# *Rainfall Amount Prediction with Climate Data and Geospatial Data*

***Jonathan Lee, Ayumu Yamagishi***

***Github: https://github.com/000Aym612/RainfallPredictionWithClimateData/tree/main***

## 1. Introduction

This report presents a deep learning approach to rainfall prediction using the **LAND (Location-Agnostic Neural Downscaling)** methodology (Hatanaka et al., 2024). The goal of the project is to predict rainfall amounts across a geographical grid of American Samoa using a combination of climate variables and topographical information. The LAND methodology addresses a fundamental challenge in environmental modeling: how to effectively combine data from different spatial scales and domains to make accurate predictions.

### 1.1. Essential Components for Prediction

1. **Climate reanalysis data:** 16 atmospheric variables from climate reanalysis datasets (Sanfilippo et al., 2023), spanning multiple pressure levels (1000, 850, 700, 500 mb) that capture large-scale weather patterns.
2. **Digital Elevation Model (DEM):** Topographical information on a region. We use two scaled DEM patches (3x3 grids): a local patch (4km/cell) and regional patch (20km/cell).
3. **Temporal Information:** Month one-hot encoding to capture seasonal patterns.

### 1.2. Rainfall Amount Prediction Models

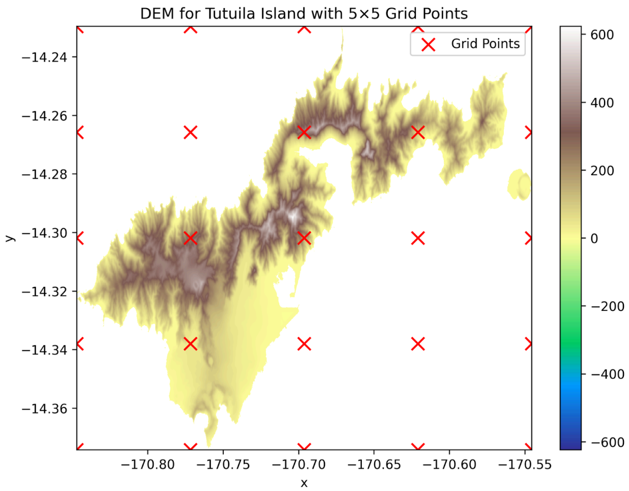
1. **Single Best Model:** A single hyperparameter-optimized LAND model with the best hyperparameters.
2. **Ensemble Cross-Validation Model:** A robust ensemble of 5 of the single best model trained at different random seeds with 5-fold cross-validation.

The ensemble approach achieved a test **$R^2$ of 0.7955**, explaining nearly 80% of the variance in rainfall patterns. This represents a significant advancement in rainfall prediction capability using the LAND methodology.

## 2. Approaches & Methodology

### 2.1 Data Sources and Preprocessing

As described above, our rainfall prediction models integrate three key components. In this section we will explore the variables, information and detailed explanation for each of the components. Before delving into the details, we shall explain the format of a single row of data. Each row of data contains: 16 climate reanalysis features, a 3x3 local DEM patch, a 3x3 regional DEM patch, a one-hot encoding of the month, and the rainfall amount (as the label). For every month from July 1958 to December, 2024, these variables were determined for 25 points evenly distributed throughout our area of interest: Tutuila island. A high-resolution DEM of Tutuila island, American Samoa acted as the boundaries for our grid (see below).



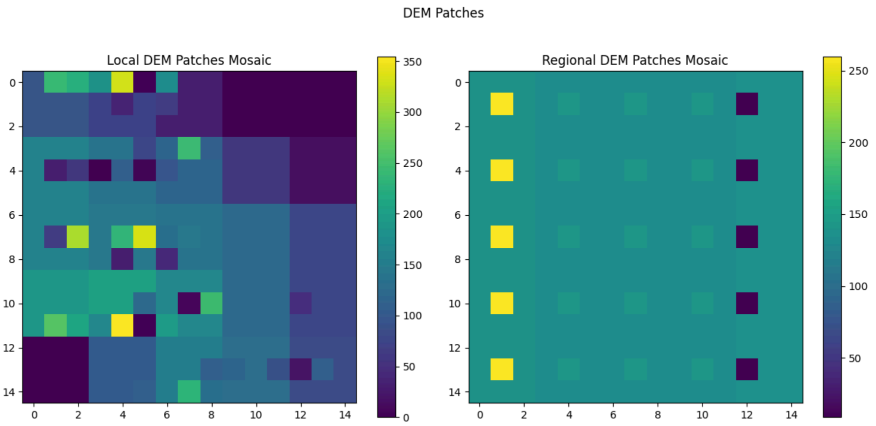
**Climate Reanalysis Data:** The 16 climate reanalysis features (Sanfilippo et al., 2023) were derived from the NCEP/NCAR Reanalysis dataset at a resolution of 2.5° × 2.5°. These features capture large-scale atmospheric conditions and include:

