Application of LSTM, GRU, and Transformer Models for Precipitation Forecasting

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Accurate rainfall prediction is critical for improving agricultural productivity in the Midwest, where farmers rely on precise forecasts to optimize planting schedules, manage irrigation systems, and prepare for extreme weather events. This study develops and compares three advanced deep learning models - Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformer models - to predict daily and weekly precipitation. Historical weather data from 1970-2020 was preprocessed to handle missing values, normalize features, and address class imbalances caused by the predominance of zero precipitation days. The LSTM model served as a baseline due to its capability to model temporal dependencies, while GRU and Transformer models were assessed for their efficiency and ability to capture complex precipitation patterns. Model performance was evaluated using a custom Large Valuation Accuracy metric, with the Transformer model achieving an accuracy of 90.80%, outperforming both the LSTM and GRU models. These results highlight the potential of Transformer-based architectures in tackling the challenges of precipitation forecasting, particularly in regions with sparse rainfall data.

1 Introduction

Rainfall prediction is crucial for agriculture, particularly in dry climates where even minimal precipitation can significantly impact farming operations. While substantial research has focused on predicting rainfall in wetter regions, there is a notable gap in studies for drier climates, which presents a challenge for farmers in arid areas like the Midwestern United States, who require accurate precipitation forecasts to optimize productivity.

Existing methods, particularly Long Short-Term Memory (LSTM) models, have been successful in wetter areas but face challenges in drier climates due to the high prevalence of days with no precipitation, creating class imbalances. This imbalance limits the LSTM's ability to generalize, leading to overfitting and poor predictions for rare, but critical, rainfall events.

This study aims to address these challenges by expanding rainfall prediction research to drier climates using historical weather data from the Midwestern United States. After establishing a baseline with LSTM models, we explore Gated Recurrent Units (GRU) and Transformer models as alternatives to better handle class imbalances and improve generalization for both low and high precipitation days. This work seeks to enhance precipitation forecasting for agricultural planning in regions with scarce but vital water resources.

2 Related Works

Rainfall prediction using machine learning (ML) and deep learning (DL) techniques has advanced beyond traditional methods by effectively capturing complex temporal and spatial patterns. Salehin et al. applied Long Short-Term Memory (LSTM) networks to predict rainfall in regions with abundant precipitation, demonstrating the model's ability to capture temporal dependencies. However, they noted LSTM's tendency to overfit in areas with sparse rainfall, where class imbalances dominated by non-rainy days hindered generalization to rare precipitation events.

Wani et al. incorporated both ML and DL techniques, such as Artificial Neural Networks (ANNs) and Random Forests (RFs), alongside traditional time series models to predict rainfall in the North-Western Himalayas. Their study showed that DL approaches outperformed others in modeling nonlinear relationships but was limited to regions with consistent rainfall. The effectiveness of these models in low-rainfall areas, characterized by data sparsity and irregular patterns, remains largely untested.

This study builds on the methodologies of Salehin et al. and Wani et al., focusing on the dry Midwest region, characterized by sparse precipitation. Starting with LSTM as a baseline, it explores the model's performance in the presence of severe class imbalance. To overcome LSTM's limitations, Gated Recurrent Units (GRU) and Transformer models are employed. GRUs offer a more efficient architecture for limited data, while Transformers leverage self-attention to capture long-range dependencies and rare events. This research aims to improve predictive accuracy for regions where minimal rainfall is critical for agriculture and the economy.

3 Data

The data for this study was sourced from weather stations across seven Midwestern U.S. states (Minnesota, Iowa, Missouri, Wisconsin, Illinois, Indiana, and Michigan) from 1970 to 2020. The dataset includes daily records with features such as temperature (min, max, average), humidity, wind speed, wind direction, dew point, and precipitation (target variable), along with other atmospheric measurements. To handle missing values, temperature-related columns were forward-filled, and precipitation values were filled with zeros to reflect the dominance of dry days. Min-max scaling was applied to normalize features, ensuring consistency across varying scales like temperature (in Fahrenheit) and precipitation (in millimeters), improving model performance. The dataset exhibited significant class imbalance, with most days showing no precipitation. Attempts to address this through oversampling were ineffective, as it failed to add meaningful variability. To preserve temporal dependencies, the data was split chronologically into training (1970-2004), validation (2005-2010), and testing (2011-2020) sets, ensuring realistic forecasting conditions without shuffling.

