International Journal of Data Science and Analysis
2026; 12(5): 114-128
http://www.sciencepublishinggroup.com/j/ijdsa
doi: 10.11648/j.ijdsa.20261205.12

Comparative Study of Lstm, Xgboost and Hybrid Lstm-xgboost Models for Rainfall Forecasting in Semi-arid Regions

Rony Mutugi Muriithi, Peter Kinyua Gachoki, Mutua Kilai*

School of Pure and Applied Sciences, Kirinyaga University, Kerugoya, Kenya

Email address:
ronymu2gi@gmail.com (Rony Mutugi Muriithi), pgachoki@kyu.ac.ke (Peter Kinyua Gachoki), kmutua@kyu.ac.ke (Mutua Kilai)

To cite this article:
Rony Mutugi Muriithi, Peter Kinyua Gachoki, Mutua Kilai. (2026). Comparative Study of Lstm, Xgboost and Hybrid Lstm-xgboost Models for Rainfall Forecasting in Semi-arid Regions. *International Journal of Data Science and Analysis, 12*(5), 114-128. https://doi.org/10.11648/j.ijdsa.20261205.12

Received: 11 July 2026; **Accepted:** 28 July 2026 **Published:** 8 September 2026

Abstract: Rainfall forecasting remains challenging in semi-arid regions due to high variability and intermittent rainfall patterns. Statistical forecasting methods such as Autoregressive Integrated Moving Average(ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) often struggle to capture the non-linear dynamics typical of such rainfall. Machine learning (ML) techniques such as Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM) networks offer improved forecasting performance but have individual limitations. This study compares LSTM, XGBoost and a hybrid LSTM-XGBoost model for monthly rainfall forecasting, using SARIMA as a baseline. The study utilizes 44 years of CHIRPS (Climate Hazards Group InfraRed Precipitation with Station) data from Tana River County, Kenya, accessed through Google Earth Engine. Data preprocessing included log transformation, stationarity testing, normalization, and feature engineering. The data was chronologically split to form the train, validation and test sets. The performance of the models was evaluated using Root Mean Square Error, Mean Absolute Error, Coefficient of Determination and Nash Sutcliffe Efficiency. Shapley Additive Explanations were used for interpretability. The XGBoost, LSTM and Hybrid LSTM-XGBoost models outperformed the baseline, achieving a Coefficient of Determination of approximately 0.61 compared with 0.52. The hybrid model performed best overall, with an RMSE of 38.68 mm, MAE of 17.23 mm, and R2 of 0.609. Hybrid forecasting approaches combining deep learning and machine learning should be further explored for rainfall prediction in semi-arid regions due to their potential to capture complex temporal patterns.

Keywords: Rainfall Forecasting, LSTM, XGBoost, LSTM-XGBoost Hybrid Model, Machine Learning

1. Introduction

Rainfall variability plays a critical role in regulating the hydrological cycle in sub-Saharan Africa. This variability influences availability of water consequently impacting agricultural productivity. Kenya's Arid and Semi-Arid Lands (ASALs) face two main challenges, which are frequent droughts and severe flood events. Recent studies have shown that climate change is worsening these extremes, with serious effects on food security and infrastructure resilience [1].

Eastern Africa exhibits complex climate pattern often referred to as the "East Africa Paradox". Global climate models project increased rainfall under anthropogenic(human caused) warming whereas observation records show a drying trend during the crucial "Long Rains" season. This discrepancy presents the challenge of accurate prediction of rainfall in the region and emphasizes the need for robust, localized forecasting systems. This study focused on Tana River County, situated within the lower catchment of the Tana River Basin, the largest basin in Kenya, covering approximately 100,000 km^2. It represents a semi-arid floodplain ecosystem.

The hydrology of Tana River County is influenced not only by local rainfall but also by flood flows originating from upstream regions. The county experiences a bimodal rainfall pattern, with an average annual precipitation of about 370 mm which is far below the 2,200mm received in the highland regions of Kenya. Despite its limited rainfall, the county supports major irrigation projects including Tana (Hola) and Bura Irrigation Schemes which are for targeted for expansion under Kenya's Vision 2030 [2]. However, ground-based meteorological data in the county are relatively sparse and fragmented, making satellite-based datasets like CHIRPS essential. Previous studies have demonstrated the reliability of CHIRPS rainfall estimates in Eastern Africa, with good accuracy and low bias, making it suitable for rainfall forecasting in data-scarce regions [3].

Rainfall forecasting has historically relied on statistical time-series models. The Autoregressive Integrated Moving Average (ARIMA) model has been a widely applied approach. It forecasts future values as linear combinations of past observations and past forecast errors [4]. Although differencing allows ARIMA to handle non-stationary series, the model is primarily based on linear assumptions and may have limitations when applied to highly variable rainfall processes. Applying stationary models to non-stationary rainfall data can lead to underestimation of extreme events, posing risks to infrastructure and disaster management [4, 5]. Moreover, classical statistical models are often unable to capture the complex, non-linear nature of atmospheric processes [4, 6, 7]

Recent studies have explored machine Learning and deep learning methods as alternatives for rainfall forecasting. Extreme Gradient Boosting (XGBoost), an ensemble machine learning algorithm has shown strong performance in regression problems. It captures non-linear relationships and handles sparse datasets through regularization techniques [5, 8]. Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, address the "vanishing gradient" problem by using gating mechanisms to retain information over long periods. This makes them well-suited for modeling seasonal lag effects in rainfall [9, 10]

Despite their advantages, standalone machine learning models still have limitations. LSTMs networks are effective in learning temporal dependencies but are computationally intensive and may struggle to accurately reproduce extreme rainfall peaks. On the other hand, XGBoost can effectively model complex relationships but does not inherently account for sequential dependencies within time-series data. To address this limitations hybrid approaches such as LSTM-XGBoost have been proposed. LSTM-XGBoost hybrid model, for example, combines temporal learning capabilities of LSTM with the robust regression capabilities of XGBoost for error correction. Hybrid models aim to reduce systematic errors and improve forecasting accuracy beyond the constituent standalone models [6, 11].

This study compares the performance of LSTM, XGBoost and a hybrid LSTM-XGBoost model against a Seasonal ARIMA baseline for rainfall forecasting in Tana River County under complex semi-arid rainfall dynamics.

2. Literature Review

2.1. Classical Statistical Forecasting Methods

Historically, statistical time-series models have long been used for rainfall forecasting in meteorology and hydrology.Among these, the Autoregressive Integrated Moving Average (ARIMA) model is a widely applied forecasting technique. The ARIMA model assumes that future observations in a time series are a linear combination of past values and previous forecast errors. [4] explain that the model integrates three key components. The Autoregression (AR) component regresses the variable of interest on its own lagged values, assuming that past rainfall values contain information about future rainfall. Integration (I) component addresses non-stationarity. A time series is regarded as stationary when its statistical characteristics do not change over time. Most hydrological time series, including rainfall, are non-stationary due to seasonal variability and long-term climatic trends. The integration component applies differencing d times to transform the series into a stationary form, stabilizing the mean. Moving Average (MA) models the current observation as a linear combination of past forecast errors. This allows the model to capture short-term irregularities and random variation in the data.

