TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

Mô hình dự báo và phát hiện các yếu tố gây ra mưa sử dụng thuật toán học sâu

Hồ Văn Lâm[1,*], Hoàng Thanh Minh[2], Lê Thị Phương Thảo[3]

[1]Khoa Công nghệ Thông tin, Trường Đại học Quy Nhơn, Việt Nam
[2]Đài Khí tượng Thủy văn tỉnh Gia Lai, Việt Nam
[3]Công ty TMA, Gia Lai, Việt Nam

Ngày nhận bài: 22/09/2025; Ngày sửa bài: 19/10/2025;
Ngày nhận đăng: 22/10/2025; Ngày xuất bản: 28/02/2026

TÓM TẮT

Nghiên cứu này sử dụng các thuật toán học sâu để xây dựng mô hình dự báo trên các tập dữ liệu thực tế có dấu hiệu mưa, nhằm dự đoán có mưa hay không tại một thời điểm cụ thể cũng như phân tích cách mưa xuất hiện dựa trên các yếu tố liên quan. Nghiên cứu cũng hướng đến hỗ trợ dự báo chính xác lượng mưa rơi xuống tại một địa điểm vào một thời điểm xác định. Trong nghiên cứu, chúng tôi xây dựng mô hình học sâu nhằm hỗ trợ dự báo thời tiết, đặc biệt là dự đoán chính xác lượng mưa - một bài toán luôn thách thức không chỉ đối với các cơ quan dự báo tại Việt Nam mà còn đối với các hệ thống dự báo tiên tiến trên thế giới. Sử dụng tập dữ liệu thu thập được, chúng tôi tiến hành mô tả các thuộc tính của các trường dữ liệu, cũng như phân tích các tham số có tương quan đến hiện tượng mưa. Sau đó, chúng tôi áp dụng thuật toán học sâu để xây dựng mô hình dự đoán khả năng có mưa có thể xảy ra hay không và xảy ra như thế nào? Các kết quả thu được có thể được ứng dụng trong thực tế để dự đoán lượng mưa tại một địa điểm và thời điểm cụ thể từ dữ liệu đầu vào là dữ liệu dấu hiệu mưa được trích xuất từ cơ sở dữ liệu dự báo thời tiết. Từ đó, nghiên cứu mở ra tiềm năng ứng dụng trí tuệ nhân tạo trong lĩnh vực dự báo khí tượng nhằm nâng cao độ chính xác và giảm thiểu rủi ro do thời tiết cực đoan gây ra.

Từ khóa: *Mô hình dự báo mưa, thuật toán LSTM, thuật toán RNN, thuật toán GRU, thuật toán học sâu.*

**Tác giả liên hệ chính.*
Email: hovanlam@qnu.edu.vn

https://doi.org/10.52111/qnjs.2026.20111

QUY NHON UNIVERSITY
JOURNAL OF SCIENCE

Predicting model and detecting factors causing rainfall using deep learning

Ho Van Lam[1,*], Hoang Thanh Minh[2], Le Thi Phuong Thao[3]

[1]*Faculty of Information Technology, Quy Nhon University, Vietnam*
[2]*Gia Lai Provincial Hydro-Meteorological Station, Vietnam*
[3]*TMA Company, Gia Lai, Vietnam*

Received: 22/09/2025; Revised: 19/10/2025;
Accepted: 22/10/2025; Published: 28/02/2026

ABSTRACT

This study aims to employ deep learning algorithms to construct predictive models using real-world datasets containing indicators of rainfall. The objective is to determine the occurrence of rainfall at a specific point in time and to analyze the underlying factors contributing to its onset. Furthermore, the research is directed toward improving the accuracy of quantitative rainfall prediction for a given location and time. This study has developed a deep learning-based framework for weather forecasting with a particular focus on accurate rainfall prediction - a task that remains highly challenging not only for meteorological agencies in Vietnam but also for state-of-the-art forecasting systems worldwide. Using the collected dataset, we conducted descriptive statistical analyses to characterize its properties and investigated the parameters exhibiting correlations with rainfall events. Based on these findings, deep learning algorithms were applied to develop a classification model capable of predicting the probability of rainfall occurrence. The experimental results demonstrate that the proposed model can be applied to operational scenarios for forecasting rainfall at specific locations and times, utilizing rainfall indicators extracted from meteorological forecast databases. The outcomes of this research highlight the potential of artificial intelligence techniques in meteorological applications, offering the prospect of enhanced prediction accuracy and reduced risks associated with extreme weather phenomena.

Keywords: *Rainfall prediction model, LSTM algorithm, RNN algorithm, GRU algorithm, deep learning algorithm.*

1. INTRODUCTION

One of the critical inputs for hydrological computation models is rainfall forecasting. Rainfall prediction is an inherently complex task, especially when forecasting for specific locations across different months and seasons. To develop a low-cost yet effective method that delivers acceptable forecasting accuracy, we employed machine learning techniques to build a daily rainfall forecasting model. Unlike traditional approaches, this study utilized datasets collected from monitoring stations, combining observed attributes with ERA5 reanalysis data, and applied suitable deep learning algorithms to construct models for rainfall prediction and related influencing factors. In this paper, we present a rainfall forecasting model developed using 16 years of data collected from monitoring stations and ERA5 reanalysis datasets. The forecast outputs from this model can support decision-making in operational forecasting and other related tasks at monitoring station locations.

Artificial Intelligence (AI) is playing an increasingly important role in meteorology and

**Corresponding author.*
Email: hovanlam@qnu.edu.vn

https://doi.org/10.52111/qnjs.2026.20111

hydrology due to its capability to process large volumes of data from observation stations, forecasts, and historical weather records. Deep Learning, a subset of AI, employs multi-layer neural networks to learn complex patterns from data and construct predictive models.

In this study, we developed deep learning models based on the Long Short-Term Memory (LSTM) architecture to predict the occurrence and probability of rainfall. Model optimization was performed through the analysis of evaluation metrics such as the confusion matrix, ROC-AUC curve, and Precision–Recall curve, alongside the identification of key variables influencing predictive performance.

We also integrated meteorological data from observation stations with deep learning algorithms to construct a rainfall forecasting model that can assist meteorologists in their forecasting tasks and be transferable to other stations when necessary. By combining meteorological expertise with observational datasets, the model can analyze factors influencing rainfall based on meteorological parameters, thereby providing predictions on rainfall occurrence and the expected rainfall intensity.

2. RAIN FORECASTING PROBLEMS

2.1. Rain forecasting problem

Currently, accurately predicting rainfall at a specific location and time remains a significant challenge for meteorological agencies worldwide.