4 Methods

4.1 LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to capture long-range temporal dependencies in sequential data. Their unique architecture, which includes memory cells and gating mechanisms, makes them particularly well-suited for time series problems such as rainfall prediction, where past weather patterns influence future outcomes. In this study, the LSTM model serves as a baseline to evaluate its performance in predicting both daily and weekly precipitation in the Midwest, a region characterized by sparse and irregular rainfall events.

4.1.1 Model Architecture and Design

The Long Short-Term Memory (LSTM) model employed in this study is designed to predict precipitation using historical weather data. The architecture consists of a multi-layer LSTM module followed by a fully connected (FC) layer. The LSTM module comprises two layers, each with 128 hidden units. These layers capture temporal dependencies in the input sequences, which consists of 26 features, including temperature, humidity, wind speed, and other relevant weather indicators.

The final time step's hidden state is passed through to a fully connected layer, which maps the 128-dimensional hidden representation to a single continuous output representing the predicted precipitation value. The choice of two LSTM layers balances model complexity and computational efficiency, while the hidden size of 128 provides sufficient capacity to capture intricate temporal patterns without overfitting. The `batch_first = True` configuration ensures the input tensors follow the structure (batch_size, seq_len, input_size), aligning with the data preprocessing pipeline.

Hyperparameters were turned iteratively, with the hidden size and number of layers selected based on validation performance. The architecture allows the model to generalize effectively across diverse temporal patterns, particularly in regions with imbalance precipitation data.

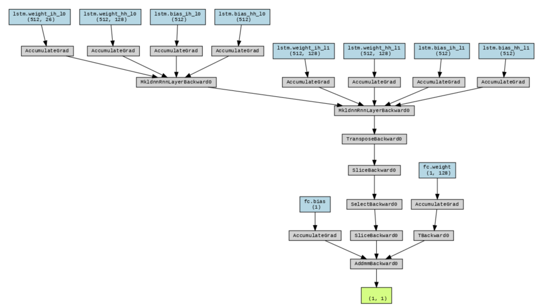

Figure 1: LSTM Model Architecture.

4.1.2 Training Process

The LSTM model was trained using the Mean Squared Error (MSE) loss function, which is appropriate for regression tasks where the objective is to minimize the squared difference between predicted and true continuous values. The Adam optimizer, with a learning rate of 0.001, was employed for parameter updates during backpropagation. Training was conducted over 50 epochs, utilizing mini-batches of the data fed through a DataLoader to efficiently handle the large-scale dataset. For each batch, gradients were reset, a forward pass through the LSTM was performed, and the loss was computed by comparing the model's outputs to the ground truth labels. This loss was then backpropagated through the network to update the model's parameters. The training loss consistently decreased throughout the epochs, from 0.00082170 at epoch 1 to 0.00077521 at epoch 50, indicating that the model effectively minimized the error during training. The steady reduction in training loss suggests that the LSTM model converged toward an optimal solution, with diminishing error after each epoch.

To assess the generalization capability of the trained model, the validation loss was computed after each epoch by evaluating the model on a held-out validation set. The model was switched to evaluation mode to disable gradient updates and prevent overfitting. The validation loss averaged 0.0007, indicating the model's strong performance on unseen data and its ability to generalize beyond the training set.

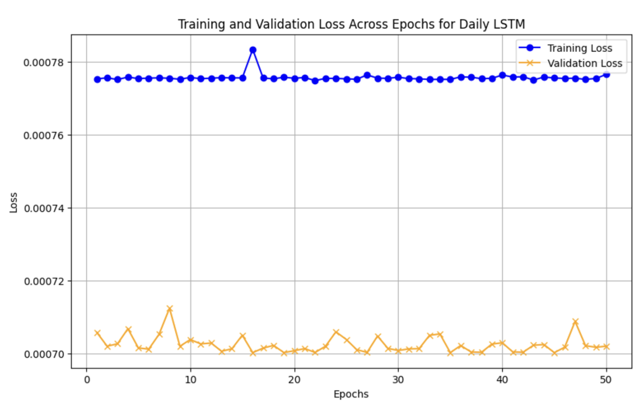

Figure 2: LSTM Training vs. Validation Loss.