The Box Jenkins methodology on which ARIMA based models are founded on follows a systematic three-stage approach: (i) model identification through analysis of autocorrelation function (ACF) and partial autocorrelation functions (PACF), (ii) parameter estimation, typically through maximum likelihood estimation and (iii) diagnostic checking of residuals to ensure model adequacy.

In hydrological time series modeling, the Seasonal ARIMA (SARIMA) variant has been used extensively to capture seasonal patterns in precipitation data. SARIMA extends ARIMA by incorporating seasonal autoregressive, differencing and moving average components represented as $\mathtt{SARIMA}(p,d,q)\ (P,D,Q)_s$. The lowercase parameters represent non-seasonal patterns, while the uppercase parameters capture seasonal effects at a fixed seasonal period s. Dimri et al. [12] successfully applied SARIMA to model 100-year-long precipitation trends in the Bhagirathi River Basin. An optimal $\mathtt{SARIMA(0,1,1)(0,1,1)}_{12}$ was identified in their study and generated 20-year forecasts. Dastorani et al. [13] compared various classical time series models (AR, MA, ARMA, ARIMA) in a semi-arid region of Iran and reported that, although performance varied by station, an $\mathtt{ARIMA(1,1,2)}$ model yielded the best results for a specific monthly rainfall datasets.

Despite their widespread adoption, ARIMA based models face critical limitations when applied to the nature of semi-arid rainfall. A fundamental requirement for the ARIMA framework is stationarity. It assumes that the statistical characteristics of a time series such as mean are invariant over time. **(author?)** [4] highlight that this assumption becomes increasingly restrictive for real-world hydrological systems characterized by climate change. They argue that conventional stochastic approaches like ARIMA cannot adequately capture the evolution of systems exhibiting non-linearity and non-stationarity.

Additionally, ARIMA models are inherently linear. However, the atmospheric processes generating rainfall involve highly non-linear interactions between temperature, pressure, humidity and topography [7]. Wani et al. [14] conducted a comparative study in the complex topography of the North-Western Himalayas and found that ARIMA consistently ranked lowest in accuracy compared to Machine Learning(ML) and Deep Learning(DL) models. The study concluded that ARIMA failed to capture the non-linear dependencies and sudden climatic changes characteristic of rainfall in mountainous and semi-arid regions.

2.2. Machine Learning Approaches

Machine learning methods are increasingly being used in rainfall forecasting because they can identify patterns in historical rainfall data that may not be easily captured by traditional statistical models. While statistical approaches often depend on assumptions about how rainfall data behaves, machine learning models learn relationships from past observations and use them to make predictions. Techniques such as ensemble learning and deep learning have gained attention in environmental studies because they can handle the irregular and changing nature of rainfall patterns.

Ensemble learning methods combine multiple base learners to generate a final prediction. The underlying principle is that combining several models can improve predictive performance by reducing the influence of individual model limitations. Common ensemble strategies include bagging, boosting, and stacking. Kundu et al. [15] identified these approaches as widely used ensemble techniques in environmental prediction tasks. Their review indicated that boosting methods have been applied more frequently and successfully in rainfall forecasting compared with other ensemble approaches.

Extreme Gradient Boosting (XGBoost) builds upon the gradient boosting framework by incorporating techniques that make model training faster and more resource-efficient. Its ability to model complex nonlinear relationships and handle missing or sparse data is particularly useful for rainfall forecasting in semi-arid conditions. **(author?)** [5] describe XGBoost as a scalable tree-boosting system that minimizes a regularized objective function to reduce overfitting which is a capacity that lacks in standard decision trees.

The application of XGBoost in rainfall prediction has been investigated in several studies. **(author?)** [8] compared XGBoost with Multivariate Linear Regression (MLR) and Random Forest (RF) for daily rainfall prediction and reported that XGBoost achieved the lowest RMSE and MAE values among the evaluated models. In a systematic review of water resources applications, Niazkar et al. [16] reported that XGBoost and hybrid XGBoost models produced the best results in 74% of the reviewed studies. These studies demonstrate the suitability of XGBoost for rainfall-related prediction tasks. However, the model does not have an internal mechanism for learning sequential dependencies since observations are treated as independent inputs. This limits its ability to directly represent temporal patterns commonly present in rainfall time series.

Deep learning refers to a class of machine learning methods that rely on deep neural network architectures to learn representations directly from data. LeCun et al. [17] describe it as a framework of layered computational models capable of learning progressively complex patterns. Unlike classical approaches, deep learning minimizes the need for manually designed features.

Artificial Neural Networks (ANNs) form the basis of deep learning algorithms. They provide a foundation for modeling complex patterns through interconnected layers of computational nodes. Standard feedforward neural networks lack the capacity to process sequential information where current values depend on past observations [9]. Recurrent Neural Networks (RNNs) overcome this limitation by using feedback connections that allow information from previous times to influence current predictions. This makes ANN them suitable for time-series modeling. However, standard RNNs suffer from the vanishing gradient problem, where error signals decay over long sequences. This limits their ability to learn long-term temporal dependencies such as seasonal rainfall patterns [10]. To overcome this limitation, the Long Short-Term Memory (LSTM) network was developed as a specialized deep learning model for capturing long-term sequential dependencies.

The LSTM overcomes the vanishing gradient problem through a unique cell structure composed of gates that regulate the flow of information which allows the network to retain or discard memory selectively. **(author?)** [9] detail the LSTM architecture. It includes a forget gate to determine which previous information is discarded, an input gate to regulate new information added to the cell state and an "output gate" to determine the prediction.

The effectiveness of LSTM models has been demonstrated in several studies. Ouma et al. [18] applied LSTMs to rainfall trend analysis in Kenya and found they outperformed Wavelet Neural Networks (WNNs) by effectively capturing non-linear seasonality and chaotic trends. Specifically, LSTM predicted mean monthly rainfall with R^2 value of 0.8610 while WNN had $R^2 = 0.7825$. Moreover, Siami-Namini et al. [19] found that LSTM models reduced error rates by an average of 84% to 87% compared to ARIMA models in financial time series forecasting. Although conducted in a different field, the results demonstrate the ability of LSTM models to learn complex temporal relationships. Wani et al. [14] further highlights the ability of LSTMs to learn non-linear relationships between past and future observations that linear models such as ARIMA may miss

Despite the strengths of LSTMs, they can be computationally intensive and at times they may struggle to accurately represent extreme rainfall events [10]. These limitations have motivated the development of hybrid models.

2.3. Hybrid Models

Hybrid models leverage the "No Free Lunch" theorem, meaning no single model performs best across all problems [7]. Hybrid models combine complementary modeling approaches to exploit the strengths of the individual models while mitigating their respective limitations, potentially improving forecasting performance[4]. Slater et al. [20] reviewed hybrid forecasting and noted that blending dynamical climate predictions with machine learning allows for the correction of systematic errors while leveraging data-driven accuracy.

Semmelmann et al. [11] proposed a hybrid model for load forecasting where the LSTM extracts temporal features from the sequence data which are then fed into an XGBoost model to perform the final regression and peak prediction. Using this approach LSTM handles temporal sequence and history while XGBoost handles the magnitude and non-linear interactions of features. In rainfall forecasting, this helps address the common issue where LSTMs capture the timing of rain well but underestimate the intensity of extreme storms.