| | |
|---|---|
| Air temperature difference (1000~500 mb) | Specific humidity at 700 mb |
| Air temperature at 2 m | Specific humidity at 925 mb |
| Geopotential height at 1000 mb | Zonal moisture transport at 700 mb (uwnd700×shum700) |
| Geopotential height at 500 mb | Zonal moisture transport at 925 mb (uwnd925×shum925) |
| Omega (vertical velocity) at 500 mb | Meridional moisture transport at 700 mb (vwnd700×shum700) |
| Potential temperature difference (1000~500 mb) | Meridional moisture transport at 925 mb (vwnd925×shum925) |
| Potential temperature difference (1000~850 mb) | Skin temperature |
| Precipitable water | Sea level pressure |

Each variable was normalized by dividing by the maximum absolute value for that variable.

**Digital Elevation Model (DEM) Data**: Topographical information was processed at two scales, and divided by 4000.0 to normalize the elevation values:

1. **Local scale:** 3×3 patches with 4 km per cell (12km total coverage)
2. **Regional scale:** 3×3 patches with 20 km per cell (60km total coverage)
3. This dual-scale approach allows the model to capture both fine-grained local terrain features and broader regional topographical patterns:



The figures show the difference between the local and regional DEM. A block of 3x3 squares in the same location on the local and regional DEM represents the input data in a single row of data. The entire "map" represents the 5x5 grid from the red marks on the DEM of Tutuila above. Thus we have  25 squares of 3x3 pixels.

All input features were standardized by dividing by 4000.0 . This preprocessing step is crucial for neural networks to effectively learn from diverse input features.

**Month Encoding:**

To capture seasonal patterns in rainfall, we included a one-hot encoding (12 separate features) representing the month of the year. This allows the model to learn seasonal patterns in rainfall distribution.

**Target Variable (Rainfall):**

We determined a monthly rainfall amount, which was interpolated to match our 5x5 grid using Gaussian processes from monthly-aggregated daily rain gauge data across Tutuila island. Rainfall data was divided by 100 for scaling to match the LAND process.

The data was then split into 90% for training (with 90% of that for the training set and 10% for the validation). The test set contained the remaining 10% of the data.

### 2.2 Neural Network Architecture

Our neural network architecture follows the LAND methodology, with separate processing branches for different data types that are later combined:

1. Climate Branch:
   - Input: 16 climate variables
   - Processing: Dense layers with batch normalization
   - Output: Feature vector of size climate_units (384 in optimal configuration)
2. Local DEM Branch:

  - Input: Flattened 3×3 local DEM patch
  - Processing: Dense layers with batch normalization
  - Output: Feature vector of size local_dem_units (64 in optimal configuration)
3. Regional DEM Branch:
   - Input: Flattened 3×3 regional DEM patch
   - Processing: Dense layers with batch normalization
   - Output: Feature vector of size regional_dem_units (32 in optimal configuration)
4. Month Branch:
   - Input: One-hot encoded month vector (12 dimensions)
   - Processing: Dense layers with batch normalization
   - Output: Feature vector of size month_units (16 in optimal configuration)

These branch outputs are concatenated and fed through two dense layers with dropout regularization:

- First dense layer: “na” units (512 in optimal configuration)
- Second dense layer: “nb” units (384 in optimal configuration)
- Output layer: Single neuron for rainfall prediction

### 2.3 Model Training Approaches

We implemented two primary training approaches:

#### 2.3.1 Single Best Model

This approach focuses on finding and training a single optimal model:

- Hyperparameter Optimization: Extensive tuning to find the best architecture and training parameters
- Full Dataset Training: Training on the entire training set with early stopping based on validation performance
- Regularization Techniques: L2 regularization, dropout, and batch normalization to prevent overfitting

#### 2.3.2 Ensemble Cross-Validation Model

This approach combines the power of ensemble learning with k-fold cross-validation:

- K-fold Cross-Validation: The dataset is divided into 5 folds, with each fold serving as a validation set once while the remaining folds form the training set.
- Ensemble per Fold: For each fold, we train 5 models with different random initializations.
- Combined Prediction: The final prediction is an average of all 25 models (5 models × 5 folds).