Rainfall is essentially the result of atmospheric processes in which water vapor in the atmosphere undergoes a phase change (condensation) into solid or liquid forms such as water, ice, or snow and falls to the ground under the influence of gravity. During the process of condensation and descent to the ground, raindrops are affected by horizontal air currents. Due to differences in environmental conditions, the raindrops themselves may partially evaporate during their fall.[1]

In recent years, meteorology and hydrology have made significant progress in forecasting large-scale heavy rainfall events. Such phenomena can be predicted 2–3 days in advance with an accuracy of about 70%, and in some cases, early warnings can be issued 5-7 days ahead. Forecast information for large-scale heavy rainfall events is generally reliable regarding the timing of rainfall onset, the affected areas, and the ending time of the event.

Early forecasting of large-scale heavy rainfall plays a crucial role in supporting flood, flash flood, landslide, and inundation warnings. These alerts are communicated to authorities and the public to enable proactive response planning and minimize damage.

However, when it comes to quantitative rainfall forecasts (for specific locations and times), current numerical weather prediction technology still faces many limitations. Notably, there are constraints in spatial resolution due to the use of numerous empirical parameters in physical models, as well as a shortage of observational input data particularly over oceans and at higher atmospheric layers.

Estimates indicate that the reliability of point-based quantitative forecasts within a 1–3 day range is only about 40–60% for light and moderate rainfall events (less than 16 mm/day).

In addition to improving the physical modeling capabilities of forecasting systems, the meteorological sector also focuses on enhancing the training and expertise of forecasters especially in utilizing intelligent decision-support systems. This allows for the integration of various types of observations and forecast products, enabling fine-tuning of rainfall and temperature predictions, as well as leveraging ensemble forecasting and other decision-support tools.[2,3]

2.2. Rainfall database

In this study, we use data from the Quy Nhon Meteorological Station a Class I meteorological station with the international code “48870”. This station is internationally recognized as a high-accuracy data source and is frequently used in weather forecasting models.

The dataset spans from 2009 to 2024 and includes hourly observational variables such as temperature, humidity, station pressure, and total rainfall. In addition, reanalysis data from ERA5 is incorporated, comprising 54 variables, primarily related to temperature, humidity, and wind vectors in the u and v directions at atmospheric pressure levels ranging from 950 hPa to 300 hPa. All data are organized as time series by hour, day, and month.

The objective of using these datasets is to explore and analyze the relationship between rainfall and other meteorological factors in the Quy Nhon area. Rainfall classification in the dataset follows the standards of the Vietnam Meteorology and Hydrology sector as follows:

- No rain = 0mm/day
- Rain < 16mm/day
- 16mm/day ≤ Moderate rain < 50mm/day
- 50mm/day ≤ Heavy rain < 100mm/day
- Very heavy rain ≥ 100mm/day

Statistics show that "No rain" has the highest occurrence with 86601 cases, followed by "Rain" with 40752 cases, and then "Moderate rain" with 9157 cases. "Heavy rain" is significantly less frequent with 2476 cases, and finally, "Very heavy rain" has the fewest occurrences with 1192 cases.

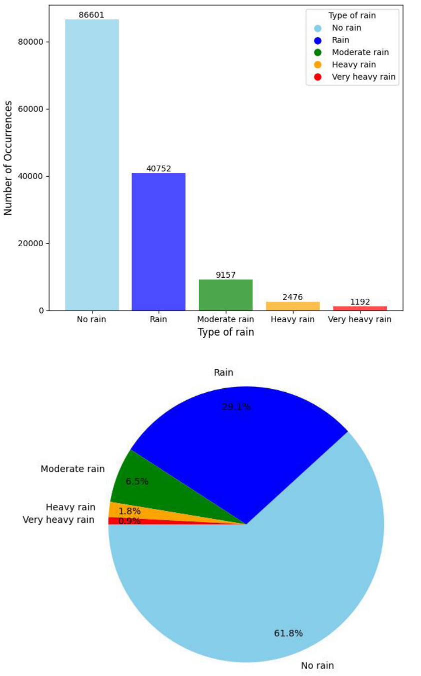

Figure 1. The distribution of rainfall categories in the dataset.

From Figure 1, it is evident that among 100 sampled values, there are 62 "No rain" cases, 29 "Rain" cases, 6 "Moderate rain" cases, 2 "Heavy rain" cases, and only 1 "Very heavy rain" case. This indicates an uneven distribution of data among rainfall categories, with heavier rainfall events occurring less frequently.

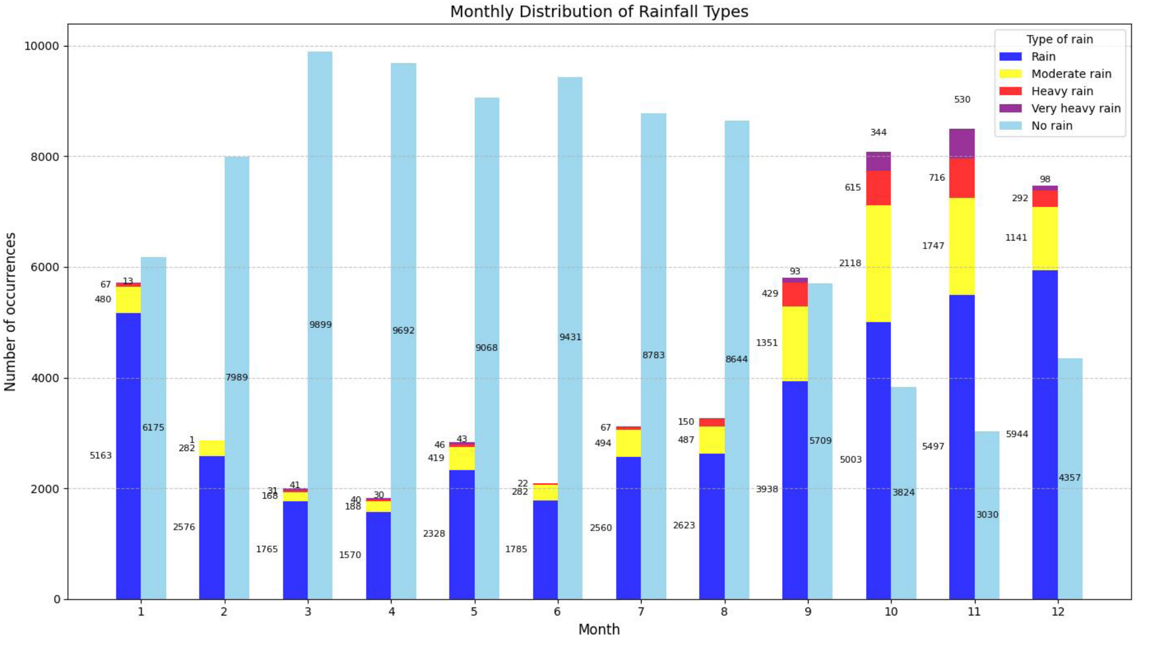

Figure 2. The distribution of rainfall categories by month.

https://doi.org/10.52111/qnjs.2026.20111

Based on Figure 2, heavy and very heavy rainfall events are primarily concentrated between September and December, with a pronounced increase in their occurrence during September, October, and November. This period corresponds to the region's main rainy season, when active weather systems deliver abundant precipitation. In these months, not only does the number of rainfall events rise significantly, but rainfall intensity also increases, contributing substantially to the region's annual total precipitation.