Hybrid models such as Convolutional Neural Network (CNN)-XGBoost have proven effective in reducing forecast errors. These models can be made transparent using Explainable AI (XAI) techniques. **(author?)** [6] developed a hybrid CNN-XGBoost model for rainfall prediction in Bangladesh, using SHapley Additive exPlanations (SHAP) to reveal that specific lagged rainfall variables were the most significant predictors. Their results showed that the hybrid approach achieved RMSE of 0.65 mm/day and MAE of 0.28 mm/day on satellite datasets outperforming standalone models. The SHAP analysis revealed that the most influential predictors included CNN-extracted spatial features and engineered rainfall variables (such as precipitation residuals and logarithmic transformations). These results demonstrate the value of XAI methods in interpreting hybrid rainfall forecasting models and understanding the factors influencing their predictions.

Decomposition-based hybrid frameworks have been effective in modeling the non-linear and non-stationary nature of rainfall. Saikia et al. [21] introduced a Wavelet-SARIMA-Transformer (W-ST) model that uses the Maximal Overlap Discrete Wavelet Transform to split rainfall data into components: linear parts modeled with SARIMA and non-linear parts handled by Transformer networks. When tested across Northeast India, the Haar-wavelet-based hybrid consistently outperformed individual models.This shows that using wavelet-based decomposition together with both linear models and deep learning models can greatly improve rainfall and climate forecasting.

2.4. Research Gap

A lot of research work on hybrid rainfall forecasting models focuses on high-rainfall monsoon regions or temperate climatic regions. For instance, **(author?)** [6] focused on Bangladesh and Wani et al. [14] focused on the Himalayas. There are relatively few studies that implement the LSTM-XGBoost model to the intermittent and highly variable rainfall patterns characteristic of Kenya's semi-arid lands where drought monitoring is as critical as flood forecasting. Furthermore, the "East Africa Paradox" noted by **(author?)** [1] creates a unique challenge where models must reconcile drying trends in "long rains" with potential intensification of extreme rainfall events. This complexity remains underexplored in current hybrid forecasting literature especially in the context of Tana River County, Kenya.

3. Materials and Methods

3.1. Data Collection

This study used 44 years (January 1981 - December 2024) of rainfall data from Tana River County, located within the Lower Tana River Basin, a semi-arid region characterized by bimodal rainfall patterns [2]. The data were obtained from Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS) dataset. The county boundary was defined using the FAO GAUL (Global Administrative Unit Layers) administrative boundary dataset as a geographic filter to extract daily rainfall data specific to the study region. The extraction was done within Google Earth Engine.

3.2. Data Preprocessing

The data were aggregated to monthly totals by summation, consistent with the monthly forecasting objective of this study. The aggregation produced a 528-month series spanning January 1981 to December 2024.

They were then subjected to quality control checks to identify missing values and anomalies. No missing values existed in the dataset. Outliers were screened using the Interquartile Range (IQR) method prior to model development.

For the ARIMA baseline, the series was tested for stationarity using the Augmented Dickey–Fuller (ADF) test, with differencing applied where required to stabilise the mean and variance.

3.2.1. Data Partitioning

The data was split into three separate subsets chronologically to maintain the sequence of observations and prevents future information from influencing model training. The training set covered January 1981 to December 2015 (420 months, 79.5%), the validation set covered January 2016 to December 2021 (72 months, 13.6%), and the test set covered January 2022 to December 2024 (36 months, 6.8%). All model evaluation metrics were computed on the test set. Establishing the partition before scaling and feature construction ensured that all subsequent preprocessing parameters were derived from the training data alone.

3.2.2. Transformation and Scaling

A log1p transformation, $x' = \log(1 + x)$, was applied to the precipitation series prior to LSTM training. This compressed the right-skewed distribution so that extreme wet months did not disproportionately dominate the mean squared error loss during gradient-based optimisation. The inverse transformation, $x = \exp(x') - 1$, was applied to all LSTM outputs to recover predictions in original millimetre units.

Following transformation, the input features and the log-transformed target were standardised using z-score scaling:

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation of the feature. The scaler was fitted exclusively on the training partition, so that information from the validation and test periods did not influence the normalisation parameters and the integrity of the temporal evaluation was preserved. Standardisation is required for the LSTM to prevent features with larger magnitudes from dominating gradient descent and to ensure faster convergence [9]. XGBoost, by contrast, was trained directly on the raw precipitation values in millimetres, as tree-based ensembles are insensitive to feature scale and to skewed target distributions and do not require a variance-stabilising transformation.

3.2.3. Feature Engineering

Eighteen predictor features were constructed for both the LSTM and XGBoost models. For the LSTM these were derived from the log-transformed series, and for XGBoost from the raw precipitation series in millimetres. The feature set comprised twelve monthly lag variables (lag-1 to lag-12), rolling mean precipitation over the preceding three and six months, rolling standard deviation over the same windows, and cyclic encodings of the calendar month using sine and cosine transformations. The cyclic encodings represent the periodic nature of the annual cycle and prevent the artificial discontinuity between December and January that would arise if the month were treated as a linear integer.

3.2.4. Sequence Construction for the LSTM

Unlike ARIMA and XGBoost, which operate on tabular feature vectors, the LSTM requires input structured as temporal sequences. Each example was built using a sliding window of the preceding 24 months of features to predict precipitation in the following month. The 24-month window was selected to span two complete annual cycles which allows the network to learn both intra-annual seasonality and inter-annual variability. This process transformed the two-dimensional feature matrix of shape (months, features) into a three-dimensional tensor (samples, timesteps, features) resulting in the shape (384, 24, 18) for the training data.

To maintain temporal continuity, the final 24 months of the preceding subset were prepended to each of the validation and test sets before sequence generation. This provided the initial predictions in each evaluation period with sufficient historical information while maintaining the chronological separation of the datasets.

3.3. Model Development

3.3.1. ARIMA Model

An Autoregressive Integrated Moving Average (ARIMA) model was used as the baseline for comparison. This linear model was selected for its established application in modeling univariate time series by the combination of autoregression (AR) and moving averages (MA) after differencing(I) to achieve stationarity [4]. The ARIMA(p, d, q) model can be expressed in compact form as:

$$\phi(B)(1 - B)^d y_t = \theta(B)\epsilon_t \tag{2}$$

where B is the backshift operator ($By_t = y_{t-1}$), $\phi(B)$ and $\theta(B)$ are the autoregressive and moving average polynomials of orders p and q respectively, d is the differencing order and ϵ_t represents white noise.

3.3.2. XGBoost

Extreme Gradient Boosting (XGBoost) builds multiple decision trees in a sequence, allowing each new tree to learn from and correct the errors of the previous trees. The final prediction is obtained as the sum of predictions from all K trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{3}$$

where f_k represents individual tree predictions and x_i are the engineered input features for sample i. For time series forecasting, XGBoost requires transforming the temporal problem into a supervised learning task through feature engineering.

Specifically, lag features were created as:

$$X_t^{(lag-k)} = y_{t-k}, \quad k \in \{1, 2, \ldots, L\} \tag{4}$$

where y_t is rainfall at time t and L is the maximum lag order. These historical values serve as input features to predict the target rainfall value at time t.