4.1.3 Handling Imbalances and Temporal Data

The LSTM model effectively addressed temporal dependencies in the dataset, as the data was not shuffled during training to preserve sequential order. This allowed the LSTM to capture long-term dependencies, which helped it predict zero precipitation days accurately, given their dominance in the dataset. However, the severe imbalance—due to the overwhelming presence of zero precipitation days—led to a model that easily predicted these days, resulting in a low training loss. While this was favorable for predicting zero precipitation, the LSTM struggled to generalize to non-zero precipitation days, especially those with low or high rainfall. Attempts to mitigate the imbalance through oversampling by duplicating sequences with rare target labels (e.g., the ones) did not improve generalization. Oversampling inflated the importance of rare examples but did not introduce new contextual variability, causing the model to overfit to these sequences and memorize them instead of learning generalizable patterns. To better address both the imbalance and the generalization issue, we explored alternative architectures, such as the GRU and Transformer models, which were more effective at capturing the complex relationships between precipitation levels and temporal dependencies.

4.1.4 Evaluation on Daily vs. Weekly Metrics

The LSTM model's performance was evaluated on daily and weekly precipitation predictions. On daily predictions, the model achieved 91.9% accuracy for binary precipitation classification but performed poorly on large precipitation events, with a large value accuracy of 0%. This suggests the model excelled in predicting zero precipitation days but struggled to generalize to days with higher rainfall due to the dataset's imbalance.

For weekly predictions, the binary accuracy dropped to 63.5%, indicating difficulty in generalizing over longer time scales. The large value accuracy remained 0%, reflecting the model's challenge in predicting weeks with significant rainfall. This suggests that the LSTM struggled with long-range dependencies, especially for rare, high-precipitation events.

Overall, while the LSTM performed well for binary precipitation classification on daily data, it struggled with larger rainfall events and weekly predictions, highlighting the need for exploring alternative architectures like GRUs and Transformers to capture both short-term and long-range dependencies.

4.2 GRU

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) designed for capturing long-range temporal dependencies in sequential data. It's streamlined architecture, which merges the forget and input gates into a single update gate, enhances computational efficiency while effectively modeling sequential patterns. This simplified design enables GRUs to balance memorization and generalization, making them well-suited for handling imbalanced datasets like ours. Given these strengths, we hypothesized that GRUs could outperform LSTMs in predicting precipitation patterns. To test this, we implemented both daily and weekly GRU models, focusing on their ability to address data imbalances and capture short and long term dependencies.

4.2.1 Model Architecture and Design

The GRU model was initially implemented with a single layer of 64 hidden states and later expanded to four layers to improve performance. A Dense output layer with a linear activation function predicted continuous precipitation values. Dropout (rate=0.2) was applied to mitigate overfitting, showing improvements in the weekly model but impairing daily model's ability to predict larger precipitation values. Consequently, dropout was retained only for the weekly GRU. To address class imbalance, we experimented with upsampling non-zero precipitation values and introducing noise to create diverse samples. While these methods improved predictions for large precipitation values, they increased loss (see Figure 3) and were excluded from the final daily GRU model. Weighted loss functions were also tested but not adopted due to similar concerns. The weekly GRU, trained on a smaller dataset, avoided resampling to reduce overfitting risk. The final architectures for both daily and weekly GRU models are depicted in Figure 3.

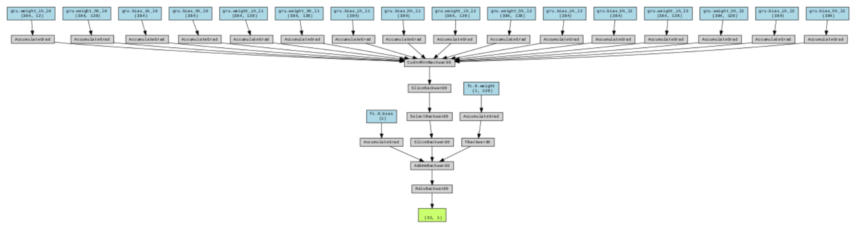

Figure 3: GRU Architecture.