Specifically, the number of moderate, heavy, and very heavy rainfall events increases sharply from September to December, whereas the rest of the year is dominated by no rain or rain events. This highlights a clear seasonal pattern in rainfall distribution within the study area.

As shown in Figure 3, rainfall events occur most frequently in the temperature range of 24°C to 30°C, with particularly high concentrations in the 24–26°C and 26–28°C intervals. In these temperature ranges, the total number of rainfall samples (from light to very heavy) accounts for the majority compared to other temperature groups. Notably, very heavy rainfall events almost exclusively occur within the 24-26°C and 26-28°C intervals, indicating that this temperature range is the most favorable for extreme rainfall. Conversely, at lower temperatures (< 22°C) or higher temperatures (> 30°C), the frequency of rainfall events especially heavy and very heavy declines sharply, with almost no extreme rainfall observed above 30°C. This suggests that samples with excessively low or high temperatures are less likely to be associated with rainfall, particularly intense rainfall events.

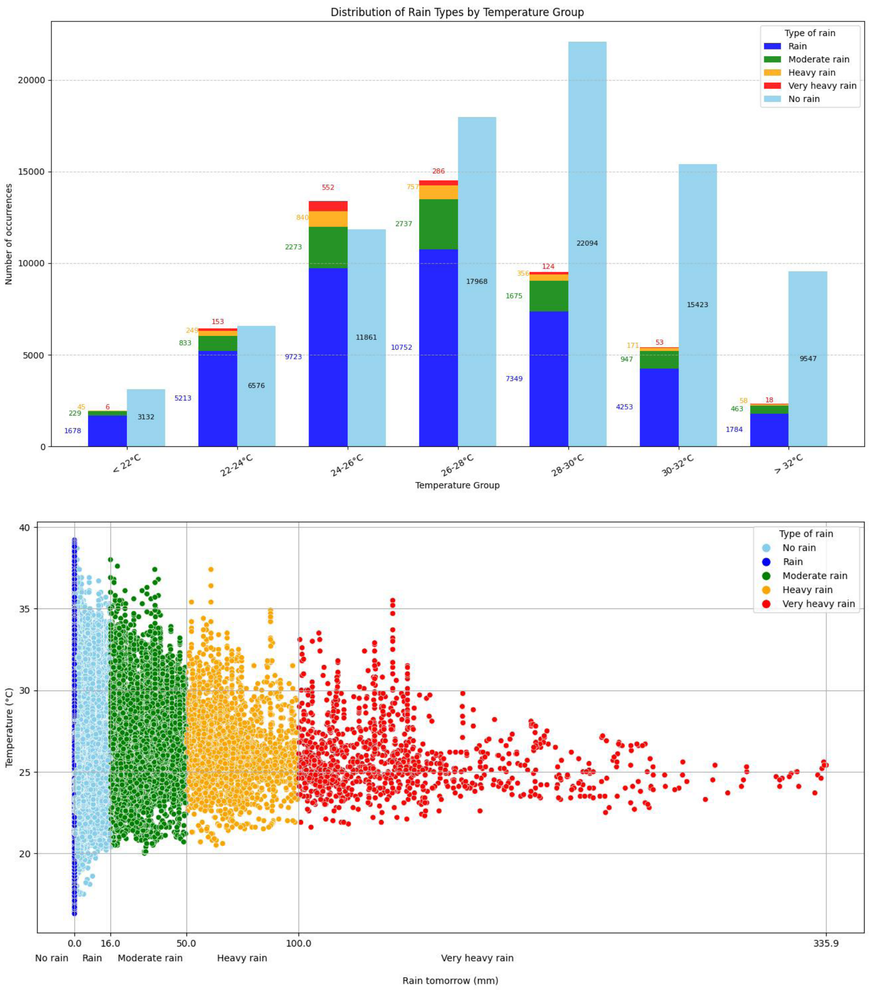

Figure 3. The distribution of rainfall types across temperature groups.

Figure 4 shows that the frequency of rainfall events (from light rain to very heavy rain) increases with higher humidity levels. In the 80-90% humidity range, the total number of rainy samples is the highest, with light rain and moderate rain dominating. This indicates that this humidity band is ideal for rain formation. The 70-80% range comes next, also showing a relatively large number of rainy samples, reflecting the clear trend that higher humidity is associated with a higher likelihood of rain.

Notably, heavy and very heavy rainfall events occur mainly in the two highest humidity groups 80–90% and >90% and are almost absent in lower humidity groups. This suggests that extreme rainfall events often happen when the air holds a very high moisture content, providing favorable conditions for intense atmospheric condensation. In contrast, humidity groups below 60% and 60-70% record relatively few rainy samples, with heavy and very heavy rain almost non-existent, indicating that drier environments have little potential to produce rainfall, especially extreme events. Moreover, the >90% humidity group is the only one with the highest number of very heavy rain samples in the entire chart, emphasizing the role of extreme humidity in triggering severe weather phenomena.