3.3.3. LSTM

The LSTM network followed a sequential architecture comprising two stacked LSTM layers. Each layer was followed by a dropout layer for regularisation and two dense layers culminating in a single-unit output. The input layer received feature sequences spanning a 24-month lookback window. The inputs were structured as a three-dimensional tensor of shape (samples, 24, 18), where each timestep carried the eighteen engineered features. They consist twelve monthly lag values (the rainfall at each of the preceding twelve months), rolling means over the preceding three and six months, rolling standard deviations over the same windows and sine and cosine encodings of the calendar month. The first LSTM layer comprised 64 units and returned full sequences to a second LSTM layer of 32 units, which returned only its final hidden state. A dropout rate of 0.2 was applied after each LSTM layer. The representation that resulted passed through a dense layer of 16 units with Rectified Linear Unit (ReLU) activation. This was followed by a final dense layer with a single unit and linear activation that produced the monthly rainfall prediction. The complete architecture comprised 34,209 trainable parameters. The final LSTM architecture was selected following preliminary experimentation, with a deliberately modest capacity chosen to balance model flexibility and generalisation.

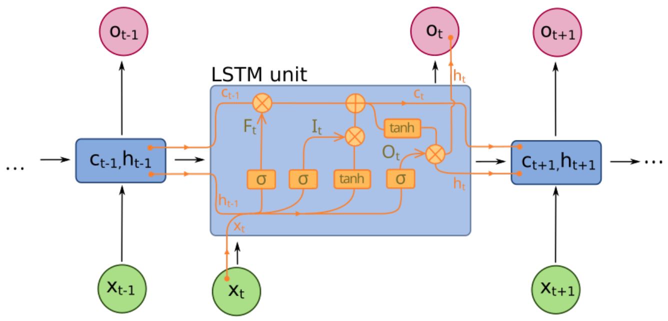

Figure 1. *LSTM unit architecture.*

The LSTM architecture maintains a cell state C_t and hidden state h_t through three gating mechanisms. At each time step, the network first decides which parts of its accumulated memory are no longer useful through a forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

New information is selectively incorporated via an input gate, which works alongside a candidate memory vector:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{6}$$

The cell state is then updated by combining weighted old and new information:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{7}$$

The network then determines how much of the updated cell state to pass forward as the output hidden state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad h_t = o_t \odot \tanh(C_t) \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\tanh(\cdot)$ is the hyperbolic tangent, $\odot$ denotes element-wise multiplication, W are weight matrices, b are bias vectors, x_t is the input at time t and h_{t-1} is the previous hidden state. These gates act as information valves, with sigmoid outputs between 0 and 1 controlling the flow of information through the network.

For rainfall forecasting, sequences of length k was structured as $\mathbf{X}_t = [y_{t-k}, y_{t-k+1}, \ldots, y_{t-1}]$ with target y_t, creating a supervised learning problem where the model learns the mapping from past rainfall values to future predictions. The LSTM model used a sequential architecture with LSTM layers followed by a dense output layer. Dropout regularization was applied to reduce overfitting. The model was trained using backpropagation through time, minimizing the Mean Squared Error (MSE) loss function:

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{9}$$

where y_i represents actual rainfall, $\hat{y}_i$ is the predicted value and n is the number of samples. Optimization was performed using the Adam optimizer, with dropout regularization randomly deactivating neurons during training to force the network to learn robust features that generalize to unseen data.

3.3.4. Development of the Hybrid Model

The hybrid LSTM-XGBoost model was developed through a sequential residual-correction process that combines the temporal learning strength of the LSTM with the error-correcting capability of XGBoost. The architecture follows a two-stage approach in which the LSTM serves as the primary forecasting model and XGBoost is trained to model the residual errors produced by the LSTM which allows it to learn systematic patterns that remain unexplained by the initial forecasts.

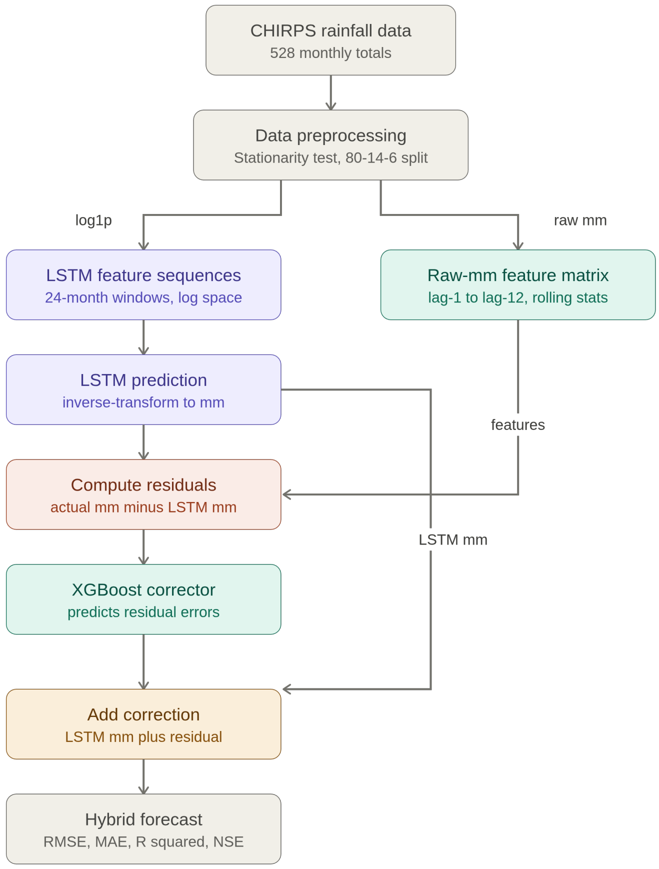

Figure 2. *Hybrid Model Design*

Stage 1: LSTM Based Prediction. The LSTM network was trained on the log-transformed and scaled rainfall sequences to produce a primary monthly precipitation forecast. The network learned the temporal dependencies in the input sequences using backpropagation through time with a mean squared error loss. Its scaled outputs were inverse-transformed (reversing both the standardisation and the log1p transformation) to recover predictions in original millimetre units. These predictions were generated for the training, validation, and test partitions.

Stage 2: Residual Computation. The residual error of the LSTM was computed as the difference between the observed precipitation and the LSTM prediction expressed in millimetres for each month in the combined training and validation period. These residuals represent the component of rainfall variability that the LSTM failed to capture, and constitute the target variable for the correction model.

Stage 3: XGBoost Residual Correction. An XGBoost regressor was then trained to predict the residual errors produced by the LSTM using the engineered feature set derived from the raw rainfall series. These features included twelve monthly lag variables, rolling means, standard deviations and cyclic month encodings.The XGBoost model was designed to learn patterns in the LSTM's prediction errors in order to identify situations in which the LSTM tended to systematically over-predict or under-predict rainfall. Key hyperparameters including the number of estimators, maximum tree depth and learning rate were optimised using time-series cross-validation.

Stage 4: Model Optimization. The final hybrid forecast was obtained by adding the XGBoost residual correction to the LSTM base prediction, with the result constrained to non-negative values for physical validity. The two components were optimised independently. The LSTM architecture (layers, units, dropout, sequence length) was tuned first, after which the residual corrector was tuned against the fixed LSTM outputs.

3.4. Performance Metrics

Four complementary metrics were used to quantitatively assess forecasting accuracy and enable model comparison.