4.2.2 Training Process

The GRU models were trained using the Adam optimizer, starting with an initial learning rate of 0.001, which was later increased to 0.002 to accelerate convergence. Training was performed over 60 epochs, chosen based on the convergence of training loss and computational efficiency (see Figure 4). A batch size of 32 was used to balance gradient updates. As with the LSTM models, the primary loss function was MSE, with MAE monitored as an auxiliary metric. When comparing the daily GRU and the weekly GRU, it is important to consider the aggregated loss for the weekly model. The mean loss for the daily model, using individual days, was 0.077, while for the weekly model, with 7-day periods aggregated the mean loss increased to 0.131. Thus, the loss of the two models should be evaluated relative to their respective data scales.

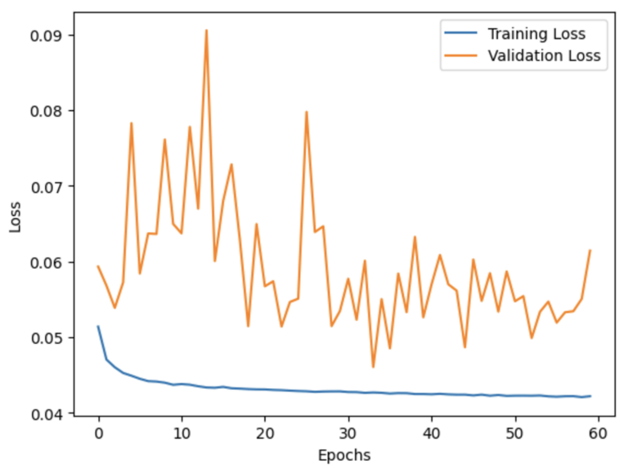

Figure 4: Daily GRU Training Loss.

4.2.3 Handling Imbalances and Temporal Data

The GRU model's simpler architecture made it better equipped to handle the data imbalances in our dataset compared to the LSTM. By combining the forget and input gates into a single update gate, the GRU reduced model complexity, enabling it to focus on relevant patterns without overfitting to the abundant zero-precipitation days. This efficiency allowed the GRU to generalize for large precipitation values due to the dominance of zeros in the daily data. However, the weekly GRU excelled in predicting large precipitation totals, achieving 76.5% large-value accuracy - more the double the 37.7% large-value accuracy of the final daily GRU. Aggregating data into weekly windows allowed the model to capture long-term trends and better handle significant precipitation events.

4.2.4 Evaluation on Daily vs. Weekly Metrics

The evaluation of the GRU models revealed distinct strengths and limitations when addressing binary classification and large precipitation values. The original daily GRU performed poorly in binary classification, achieving only 39.1% accuracy. However, resampling and fine-tuning significantly improved its ability to distinguish between near-zero and non-zero precipitation events, boosting binary accuracy to 78.2% in the resampled GRU and 90.8% in the final model. These results underscore the effectiveness of targeted adjustments in mitigating class imbalances.

In contrast, the weekly GRU achieved a lower binary accuracy of 56.4%, reflecting the challenges of classifying near-zero values in aggregated data, where temporal patterns are less pronounced, and zeros less frequent. Despite these limitations, the weekly GRU demonstrated a notable advantage in predicting large precipitation totals, achieving 76.5% large-value accuracy – more than double the 37.7% of the final daily GRU. This performance gap highlights the weekly model's ability to capture long-term trends in precipitation, making it more effective for forecasting significant rainfall events. Aggregating data into weekly windows allowed the weekly GRU to overcome the dominance of zeros and focus on meaningful temporal patterns associated with larger precipitation values.

4.3 Transformer

The Transformer model has transformed sequence modeling through its self-attention mechanism, enabling it to capture long-range dependencies without relying on recurrent structures. By processing entire sequences in parallel, Transformers achieve greater computational efficiency and excel at modeling complex temporal patterns. In this study, the Transformer is utilized to predict daily and weekly precipitation in the Midwest, offering a robust solution to the challenges of sparse and irregular rainfall in the region.

4.3.1 Model Architecture and Design

The Transformer architecture begins with an embedding layer that transformers 13 weather-related features, such as temperature and humidity, into a 64-dimensional representation suitable for processing within the Transformer framework. To retain the sequential structure of the data, positional encodings are added to the embeddings, allowing the model to distinguish temporal order and capture essential time-series information.