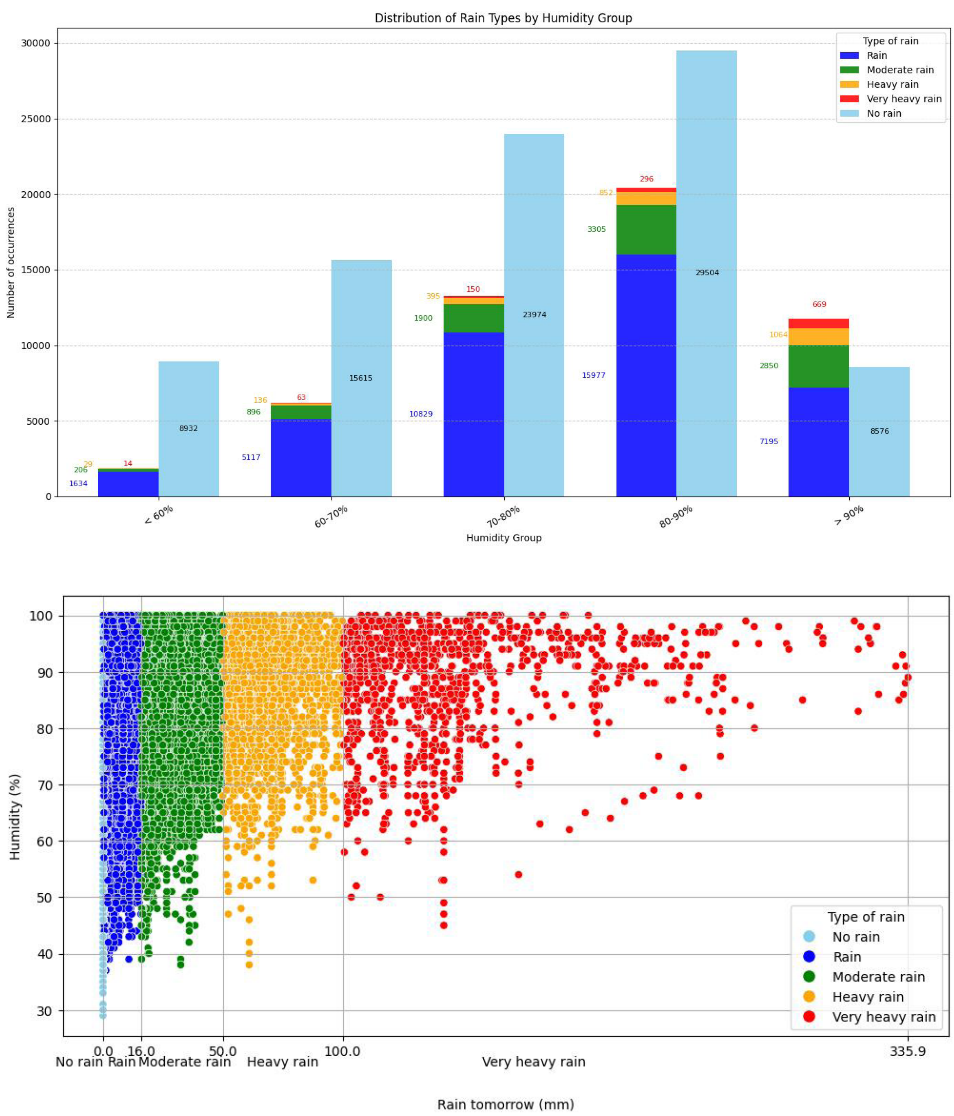

Figure 4. Distribution of rainfall types by humidity group.

In Figure 5, rainfall samples are concentrated mainly in the pressure range of 1005-1015 mb, with the 1005-1010 mb and 1010-1015 mb groups clearly dominating. This range not only shows a high number of rainy samples but also a noticeable increase in strong rainfall events, reflected in the frequent appearance of orange and red bars representing heavy and very heavy rain. This suggests that this pressure range is favorable for atmospheric conditions that lead to the formation and growth of convective rain clouds.

On the other hand, at the extremes of pressure specifically <1000 mb and >1025 mb the number of rainy samples is very low, and heavy rainfall events are almost absent, indicating that both very low and very high pressure are not ideal environments for rain. The 1015-1020 mb group still maintains a considerable number of rainy samples but shows a slight decrease compared to the preceding range, suggesting that when pressure exceeds 1015 mb, the likelihood of rain begins to decline. Similarly, the 1020-1025 mb and >1025 mb groups are dominated by non-rain samples, reflecting a more stable atmosphere with fewer conditions supporting rainfall development.

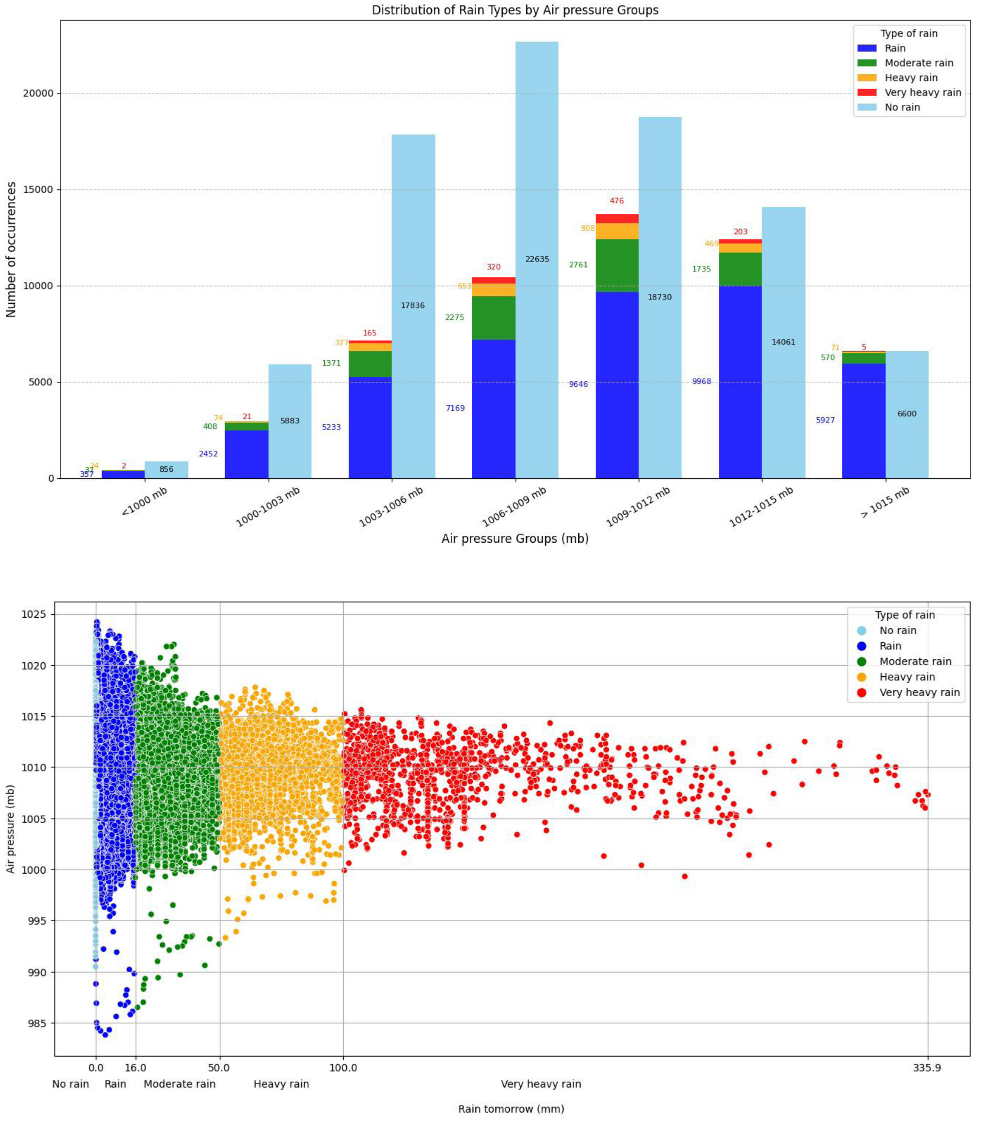

Figure 5. Distribution of rainfall types by pressure group.

Because the features in the dataset are independent, analyzing their correlations is essential to assess both their interrelationships and their relationship with the target variable in this case, the likelihood of rain the next day (rain_tomorrow).