Root Mean Square Error (RMSE) measures the average magnitude of prediction errors, with higher penalty for large errors:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

Mean Absolute Error (MAE) measures the average absolute deviation between predicted and observed values:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{11}$$

Coefficient of Determination (R^2) represents the proportion of variance in rainfall explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{12}$$

Nash-Sutcliffe Efficiency (NSE) assesses the predictive skill relative to a simple mean baseline:

$$NSE = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{13}$$

where y_i represents observed rainfall at time i, $\hat{y}_i$ is the corresponding predicted value, $\bar{y}$ is the mean of observed values and n is the total number of observations in the test set. RMSE and MAE quantify prediction error magnitude in the original units (mm), with RMSE assigning greater penalty to the larger errors due to the squaring operation. R^2, the coefficient of determination, measures the proportion of variance explained by the model. NSE evaluates predictive skill relevant to the mean of the observed data explained variance. For both R^2 and NSE, values closer to 1 indicate better predictive performance. Lower RMSE and MAE values indicate higher accuracy. Models were compared using all four metrics on the test set supplemented by visual analysis through plots

To improve the interpretation of model predictions and understand the factors influencing the decision-making process, explainable AI techniques were applied. SHapley Additive exPlanations (SHAP) was used to quantify the contribution of individual features to the model output and provide explanations for specific predictions. By this analysis, the study assessed whether the hybrid model relied primarily on temporal patterns learned by the LSTM or on the engineered features, providing insights into the factors influencing rainfall predictability in the study region.

4. Results and Discussion

4.1. Development of the Baseline SARIMA Model for Monthly Rainfall Forecasting in Tana River County

A Seasonal ARIMA(0,0,2)(2,0,1)[12] model, identified through stepwise Akaike Information Criterion (AIC) minimisation, was used as the linear baseline. On the test set (January 2022 to December 2024), it achieved an RMSE of 43.04 mm, an MAE of 21.88 mm and an R^2 of 0.516.

Figure 3 presents the Seasonal ARIMA forecasts against the observed monthly rainfall over the test period, together with the 95 percent forecast intervals. The model captured the broad seasonal pattern of alternating wet and dry months. However, it consistently underestimated peak rainfall events. This is reflected in the relatively high Mean Absolute Error. Although the forecast intervals were wide, the most extreme observed event in the test period, about 308 mm in late 2023, lay outside the upper bound. This shows that the model did not adequately capture the full intensity of extreme rainfall variability.

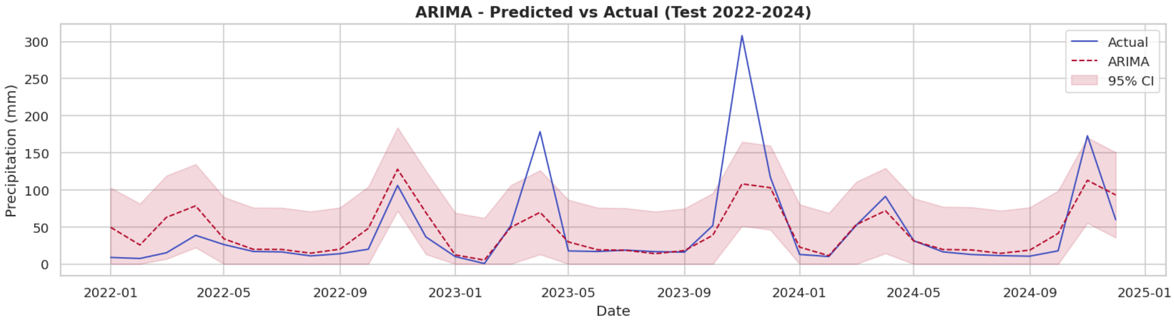

Figure 3. *Seasonal ARIMA forecasts and observed monthly rainfall on the test set (2022–2024).*

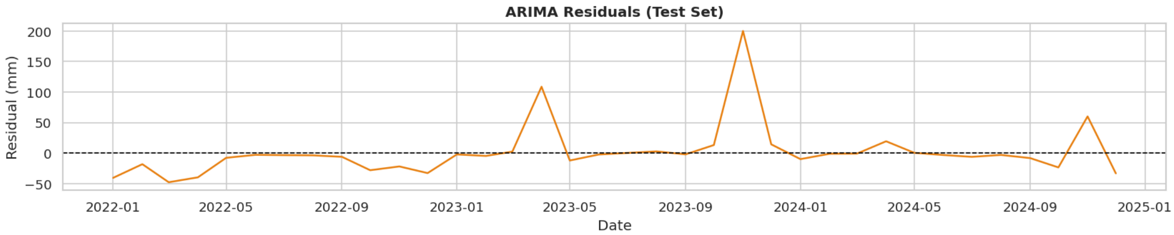

Figure 4. *Seasonal ARIMA residuals on the test set, 2022–2024.*

The residuals (Figure 4) show a clear pattern rather than random variation, with large positive errors occurring during wet season peaks. These include errors above 100 mm in March 2023 and above 200 mm in late 2023, while residuals during dry months remain small. This indicates that the model mainly under-predicted extreme rainfall rather than balancing over and under predictions hence suggesting that important nonlinear information remains in the data. This finding aligns with known limitations of linear models when applied to rainfall processes that are both non linear and highly variable **(author?)** [4] and with Wani et al. [14] who report that classical time series models like ARIMA show the weakest performance among statistical, machine learning and deep learning methods in similar settings.

4.2. Long Short-Term Memory Model Development for Monthly Rainfall Forecasting in Tana River County

The Long Short Term Memory (LSTM) network consisted of stacked LSTM layers of 64 and 32 units, a dropout rate of 0.2, a dense layer of 16 units and a single linear output layer. This gives a total of 34,209 trainable parameters.

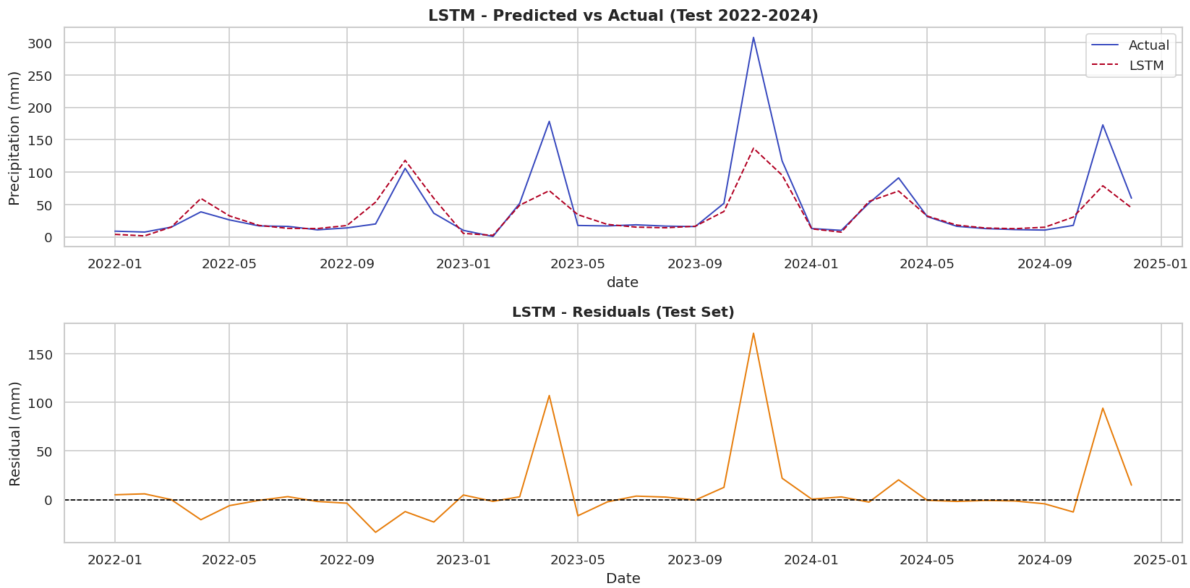

Figure 5. *LSTM forecasts and observed monthly rainfall on the test set (upper panel) with residuals (lower panel), 2022–2024.*

