j.ijinfomgt.2020.102282.

[26] J. Luo, Z. Zhang, Y. Fu, and F. Rao, “Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms,” Results Phys., vol. 27, 2021, doi: 10.1016/j.rinp.2021.104462.

[27] N. AlDahoul et al., “Suspended sediment load prediction using long short-term memory neural network,” Sci. Rep., vol. 11, no. 1, 2021, doi: 10.1038/s41598-021-87415-4.

[28] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” PeerJ Comput. Sci., vol. 7, 2021, doi: 10.7717/PEERJ-CS.623.

[29] X. Zhang, Q. Zhang, G. Zhang, Z. Nie, Z. Gui, and H. Que, “A novel hybrid data-driven model for daily land surface temperature forecasting using long short-term memory neural network based on ensemble empirical mode decomposition,” Int. J. Environ. Res. Public Health, vol. 15, no. 5, 2018, doi: 10.3390/ijerph15051032.