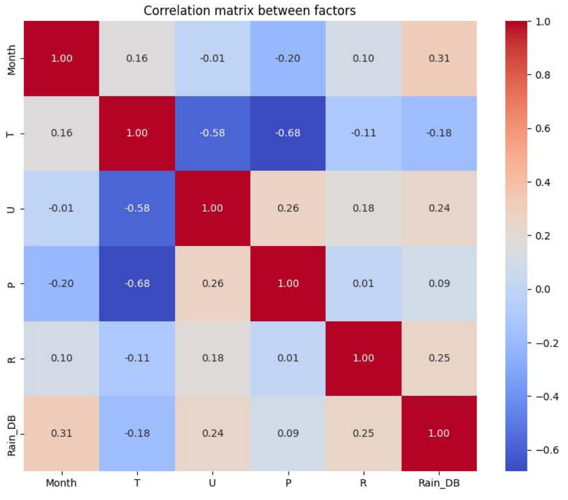

Figure 6. Correlation matrix between factors.

From Figure 6, we observe that humidity, month, and rainfall all show positive correlations with the probability of rain on the following day, with previous-day rainfall exhibiting the strongest positive correlation with next-day rain. In contrast, factors such as temperature have a negative correlation with the likelihood of rain. Notably, atmospheric pressure shows a very weak correlation with next-day rain, with a coefficient of only 0.09.

3. ALGORITHMS AND PREDICTION MODELS

3.1. Deep learning algorithms

Deep learning is an important branch of artificial intelligence that focuses on building and training multi-layer neural networks to automatically learn complex features from data.

Unlike traditional machine learning methods, deep learning can extract features directly from raw data, reducing dependence on manual preprocessing steps while effectively capturing complex nonlinear relationships between input variables.

Thanks to these capabilities, deep learning has become a powerful tool in fields that require processing large and complex datasets, such as computer vision, natural language processing, and especially time series forecasting in meteorology and hydrology.[5]

In the context of weather forecasting and hydrometeorological phenomena, deep learning algorithms are widely applied to predict variables related to rainfall, temperature, humidity, pressure, and other meteorological parameters.

Sequential neural networks such as RNNs (Recurrent Neural Networks) allow the model to retain information from previous time steps, while more advanced variants such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are specifically designed to address the vanishing gradient problem, enabling the learning of long-term dependencies in time series data.[6]

The choice of an appropriate deep learning algorithm depends on the specific characteristics of the problem and the data.

For example, with datasets containing long time series and requiring the capture of complex relationships among meteorological variables, LSTM is often preferred for its long-term memory capabilities, while GRU can be used when reducing the number of parameters and speeding up training is a priority. Thus, deep learning not only offers more accurate forecasting but also provides flexibility in uncovering hidden features in hydrometeorological data.

RNN Algorithm

RNN (Recurrent Neural Network) is a neural network architecture specifically designed to process time series data, where the current value depends on previous values. Unlike traditional neural networks, RNNs have the ability to retain information from previous time steps through a hidden state, enabling the model to predict future values based on historical data.[5]

RNN Training Algorithm: RNNs are trained using Backpropagation Through Time (BPTT), an extension of backpropagation, to update weights based on the gradient of the loss function with respect to the entire time sequence:[4]

Step 1: Weight Initialization - Randomly initialize the weights W_h (hidden state weights), W_x (input weights), W_y (output weights) along with biases b (hidden state bias) and c (output bias).

Step 2: Forward pass - Iterate through the entire time sequence. At each time step t, compute the hidden state h_t based on the current input x_t and the previous hidden state $h_{t\text{-}1}$ according to the formula:

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b) \quad (1)$$

Then, compute the predicted output y_t from the hidden state h_t:

$$y_t = \text{softmax}\big(W_y h_t + c\big) \quad (2)$$

Step 3: Compute the loss function - Use an appropriate loss function based on the predicted output y_t and the actual label y^t

Step 4: Backward pass – Backpropagation of the error from the final time steps to the initial ones, computing the gradient of the loss function with respect to the weights W_x, W_h, W_y, b, c.

Step 5: Update weights - Use an optimization algorithm to update the weights based on the computed gradients, minimizing the loss function.

Step 6: Repeat - The process of forward pass, loss calculation, backward pass, and weight updates is repeated over many epochs until the model converges or meets the early stopping criterion.

Step 7: Prediction - Once the model is trained, the RNN can take a new input sequence and continuously compute the hidden states to predict the corresponding output sequence.

LSTM Algorithm

LSTM (Long Short-Term Memory) is an improved recurrent neural network (RNN) architecture designed to handle long time-series data and overcome the vanishing gradient problem often found in traditional RNNs. LSTM can retain long-term information thanks to its gating mechanism, which controls which information is kept, updated, or discarded in the cell state.[5,7]

LSTM training algorithm: LSTM is also trained using Backpropagation Through Time, an extension of backpropagation, to update weights based on the gradient of the loss function over the entire time series.

Step 1: Initialize weights and states - Randomly initialize the weights for the forget gate W_f, input gate W_i, output gate W_o, cell input W_c along with the biases b_f, b_i, b_o, b_c. The hidden state h_0 and the cell state C_0 are usually initialized as zero vectors.

Step 2: Forward pass - Iterate through the entire time series. At each time step t:
Forget gate:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \quad (3)$$

Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

Cell input:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

Update cell state:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (6)$$

Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

Hidden state:

$$h_t = o_t * \tanh(C_t) \quad (8)$$

Output prediction (if needed):

$$y_t = \text{output\_layer}(h_t) \quad (9)$$

Step 3: Compute the loss function - Use an appropriate loss function based on the predicted output y_t and the actual label.

Step 4: Backward pass – Backpropagation of errors from the last time step to the first (Backpropagation Through Time), computing the gradients of the loss with respect to all weights W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c.

Step 5: Update weights - Use an optimization algorithm to update weights based on computed gradients, minimizing the loss function.

Step 6: Repeat - Perform forward pass, loss computation, backward pass, and weight updates over many epochs until the model converges or meets early stopping criteria.

Step 7: Prediction - Once trained, the LSTM can take a new input sequence and compute hidden states sequentially to predict the corresponding output sequence.

GRU Algorithm

GRU (Gated Recurrent Unit) is an improved recurrent neural network architecture, similar to LSTM but with a simpler structure. It combines certain gates to reduce the number of parameters while still maintaining the ability to remember long-term information. GRU has two main gates: the update gate and the reset gate, which control which information should be retained or discarded in the hidden state.

The GRU training algorithm also uses Backpropagation Through Time to update weights based on the gradients of the loss function across the entire time sequence.

Step 1: Initialize weights and states - Randomly initialize the weights for the update gate W_z, reset gate W_r, candidate state W_h along with the biases b_z, b_r, b_h. The initial hidden state h_0 is usually set as a zero vector.

Step 2: Forward pass - Iterate through the entire time sequence. At each time step t:

- Update gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{10}$$

- Reset gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{11}$$

- Candidate hidden state:

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \tag{12}$$

- New hidden state:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{13}$$

- Output prediction (if needed):

$$y_t = \text{output\_layer}(h_t) \tag{14}$$

Step 3: Compute the loss function - Use an appropriate loss function based on the predicted output y_t and the actual labels.