Figure 5 presents the LSTM forecasts (dashed) against the observed monthly rainfall (solid) over the test period in the upper panel, with the prediction residuals in the lower panel. The model followed the seasonal alternation of wet and dry months closely and captured the timing of each rainfall peak. It fell short of the highest peaks, most clearly the late-2023 short-rains maximum of about 308 mm, which it underpredicted by a wide margin. These patterns are reflected in the metrics. The LSTM achieved an RMSE of 38.68 mm, an MAE of 17.27 mm and an R^2 of 0.609 . This represents an improvement over the ARIMA baseline across all metrics, with RMSE reduced by 4.36 mm and explained variance increasing by about nine percentage points. This improvement suggests that the monthly rainfall series contains nonlinear temporal patterns that the linear model could not capture, while the LSTM was able to learn part of this structure.

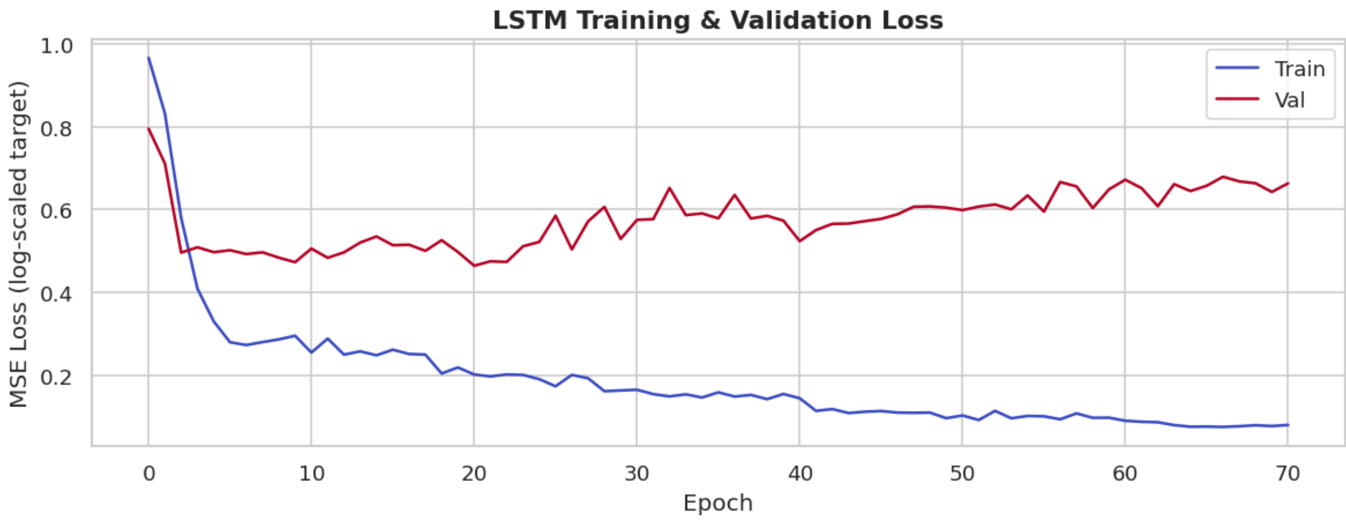

Figure 6. *LSTM Training and Validation Loss.*

The training and validation loss curves (Figure 6) show stable convergence, with the lowest validation loss of 0.464 reached at epoch 21 before early stopping. A noticeable gap between training and validation loss was observed, with training loss falling to about 0.10 and validation loss stabilising near 0.60. Rather than indicating classical overfitting, this gap is more plausibly explained by a shift in data distribution between the training period (1981-2015) and the validation period (2016-2021), which includes a higher frequency of extreme rainfall events. These results are consistent with previous studies. The improvement aligns with findings that LSTM models generally outperform linear approaches in time series forecasting as reported by Siami-Namini et al. [19]. Wani et al. [14] ranks deep learning models above ARIMA in rainfall prediction tasks. However, the magnitude of improvement here is smaller than that reported in lower variability settings. The use of LSTM in hydrological time series is supported by **(author?)** [9] and **(author?)** [10], who highlight its ability to retain seasonal information over long sequences. The R^2 obtained here is lower than that reported by Ouma et al. [18], who achieved 0.861 for monthly rainfall in the Nzoia River Basin. This difference is most likely due to climatic context rather than model weakness. The Nzoia Basin is a humid region with more regular rainfall patterns. Tana River County is semi-arid with much higher variability reflected in a coefficient of variation of about 114%. This higher variability reduces the inherent predictability of the series, meaning that a lower R^2 reflects a more difficult forecasting environment rather than poorer model performance.

4.3. Extreme Gradient Boosting (XGBoost) Model Development for Monthly Rainfall Forecasting in Tana River County

The XGBoost model was trained directly on the raw precipitation series using the eighteen engineered features described in Section 3.2.3, with its main hyperparameters, including the number of estimators, maximum tree depth and learning rate, tuned using time series cross validation with a defined search space. The model achieved an RMSE of 38.68 mm and an R^2 of 0.609, matching the LSTM on both metrics and improving on the ARIMA baseline by the same margin. This is a notable result, as a tree based model without an explicit mechanism for sequence modelling explained the same proportion of variance as the LSTM. The main difference between the two models lies in MAE, where XGBoost recorded 21.68 mm compared to the LSTM's 17.27 mm, indicating larger average errors in monthly predictions.