This approach provides several advantages:

- Robust performance across different data subsets
- Reduced variance in predictions
- Better generalization to unseen data
- Uncertainty quantification through model variance

## 3. Hyperparameter Tuning

### 3.1 Extended Hyperparameter Optimization

We implemented a comprehensive hyperparameter tuning system using Keras Tuner with Bayesian optimization. This approach efficiently explores the parameter space by using the results of previous trials to inform the selection of hyperparameters for subsequent trials.

#### 3.1.1 Tunable Parameters

Our tuning process explored the following hyperparameters:

1. Network Architecture:
   - Number of neurons in hidden layers (na, nb)
   - Units for each branch (local_dem_units, regional_dem_units, month_units, climate_units)
   - Activation functions (ReLU, ELU, SELU)
   - Optional residual connections
2. Regularization:
   - Dropout rate
   - L2 regularization strength
   - Batch normalization configuration
3. Optimization Parameters:
   - Learning rate
   - Weight decay for AdamW optimizer
   - Learning rate schedule parameters (warmup proportion, decay steps)

#### 3.1.2 Tuning Process

The hyperparameter tuning process was configured with the following settings:

- Algorithm: Bayesian Optimization with Hyperband early stopping
- Number of Trials: 100
- Epochs per Trial: Up to 50 with early stopping
- Objective: Minimize validation loss (MSE)
- Early Stopping: Patience of 15 epochs

This extended tuning process allowed us to explore a much larger hyperparameter space than traditional grid or random search methods, resulting in significantly improved model performance.

#### 3.1.3 Optimal Hyperparameters

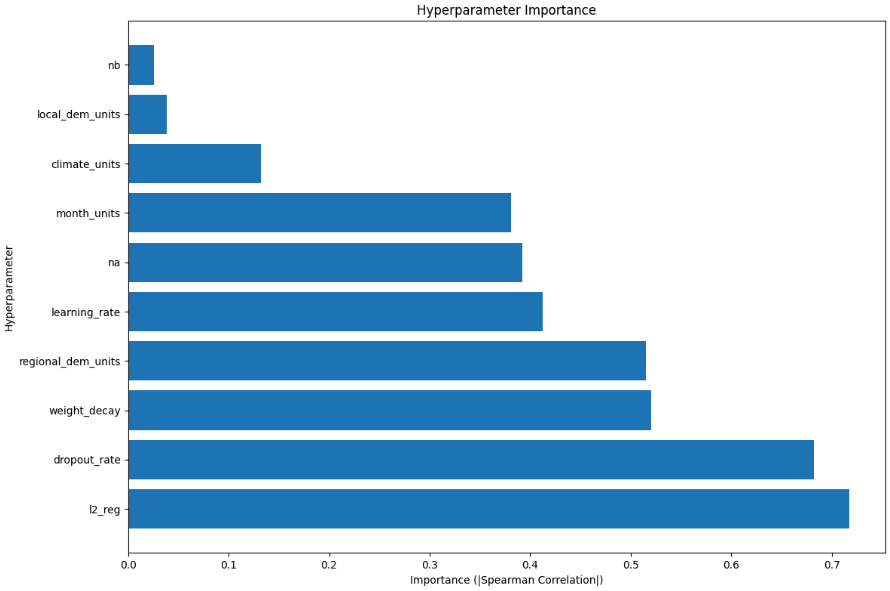
The best hyperparameters found through our extended tuning process were:

| | |
|---|---|
| **na**: 512 | **nb**: 384 |
| **dropout_rate**: 0.1 | **l2_reg**: 1e-06 |
| **learning_rate**: 0.01 | **weight_decay**: 1e-07 |
| **local_dem_units**: 64 | **regional_dem_units**: 32 |
| **month_units**: 16 | **climate_units**: 384 |
| **use_residual**: False | **activation**: relu |

These optimal hyperparameters were used in both our single best model and as the base configuration for our ensemble models.