Step 4: Backward pass – Backpropagation of the error from the last time steps to the first, computing the gradients of the loss function with respect to all weights W_z, W_r, W_h, b_z, b_r, b_h.

Step 5: Update weights - Use an optimization algorithm to update the weights based on the computed gradients, minimizing the loss function.

Step 6: Repeat - The process of forward pass, loss computation, backward pass, and weight update is repeated for many epochs until the model converges or meets an early stopping criterion.

Step 7: Prediction - Once trained, the GRU can take a new input sequence and compute the hidden state continuously to predict the corresponding output sequence.

3.2. Prediction models using deep learning

The dataset from the Quy Nhon Meteorological Station, after being cleaned and encoded to convert categorical features into numerical values, can be used as input for deep learning models.The objective is to train and compare the performance of three deep learning algorithms: LSTM, RNN, and GRU. These models are highly suitable for time series data and have proven effective in weather forecasting tasks thanks to their ability to capture temporal dependencies, automatically extract features from raw data, and model complex nonlinear relationships between variables.[4,9]

The processed dataset was divided into two subsets, with 80% allocated for training and 20% reserved for testing. This partitioning strategy ensures that the models have sufficient data for effective learning while maintaining an independent subset for objective evaluation of generalization capability on unseen samples.

The input dataset consists of 54 meteorological variables derived from ERA5 reanalysis data including temperature, humidity, wind, and pressure at various atmospheric levels combined with surface observational time series from the Quy Nhon meteorological station. This integration captures both local-scale conditions and large-scale atmospheric circulation patterns, providing a comprehensive representation of the meteorological environment. The prediction target is the 24-hour accumulated rainfall, categorized into discrete rainfall intensity levels.

For efficient model training, rainfall observations were discretized into five distinct categories: (0) no rain, (1) rain, (2) moderate rain, (3) heavy rain, and (4) very heavy rain. These categorical labels were subsequently encoded using one-hot encoding to form five-dimensional binary vectors, each representing a specific rainfall class.

Following model architecture design, parameter optimization was conducted through systematic tuning of the training algorithm, loss function, and key hyperparameters. Specifically, all models employed the Adam optimizer with an initial learning rate of $\alpha = 0.001$. Adam, which combines the advantages of Momentum and RMSProp, dynamically adapts the learning rate during training, making it particularly effective for nonlinear and noisy meteorological datasets.[8]

The Categorical Cross-Entropy loss function was adopted as it is well-suited for multi-class classification tasks, enabling the models to learn class probability distributions rather than rigid decision boundaries.[8] The training process was executed over 70 epochs with a batch size of 64, a configuration that balances learning efficiency and overfitting prevention. This setup ensures stable gradient updates, effective utilization of computational resources, and robust model convergence, ultimately leading to improved predictive performance on independent test data.

Table 1. Comparison among the machine learning model.

Model	Accuracy	Kappa	AUC-ROC
LSTM	0.8211	0.6462	0.9544
RNN	0.7675	0.5254	0.9196
GRU	0.8115	0.6290	0.9514

Table 1 presents the performance comparison of the three deep learning models—LSTM, RNN, and GRU—in a multi-class classification problem. The results show that LSTM achieved the highest performance across all three evaluation metrics, with an Accuracy of 0.8211, a Kappa of 0.6462, and an impressive AUC-ROC of 0.9544. This indicates that LSTM not only delivers high predictive accuracy but also effectively distinguishes between classes, especially in the context of complex data.

GRU also demonstrated competitive performance, with an Accuracy of 0.8115, a Kappa of 0.6290, and an AUC-ROC of 0.9514 only slightly lower than LSTM. This suggests that GRU is an efficient choice when balancing accuracy and computational complexity is important.

In contrast, RNN achieved significantly lower results, with an Accuracy of only 0.7675, a Kappa of 0.5254, and an AUC-ROC of 0.9196. This gap is particularly evident in the Kappa score, indicating that RNN struggles to accurately classify classes when dealing with imbalanced data. Notably, while the AUC-ROC scores of all three models exceed 0.9, the differences in Accuracy and Kappa indicate that LSTM and GRU handle the task more effectively than the traditional RNN.

These results suggest that modern neural network architectures such as LSTM and GRU are better suited for multi-class classification problems than basic RNNs, due to their ability to capture long-term dependencies in time series data.

Table 2. Regression error comparison of neural network models.

Model	RMSE	MAE
LSTM	0.5024	0.2012
RNN	0.5766	0.2617
GRU	0.5106	0.2096

Table 2 presents the RMSE and MAE for three neural network models: LSTM, RNN, and GRU. The LSTM achieved the lowest errors (RMSE = 0.5024, MAE = 0.2012), followed by the GRU (RMSE = 0.5106, MAE = 0.2096), while the RNN exhibited the highest errors (RMSE = 0.5766, MAE = 0.2617). The average deviation between predicted and true labels is approximately 0.2–0.5 units on the 0–4 scale, indicating that all models reliably forecast rainfall categories. Overall, models with gating mechanisms (LSTM and GRU) outperform the standard RNN in capturing temporal dependencies and reducing prediction errors.

Confusion Matrix

Analysis of the confusion matrix in Figure 7 for the three deep learning models (LSTM, RNN, GRU) in the rainfall classification task (0: no rain, 1: rain, 2: moderate rain, 3: heavy rain, 4: very heavy rain) reveals distinct patterns.

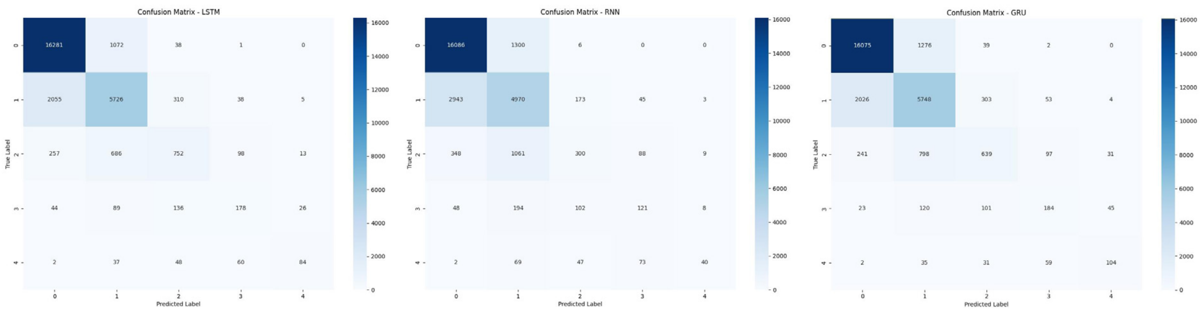

Figure 7. Confusion Matrices for LSTM, RNN and GRU Models.