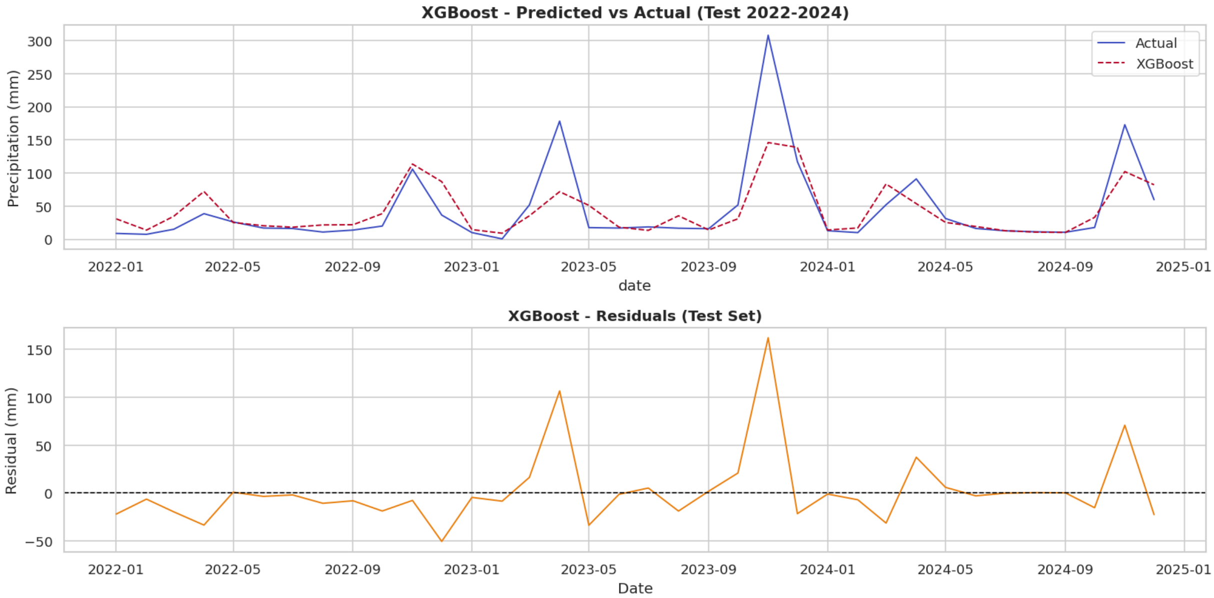

Figure 7. *XGBoost forecasts and observed monthly rainfall on the test set (upper panel) with residuals (lower panel), 2022–2024, Tana River County.*

Figure 7 presents the XGBoost forecasts (dashed) against the observed monthly rainfall (solid) in the upper panel, with the residuals below. Through 2022 the two lines track closely at low values, though the prediction overshoots the minor April peak, reaching about 75 mm against an observed 40 mm. They diverge most at the two largest events, where the observed 178 mm of early 2023 is met with about 73 mm and the 308 mm of late 2023 with roughly 145 mm. At several moderate months the dashed line again sits above the observed series. The residual panel shows this directly, with small negative values where the model over-predicts, dipping to about negative 50 mm, alongside two large positive spikes of roughly 110 mm and 160 mm at the major peaks. Overall, the model captured the timing of seasonal rainfall variations but remained limited in representing the magnitude of extreme events.

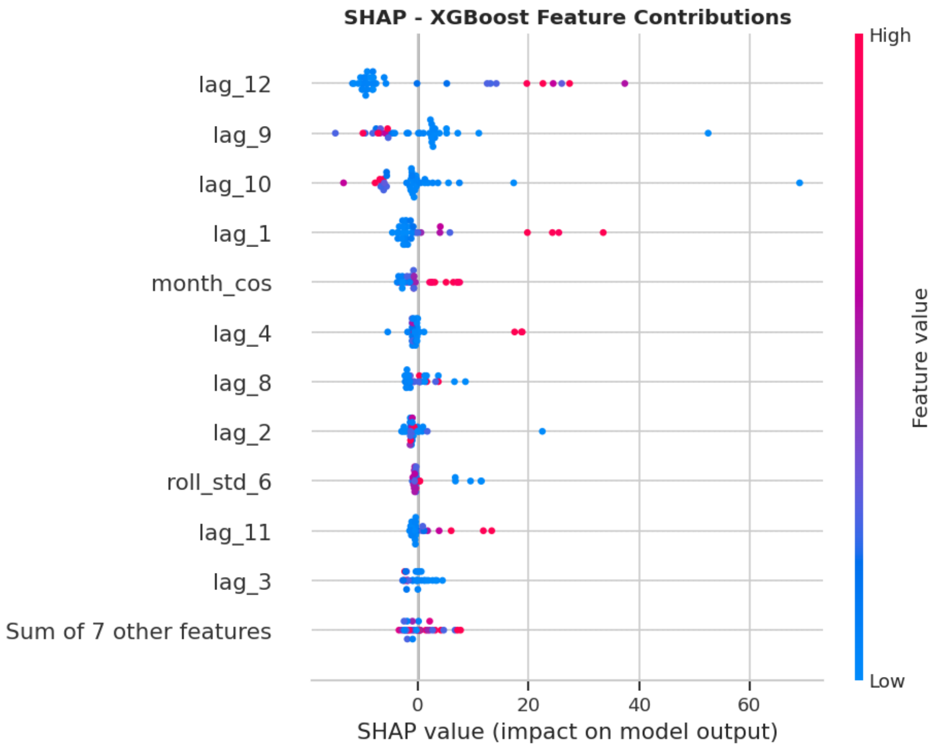

Figure 8. *SHAP feature contributions for the standalone XGBoost model.*

SHAP analysis (Figure 8) shows that the model relied mainly on the twelve month lag, the nine and ten month lags, the one month lag and the cosine month encoding. This indicates that predictions were driven largely by the seasonal cycle and short term rainfall persistence, rather than spurious relationships. The residuals (Figure 7, Lower Panel) show a similar pattern to the other models, with large positive errors at wet season peaks approximately 100 mm in March 2023 and 160 mm in late 2023 confirming that all standalone models struggle with extreme rainfall.

These findings are consistent with literature showing strong performance of tree boosting methods in rainfall prediction. However, the slightly higher MAE compared to expectations is likely due to the limited information available from a single rainfall series, despite the engineered features. Studies such as El Hafyani et al. [22] show that performance improves when additional predictors are included, suggesting that further gains would be possible with multivariate inputs.

Overall, both machine learning models outperformed the SARIMA baseline by a similar margin. They achieved an explained variance of about 0.609 with the LSTM performing better in terms of absolute error. However, both standalone models exhibited similar structured residual patterns and failed to capture the largest wet season peaks. **(author?)** [6] note that while standalone models capture general precipitation trends, they universally struggle with high fluctuations. This leads to severe discrepancies during rainfall peaks. This suggests that extreme rainfall events remain the main source of unexplained variation in the series.

4.4. Development of Hybrid LSTM-XGBoost Model for Rainfall Forecasting

As set out in Section 3.3.4, the hybrid used a two stage residual correction design in which the LSTM produced the primary forecast and an XGBoost model corrected its errors (Figure 2).