The core of the model comprises two Transformer encoder layers, each with four attention heads. The multi-head attention mechanism enables the model to simultaneously focus on different parts of the input sequence, effectively capturing complex relationships among weather variables. Each encoder layer includes a feed-forward network with a hidden size of 256, providing sufficient capacity to model intricate temporal patterns.

The encoded outputs are passed through a fully connected layer, which maps the 64-dimensional representation to a single continuous value representing predicted precipitation. The choice of two encoder layers and four attention heads, based on empirical tuning, balances model complexity with computational efficiency. This architecture effectively models both short-term and long-term dependencies, making it highly suited to the challenges of precipitation forecasting in arid regions.

4.3.2 Training Process

The Transformer model was trained to minimize the MSE loss, using the Adam optimizer with a learning rate of 0.00001. Initial experiments with higher learning rates, such as 0.01, resulted in poor performance, particularly for large precipitation values, highlighting the importance of fine-tuning this hyper parameter. Training spanned 100 epochs with a batch size of 32. Loss curves showed consistent decreases before plateauing, indicating effective learning without overfitting (see Figure 4.) The model's computational efficiency allowed rapid experimentation with hyper parameters, leading to optimized performance. Validation loss trends confirmed minimal overfitting, eliminating the need for early stopping. The final model excelled in capturing complex relationships in the data, especially for forecasting large precipitation events.

4.3.3 Handling Imbalances and Temporal Data

The dataset exhibited a significant class imbalance, dominated by zero-precipitation events, posing a challenge for effective prediction. The Transformer model addressed this issue by leveraging its self-attention mechanism and positional encoding to capture complex temporal dependencies while mitigating the impact of imbalance. Positional encoding integrated the temporal order of weather events into the model, enabling it to understand sequential patterns essential for precipitation prediction. To provide historical context, the input data was preprocessed into overlapping five-week

Figure 5: Transformer Model Architecture.

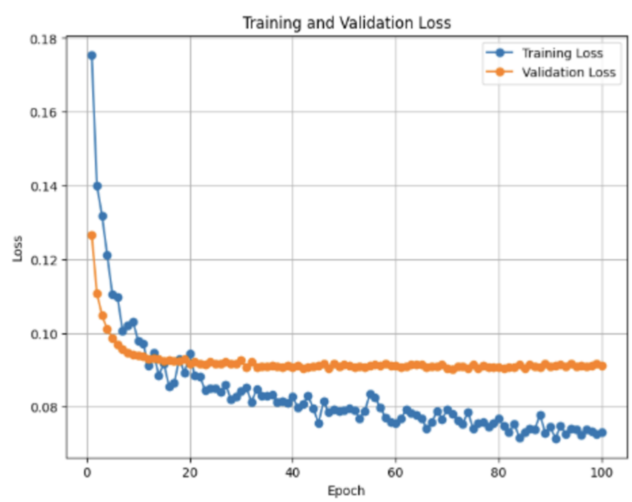

Figure 6: Transformer Training vs. Validation Loss.

sequences, each paired with the target precipitation value for the subsequent week. This approach ensured that the model could learn from meaningful patterns in the data.

Unlike oversampling techniques, which can lead to overfitting by duplicating rare events, the Transformer inherently focused on relevant features through its multi-head attention mechanism. This allowed the model to weigh different parts of the input sequence according to their importance, emphasizing critical features and reducing the adverse effects of the class imbalance. By capturing complex temporal relationships and concentrating on key patterns, the Transformer demonstrated improved robustness and accuracy in predicting both common and rare precipitation events.

4.3.4 Evaluation on Daily vs. Weekly Metrics

The Transformer model's performance in predicting daily and weekly precipitation showed marked improvement over traditional RNN-based approaches. For weekly precipitation, the Transformer achieved a test loss of 0.086, significantly outperforming the LSTM and GRU models, which had test losses of approximately 0.130. The model also demonstrated a binary accuracy of 74.4% for precipitation occurrence, surpassing the weekly performance of other models by ten percentage points.

A key strength of the Transformer was its ability to predict large precipitation values, achieving a large-value accuracy of 90.8%. This highlights the model's effectiveness in capturing the nuances of significant rainfall events, which are crucial for weather forecasting and risk management in regions where precise predictions are essential for agricultural planning and disaster mitigation.