### 3.2 Hyperparameter Importance Analysis

The hyperparameter importance analysis (see figure below) revealed the following ranking, based on the absolute Spearman correlation between each hyperparameter and validation performance:

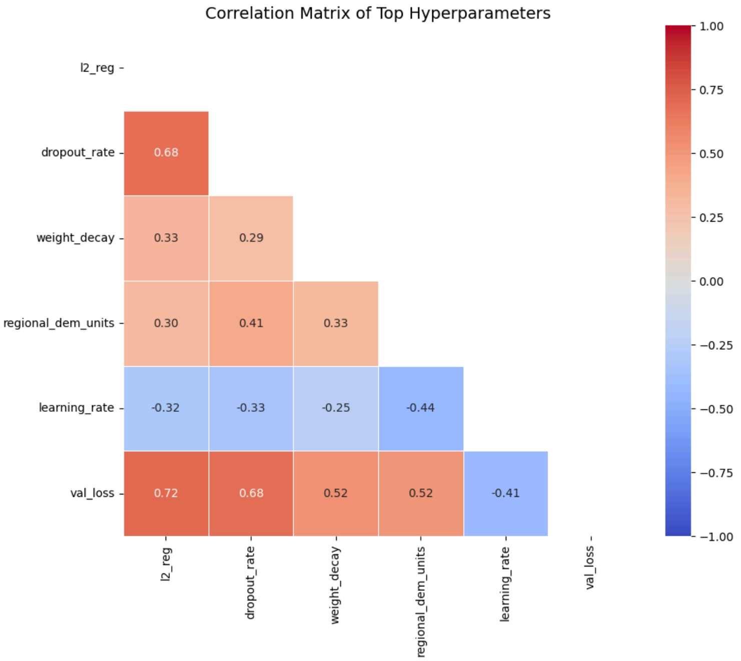


1. **L2 Regularization (l2_reg):** Most important, with the highest correlation to model performance. Careful tuning of L2 regularization was critical to prevent overfitting.
2. **Dropout Rate (dropout_rate):** Also highly influential, confirming the importance of regularization in this task.

3. **Weight Decay (weight_decay):** Played a significant role in optimizing the AdamW optimizer's effectiveness.
4. **Regional DEM Units (regional_dem_units):** The number of units for regional DEM processing had a strong impact, highlighting the value of regional topographical context.
5. **Learning Rate (learning_rate):** As expected, learning rate was a sensitive parameter affecting convergence and final accuracy.
6. **na (na):** The size of the first dense layer after concatenation contributed noticeably to performance.
7. **Month Units (month_units)**: The capacity of the month encoding branch was moderately important.
8. **Climate Units (climate_units):** The number of units for climate variable processing had a smaller, but still positive, effect.
9. **Local DEM Units (local_dem_units):** The local DEM branch size was of minor importance.
10. **nb (nb):** The second dense layer size had the least impact among all tested hyperparameters.

The importance of each hyperparameter is further illustrated in the figure below, which shows the correlation matrix between the top 5 hyperparameters and validation loss. In this matrix, a higher absolute value indicates a stronger relationship with model performance (validation loss):

- L2 regularization and dropout rate show the strongest positive correlations with validation loss, confirming their critical role in preventing overfitting and improving generalization.
- Weight decay and regional_dem_units also display moderate positive correlations, indicating their importance for regularization and spatial feature extraction, respectively.
- Learning rate is negatively correlated with validation loss, highlighting that careful tuning of this parameter is essential for effective model convergence and performance.



## 4. Single Best Model

### 4.1 Architecture and Implementation

The single best model approach focuses on training a single optimal model with the best hyperparameters identified through our extensive tuning process. While simpler than the ensemble approach, this model still incorporates the core LAND methodology and advanced architectural features.

#### 4.1.1 Model Structure

The single best model follows the same LAND-inspired architecture described in the methodology section:

- Separate processing branches for climate variables, local DEM, regional DEM, and month encoding
- Optimal branch sizes determined through hyperparameter tuning:
    - Climate branch: 384 units
    - Local DEM branch: 64 units
    - Regional DEM branch: 32 units
    - Month encoding branch: 16 units
- Two dense layers after concatenation:
    - First layer: 512 units with ReLU activation
    - Second layer: 384 units with ReLU activation
- Regularization techniques:
    - Dropout (rate = 0.1)
    - L2 regularization (1e-06)
    - Batch normalization after each dense layer

#### 4.1.2 Training Process

The single best model was trained using:

- AdamW optimizer with:
    - Learning rate: 0.01
    - Weight decay: 1e-07
- Cosine decay learning rate schedule with warm up
- Early stopping with patience of 15 epochs
- Batch size of 314 (derived from the LAND paper)
- Training/validation split of 80%/20%

### 4.2 Performance Results

The single best model showed strong performance on the validation set but exhibited signs of overfitting when evaluated on the test set:

#### 4.2.1 Validation Performance

- $R^2$: 0.7212
- RMSE: .4913 in
- MAE: .1324 in

#### 4.2.2 Test Performance

- $R^2$: 0.2472
- RMSE: .8715 in
- MAE: .1669 in

The significant gap between validation and test performance ($R^2$ drop from 0.7212 to 0.2472) indicates that the single model approach, despite using optimal hyperparameters and regularization techniques, still suffers from overfitting to the training/validation data.

### 4.3 Limitations of the Single Model Approach

The single best model approach has several limitations compared to an ensemble approach:

1. Higher Variance: A single model is more susceptible to variance in prediction, leading to less stable performance across different datasets.
2. Overfitting Risk: Despite regularization techniques, a single model is more prone to overfitting to the training data.
3. Initialization Sensitivity: Performance can vary significantly based on random weight initialization; this is especially evident given the uneven rainfall data (more data available for the later dates).
4. Limited Generalization: A single model may struggle to generalize across different patterns in the data.

These limitations are evident in the performance gap between validation and test sets. While the single best model achieved good validation performance ($R^2 = 0.7212$), its test performance ($R^2 = 0.2472$) was significantly lower, highlighting the challenges of generalization with a single model approach.

## 5. Ensemble Cross-Validation Model

### 5.1 Architecture and Implementation

Our ensemble cross-validation model represents the most advanced implementation of the LAND-inspired rainfall prediction system. This approach combines the strengths of ensemble learning with k-fold cross-validation to achieve robust and accurate predictions.

#### 5.1.1 Ensemble Structure

The ensemble model consists of:

- 5 cross-validation folds: The dataset is divided into 5 equal parts, with each part serving as a validation set once while the remaining folds form the training set.

- 5 models per fold: For each fold, we train 5 identical models with different random initializations.
- 25 total models: The full ensemble combines predictions from all 25 models.

Each individual model follows the LAND architecture described in the methodology section, with the optimal hyperparameters identified through our extended tuning process.

### 5.1.2 Training Process

The training process for the ensemble model follows these steps:

1. Data Splitting: The dataset is first split into training (80%) and test (20%) sets. The training set is then further divided into 5 folds for cross-validation.
2. Per-Fold Training: For each fold:
   - Train 5 models using the training data (4/5 of the training set)
   - Validate each model on the validation data (1/5 of the training set)
   - Save model weights and performance metrics
3. Ensemble Aggregation: Combine all 25 models by averaging their predictions.
4. Test Evaluation: Evaluate the full ensemble on the held-out test set.

The training used early stopping with a patience of 15 epochs to prevent overfitting, and employed the AdamW optimizer with a cosine decay learning rate schedule.

## 5.2 Performance Results

The ensemble cross-validation model achieved exceptional performance:

### 5.2.1 Cross-Validation Results

| **Measure** | **Average across 5 folds** | **Best fold (Fold 5)** |
|---|---|---|
| $R^2$ | 0.6253 | .7012 |
| RMSE | .6334 in | .5895 in |
| MAE | .1443 in | .1382 in |

### 5.2.2 Test Set Results

Performance on the held-out test set:

- $R^2$: 0.7955
- RMSE: .4542 in
- MAE: .1325 in

The ensemble model's test $R^2$ of 0.7955 represents a vast improvement on a single model and demonstrates the effectiveness of the ensemble cross-validation strategy. Below is the figure showing the actual vs predicted rainfall values for the ensemble model:

### 5.3 Advantages of the Ensemble Approach

The ensemble cross-validation approach offers several key advantages:

1. Reduced Variance: By averaging predictions from multiple models, the ensemble reduces the variance component of prediction error.
2. Improved Generalization and Reduced Overfitting: The cross-validation structure ensures the model generalizes well across different subsets of the data. The ensemble structure inherently reduces overfitting to any particular subset of the training data.
3. Robustness to Initialization: Using multiple random initializations per fold mitigates the impact of poor initialization on model performance.

These advantages are clearly reflected in the exceptional test set performance, where the ensemble model achieved an $R^2$ of 0.7955, significantly outperforming the single best model approach.