For the LSTM model, classification performance decreases as rainfall intensity increases: it achieves the highest accuracy with 16281 correctly classified cases for the no-rain class (0), drops to 5726 for the light rain class (1), and only 84 correct predictions for the very heavy rain class (4). Notably, the model tends to confuse rainfall classes, as evidenced by 37 very heavy rain cases (4) being misclassified as light rain (1) and 2 cases as no rain (0). This outcome reflects a common challenge for models when dealing with minority classes that occur infrequently in the dataset, especially extreme rainfall events. The large disparity in the number of correct predictions between the majority class (no rain) and the minority classes (various rainfall types) highlights the need to apply data imbalance handling techniques to improve the model's overall performance.

The RNN model demonstrates the most unstable performance among the three, with gaps in its confusion matrix. Although it achieves 16086 correct predictions for the no-rain class (0), it misclassifies up to 4900 cases. Its performance drops sharply for rainfall classes, with only 4970 correct predictions for light rain (1) and 102 for very heavy rain (4).

The GRU model shows inconsistent classification performance across classes. For the no-rain class (0), it performs well with 16075 correct predictions, but a wide range of abnormal values (1276–5748) suggests potential overfitting or issues in the data normalization process. Performance declines significantly for rainfall classes: the moderate rain class (2) achieves only 1216 correct predictions, heavy rain (3) drops to 398, and the very heavy rain

https://doi.org/10.52111/qnjs.2026.20111

class (4) is almost entirely unrecognized, with correct predictions ranging from 0–4. This steep decline in accuracy with increasing rainfall intensity, along with extremely low values at the bottom of the matrix, points to limitations in input data quality or a suboptimal preprocessing pipeline, especially for extreme rainfall cases.

Overall, all three models struggled to accurately classify minority classes, with LSTM producing the most stable results despite being imperfect, while RNN and GRU exhibited several anomalies that require further inspection of data quality and training processes. The large disparity in the number of predictions across classes highlights the issue of data imbalance.

Training progress - Loss log

Analysis of the learning curves (Figure 8) shows a clear difference in the training processes of the three models: LSTM, RNN, and GRU. All three display a general downward trend in loss as the number of epochs increases, but with distinct characteristics.

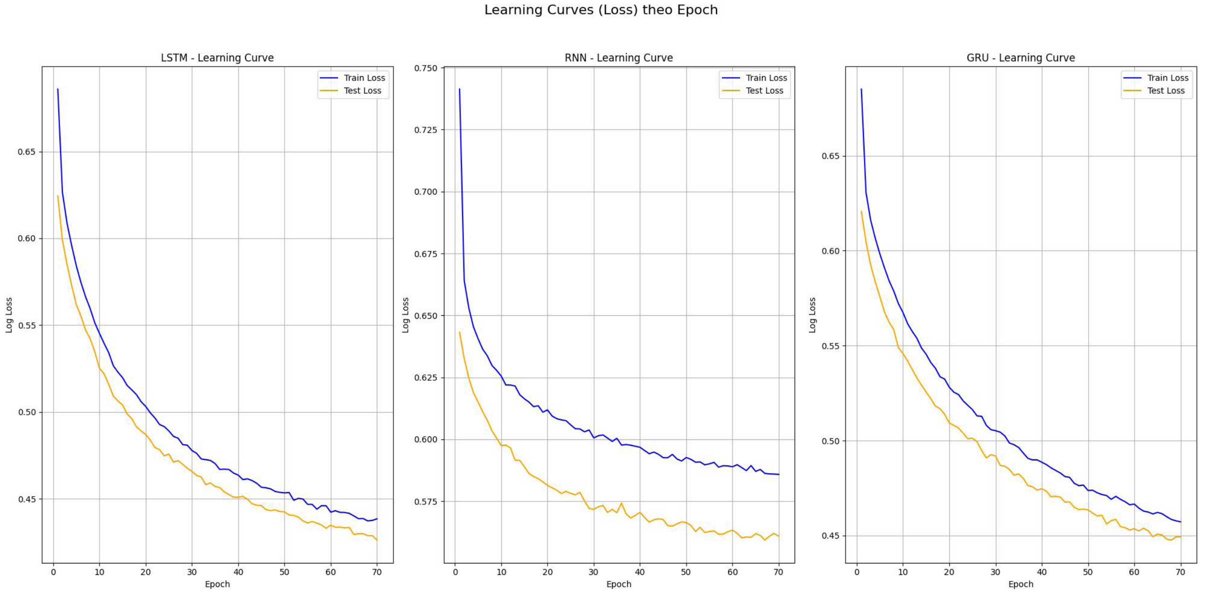

Figure 8. Training progress and loss log for LSTM, RNN, and GRU.

The LSTM model shows the most stable learning curve, with test loss decreasing gradually from 0.75 to 0.575 after 70 epochs, indicating a slow yet steady learning process. Notably, the gap between train loss and test loss is relatively small, suggesting that the model suffers little from overfitting.

RNN improves more quickly in the early stages, with test loss dropping from 0.65 to 0.45 in the first 30 epochs. However, it later exhibits strong fluctuations (especially between epochs 40–50), reflecting the inherent instability of the traditional RNN architecture.

GRU demonstrates a balance between the two above learning faster than LSTM but with more stability than RNN. Its test loss decreases steadily from 0.65 to 0.30. However, the sudden drop in test loss at epoch 20 (from 0.60 to 0.35) followed by subsequent oscillations may point to optimization issues or an unsuitable learning rate.

In summary, LSTM shows an advantage in stability, GRU learns faster but is less stable, and RNN struggles to maintain consistent performance across epochs. These results align with theory: LSTM is designed to address the vanishing gradient problem faced by traditional RNNs, while GRU is a simplified version of LSTM with fewer parameters.

Feature Importance

Analysis of Figure 9 shows the contribution levels of input features for three deep learning models: LSTM, RNN, and GRU, highlighting the 15 most important features for each model. The importance scores were determined using the permutation importance method, which measures the decrease in model accuracy when the values of a particular feature are randomly shuffled. A greater drop in accuracy indicates higher importance, reflecting the extent to which each meteorological variable influences the model's predictive performance.

For LSTM, temperature-related factors at various pressure levels, such as t_750hPa, t_950hPa, t_300hPa, and the Month variable, rank at the top, indicating that this model focuses on temperature variations across both time and atmospheric height. Additionally, humidity features like e_700hPa and wind components such as v_600hPa also play significant roles.