5 Results

The LSTM model achieved the lowest test loss and the highest binary accuracy of all models, excelling at predicting the presence or absence of precipitation on a daily basis. However, its large-value accuracy was 0% (see Table 1), highlighting a significant limitation in forecasting significant rainfall events. This suggests that the model overfittted to the majority of zero-precipitation days, effectively learning to predict "no rain" but failing to capture larger rainfall amounts.

Both the daily and weekly GRU models showed moderate improvements over the LSTM in predicting larger precipitation values, with large-value accuracies of 37.7% and 76.5%, respectively. However, their overall performance remained suboptimal, with binary accuracies lower than the LSTM, indicating less effectiveness in classifying precipitation events. While the GRU models reduced overfitting compared to the LSTM, they still struggled to generalize to more significant rainfall events.

The Transformer model outperformed the RNN-based weekly models across all evaluation metrics. It achieved the lowest test loss (0.086), indicating superior overall accuracy. The binary accuracy of 74.4% was significantly higher than that of the weekly RNN models, demonstrating better performance in identifying precipitation events. Most notably, the Transformer model excelled in large-value accuracy, achieving 90.8%, reflecting its capacity to predict significant rainfall events accurately. This performance is attributed to the Transformer's ability to capture long-term dependencies and critical time steps through attention mechanisms and positional encoding, overcoming the overfitting challenges seen in the RNN models.

Table 1: Model Evaluation Metrics

Metric	LSTM Daily	GRU Daily	LSTM Weekly	GRU Weekly	Transformer Weekly
Test Loss	0.0007	0.051	0.130	0.131	0.086
Binary Above/Below	91.9%	90.8%	63.5%	56.4%	74.4%
Large Value Accuracy	0%	37.7%	77.7%	76.5%	90.8%

6 Discussion and Future Directions

The evaluation results indicate that while traditional RNN models (LSTM and GRU) struggled with overfitting and had limited success in predicting large precipitation values, the Transformer model addressed these issues effectively. By leveraging self-attention mechanisms and capturing complex temporal dependencies, the Transformer delivered superior predictions, particularly for significant precipitation events. These improvements are crucial for practical applications such as agriculture, where accurate weather forecasts play a vital role in optimizing farming practices. For instance, reliable predictions of rainfall patterns can directly influence irrigation strategies, crop management, and yield forecasts. In particular, forecasting extreme precipitation events is essential for managing water resources and preventing crop damage due to flooding or drought. Accurate predictions of rainfall distribution and timing can help farmers adjust planting schedules, optimize resource use, and mitigate risks associated with unpredictable weather patterns. Future work could focus on further optimizing Transformer models for real-time forecasting and possibly creating hybrid architectures such as a GRU-Transformer in order to better capture temporal complexities in arid climates.

References

[1] Farooq, R., Imteaz, M. A., Shangguan, D., & Hlavčová, K. (2024). Machine learning algorithms to forecast wet-period rainfall using climate indices in the Northern Territory of Australia. Scientific Reports, 14, 100397. https://doi.org/10.1016/j.sctalk.2024.100397

[2] Hassan, M. M., et al. (2023). Machine learning-based rainfall prediction: Unveiling insights and forecasting for improved preparedness. IEEE Access, 11, 132196-132222. https://doi.org/10.1109/ACCESS.2023.3333876

[3] Hernandez, E. J., Sanchez-Anguix, V., Julián, V., & Palanca, J. (2016). Rainfall prediction: A deep learning approach. In Lecture Notes in Computer Science (Vol. 9746, pp. 137-144). Springer. https://doi.org/10.1007/978-3-319-32034-2-13

[4] Salehin, I., Talha, I., Hasan, M., Dip, S., Saifuzzaman, M., & Nessa, N. (2021). An artificial intelligence-based rainfall prediction using LSTM and neural network. Proceedings of the IEEE Women in Engineering Conference (WIECON-ECE), 1-5. https://doi.org/10.1109/WIECON-ECE52138.2020.9398022.

[5] Wani, O. A., Mahdi, S. S., Yeasin, M., et al. (2024). Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas. Scientific Reports, 14, 27876. https://doi.org/10.1038/s41598-024-77687-x