## 6. Comparison and Analysis

### 6.1 Performance Comparison

The table below summarizes the performance metrics for both the ensemble cross-validation model and the single best model:

| Model | Validation $R^2$ | Test $R^2$ | Test RMSE (in) | Test MAE (in) |
|---|---|---|---|---|
| Ensemble CV Model | 0.6253 (avg) | 0.7955 | .4542 | .1325 |
| Single Best Model | 0.7212 | 0.2472 | .8715 | .1669 |

This comparison reveals several important insights:

1. Generalization Gap: The single model shows a significant drop in performance from validation to test set ($R^2$ drop of 0.474), while the ensemble model actually performs better on the test set than on validation ($R^2$ improvement of 0.1702).
2. Superior Test Performance: The ensemble model's test $R^2$ (0.7955) is substantially higher than the single model's (0.2472), representing a 221% improvement in explained variance.
3. Error Reduction: The ensemble approach reduces RMSE by 48% (from .8715 in to .4542 in) and MAE by 20% (from .1669 in to .1325 in) compared to the single model.

### 6.2 Analysis of Model Behavior

#### 6.2.1 Ensemble Model Advantages

The ensemble cross-validation model's superior performance can be attributed to several factors:

1. Variance Reduction: By averaging predictions from 25 different models, the ensemble significantly reduces the variance component of prediction error.
2. Robust Generalization: The cross-validation structure ensures that each data point is predicted by models that never saw it during training, leading to better generalization.
3. Initialization Robustness: Multiple random initializations per fold mitigate the impact of poor initialization on model performance.
4. Implicit Regularization: The ensemble structure acts as an implicit regularization mechanism, reducing overfitting to the training data.

### 6.2.2 Single Model Limitations

The single best model's performance gap between validation and test sets highlights several limitations:

1. Overfitting: Despite regularization techniques, the single model shows signs of overfitting to the training/validation data.
2. Higher Variance: The single model is more sensitive to the specific patterns in the training data, leading to less stable performance on unseen data.
3. Limited Representation: A single model may struggle to capture the full complexity of rainfall patterns across different geographical and temporal contexts.

### 6.3 Computational Considerations

While the ensemble approach delivers superior performance, it comes with increased computational requirements:

1. Training Time: The ensemble approach requires training 25 models instead of 1, increasing training time by approximately 25x.
2. Storage Requirements: Storing weights for 25 models requires significantly more storage space than a single model.
3. Inference Complexity: Making predictions with the ensemble requires running 25 forward passes through different models, increasing inference time.

These computational trade-offs should be considered when deploying the model in production environments. However, the substantial performance improvements offered by the ensemble approach generally justify the increased computational requirements for applications where prediction accuracy is paramount.

## 7. Limitations and Areas for Improvement

1. Computational Requirements: The ensemble approach's superior performance comes at the cost of increased computational requirements for both training and inference.
2. Data Dependencies: The model's performance is dependent on the quality and resolution of input data, particularly climate reanalysis and DEM data.

3. Temporal Scope: The current implementation uses monthly temporal resolution, which may not capture shorter-term rainfall dynamics.
4. Spatial Transferability: The model's performance when transferred to different geographical regions remains to be thoroughly evaluated.
5. Evaluation Metrics: While this study reports R², RMSE, and MAE, incorporating additional metrics would provide a more comprehensive assessment:
    - Occurrence Metrics: Convert rainfall to a binary series (1 if >0 in, 0 otherwise) and compute the Jaccard Index and frequency bias.
    - Intensity Metrics: Restrict to wet days only (rainfall >0 in) and compute Mean Bias , intensity MAE, and Pearson correlation coefficient between R_est and R_obs.

## 8. Conclusion

In conclusion, this project has demonstrated the power of combining the LAND methodology with ensemble learning techniques for rainfall prediction. The achieved performance (test R² of 0.7955) represents a significant advancement in rainfall prediction capabilities and provides a strong foundation for future research and applications in this domain.

## References

-Hatanaka, Y., Indika, A., Giambelluca, T., & Sadowski, P. (2024). Statistical Downscaling of Climate Models with Deep Learning. March 2024.

-Sanfilippo, K., Elison Timm, O., Frazier, A.G., & Giambelluca, T.W. (2023). Effects of systematic predictor selection for statistical downscaling of rainfall in Hawaiʻi. International Journal of Climatology, 44:571–591.