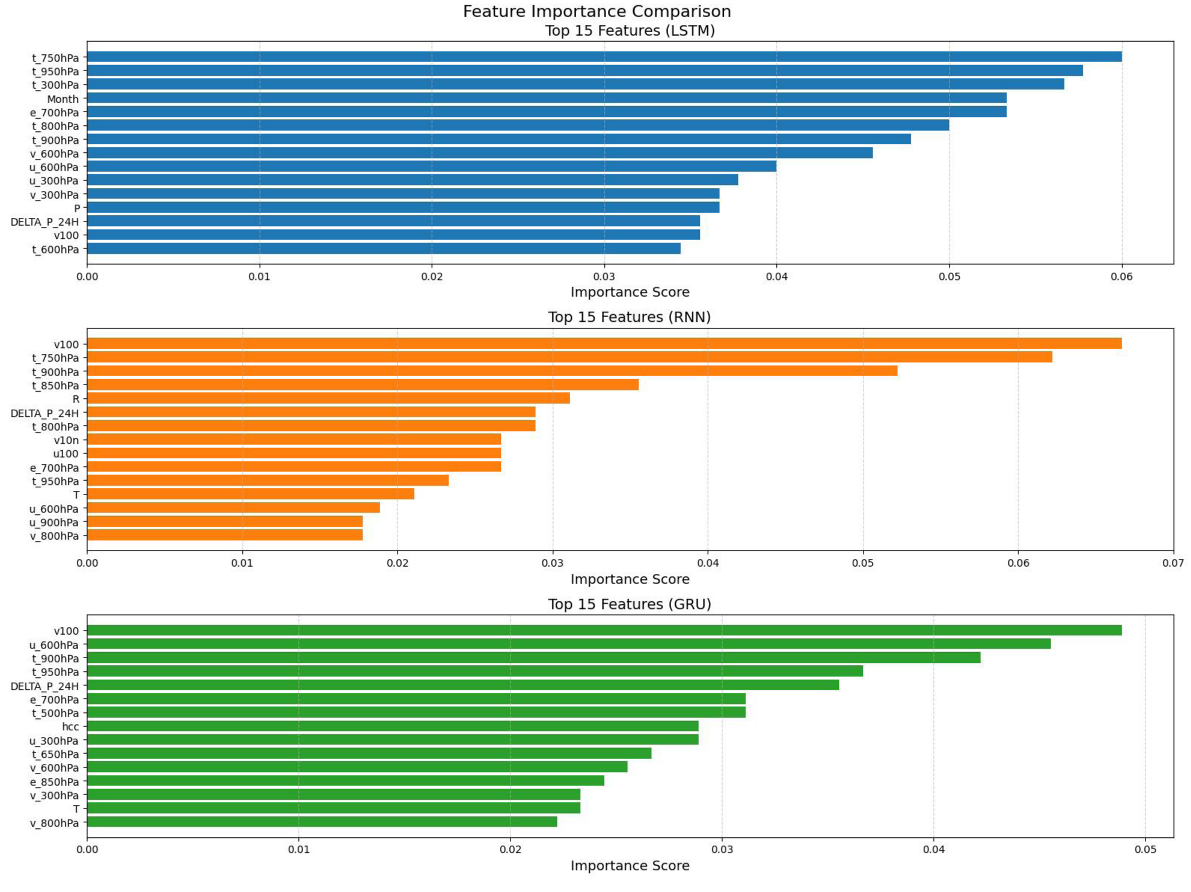

Figure 9. Importance of features.

For RNN, features related to surface and lower-level wind, such as v100 and v10n, along with mid and low-level temperatures (t_750hPa, t_900hPa) dominate. Notably, actual rainfall (R) and 24-hour pressure variation (DELTA_P_24H) are among the top features, reflecting the RNN's ability to strongly leverage local and direct weather signals.

Meanwhile, GRU prioritizes v100 and u_600hPa (zonal wind at 600hPa), combined with temperatures at multiple levels (t_900hPa, t_750hPa, t_650hPa) and mid-level humidity (e_700hPa). The 24-hour pressure variation (DELTA_P_24H) is also among the important features, highlighting the role of large-scale dynamic factors.

Overall, all three models utilize a combination of temperature, wind, and pressure information, but LSTM emphasizes multi-level temperature analysis, RNN is more sensitive to surface wind and rainfall signals, while GRU balances wind, temperature, and pressure. These differences reflect the distinct strategies each architecture employs in rainfall prediction.

4. CONCLUSION

The study applied three deep neural network architectures LSTM, RNN, and GRU to classify rainfall by intensity, using observational data from the Quy Nhon meteorological station combined with ERA5 reanalysis data. The results indicate that all three models achieved good forecasting performance, with LSTM demonstrating superior accuracy, class discrimination ability, and training stability. GRU performed closely to LSTM, offering a balanced choice between accuracy and computational cost, while the traditional RNN was less stable and faced limitations in classifying minority classes.

Feature importance analysis revealed that the temporal factor (Month) plays a prominent role, clearly reflecting the seasonal nature of rainfall in the study area. In addition, meteorological features such as wind, temperature, humidity, and pressure variations at multiple atmospheric levels also contributed significantly, with each model tending to exploit information differently: LSTM focused on multi-level temperature variations, RNN was more sensitive to surface wind and rainfall signals, and GRU maintained a balance between wind, temperature, and pressure factors.

However, there exist a number of practical limitations in this study. Although ERA5 reanalysis data supplement large-scale information, they exhibit a time lag in updates, causing the model to rely primarily on surface observations and limiting its ability to respond rapidly to short-term atmospheric fluctuations. Moreover, model performance may decline over time if not periodically updated, due to climate change and evolving structures of meteorological variables. Therefore, regularly incorporating new data, retraining the model, and integrating real-time observations with multi-layer forecasting models and advanced deep learning techniques will be crucial for improving accuracy, adaptability, and resilience to extreme weather event.

Acknowledgment

This research is conducted within the framework of science and technology projects at institutional level of Quy Nhon University under the project code T2025.898.18.

REFERENCES

1. M. P. Plummer. Rainfall formation and precipitation microphysics, *Atmospheric Research*, **2017**, *183*, 12-24.
2. J. Smith, A. Brown, T. Nguyen. Artificial intelligence and numerical weather prediction models: a technical survey, *Science Direct*, **2024**, *1*, 1-3.
3. M. Johnson, H. Lee. Quantitative precipitation forecasting using an improved weighted moving average probability-matching method, *Atmosphere (MDPI)*, **2023**, *12*, 1346.
4. Y. LeCun, Y. Bengio, G. Hinton. Deep learning, *Nature*, **2015**, *521*, 436-444.
5. S. Hochreiter, J. Schmidhuber. Long short-term memory, *Neural Computation*, **1997**, *9*, 1735-1780.
6. C. M. Bishop. *Pattern recognition and machine learning*, Springer, USA, 2006.
7. Y. Yu, X. Si, C. Hu, J. Zhang. A review of recurrent neural networks: LSTM cells and network architectures, *Neural Computation*, **2019**, *31*, 1235-1270.
8. I. Goodfellow, Y. Bengio, A. Courville. *Deep learning*, MIT Press, Cambridge, USA, 2016.
9. O. A. Wani, S. A. Bhat, S. Ahmad, J. A. Sofi, K. Ahmad, A. Q. Malik, S. A. Romshoo. Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas, *Scientific Reports,* **2024,** *14*, 27876.