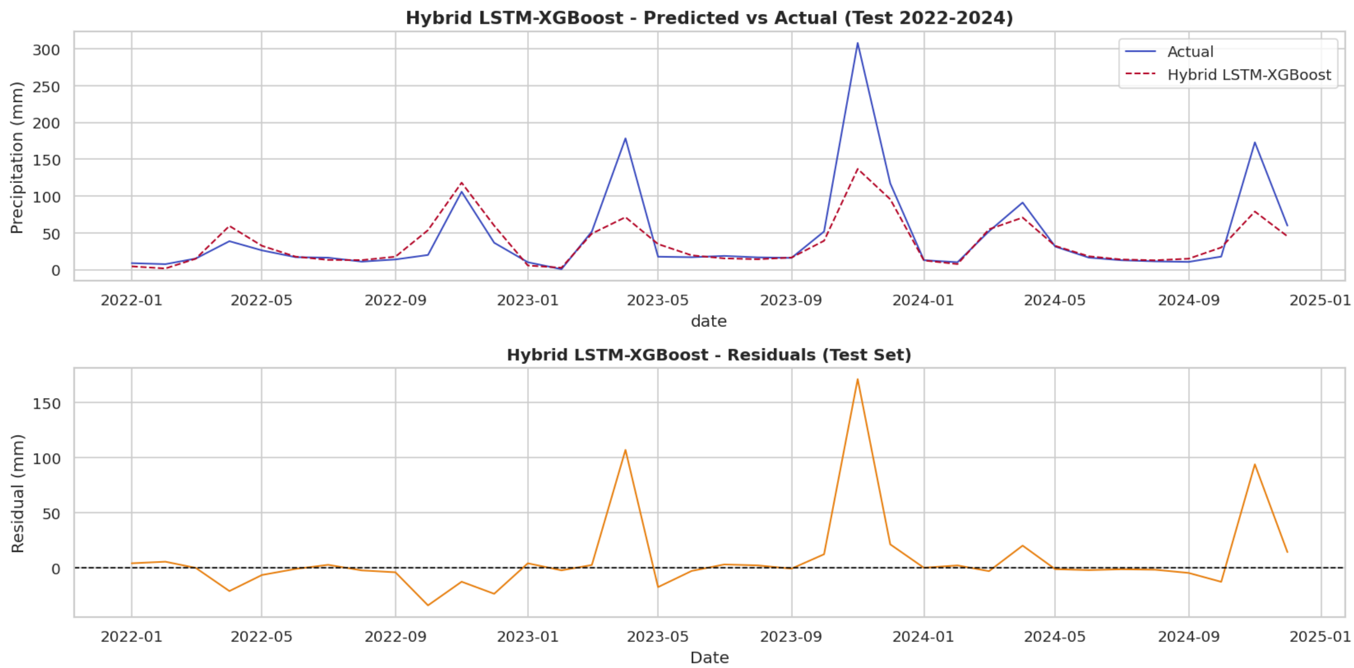

Figure 9. *Hybrid LSTM–XGBoost forecasts and observed monthly rainfall on the test set (upper panel) with residuals (lower panel), 2022–2024, Tana River County.*

Figure 9 presents the hybrid forecasts (dashed) against the observed monthly rainfall (solid) over the test period in the upper panel, with the residuals in the lower panel. The predicted line follows the observed series closely throughout, matching the dry-season troughs almost exactly and departing only slightly above it at a few minor peaks such as April and December 2022. As with the standalone models, the forecasts diverged from the observations only at the largest events. At the early 2023 peak, the model predicted approximately 72 mm against an observed 178 mm, and at the late-2023 maximum it predicted approximately 138 mm against an observed 308 mm. These shortfalls are evident in the residual panel, which remains close to zero across most of the record but rises to approximately 107 mm and 172 mm at the two peaks respectively.

The model achieved an RMSE of 38.68 mm, an MAE of 17.23 mm and an R^2 of 0.609, giving the best overall performance albeit marginal, across all metrics. It also improved on the standalone LSTM, reducing both error measures, although the improvement was small since the LSTM already performed strongly.

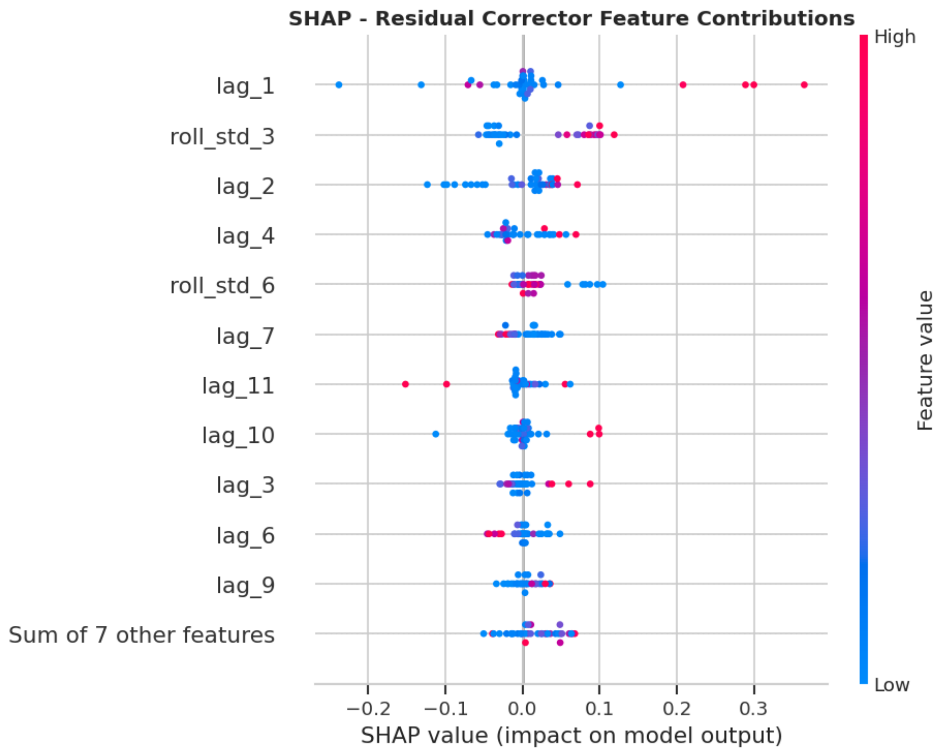

Figure 10. *SHAP feature contributions for the XGBoost residual corrector in the hybrid model.*

The correction stage behaved in a coherent and interpretable way. SHAP analysis of the corrector (Figure 10) shows that it relied mainly on short term signals such as the one month lag, the three month rolling standard deviation and the two and four month lags, rather than the annual cycle features that dominated the standalone XGBoost model. This indicates that the corrector focused on fine scale adjustments left by the LSTM, which is the expected role of a residual model. The remaining errors were concentrated at extreme wet season peaks(Figure 9, lower panel), showing that the correction improved routine deviations but the largest rainfall events remained difficult to predict.

The limited size of the improvement is itself informative. It reflects the strength of the LSTM base model, which had already captured most of the predictable structure in the series, leaving little systematic error for the corrector to learn from. The hybrid therefore achieves the best overall accuracy in the study, while also indicating that the standalone LSTM operates close to the limits imposed by the rainfall data. The Diebold Mariano tests confirm that the hybrid significantly outperforms the ARIMA baseline and the standalone XGBoost model, while performing similarly to the LSTM.

The design follows the approach of Semmelmann et al. [11], who combined LSTM feature extraction with XGBoost regression and the broader rationale of Slater et al. [20], where hybrid models are used to correct systematic errors in base models. Studies such as Latif et al. [23] who increased R^2 from 0.43 to 0.98 using a SARIMA ANN residual hybrid and **(author?)** [6] report larger gains. However, this difference is expected since those studies used less performant base models, leaving more error for the hybrid stage to correct whereas the strong LSTM in this study leaves limited room for improvement. The results are therefore consistent with the principle that residual correction is most effective when the base model is less performant.

4.5. Comparative Evaluation of the LSTM, XGBoost and Hybrid LSTM-XGBoost Models

Table 1. *Comparison of forecasting model performance on the test set (2022–2024).*

Model	RMSE (mm)	MAE (mm)	R^2	NSE
ARIMA	43.041252	21.8807	0.516198	0.516198
LSTM	38.680267	17.271548	0.609270	0.609270
XGBoost	38.681869	21.675314	0.60923	0.60923
Hybrid LSTM-XGBoost	38.679798	17.234052	0.609279	0.609279

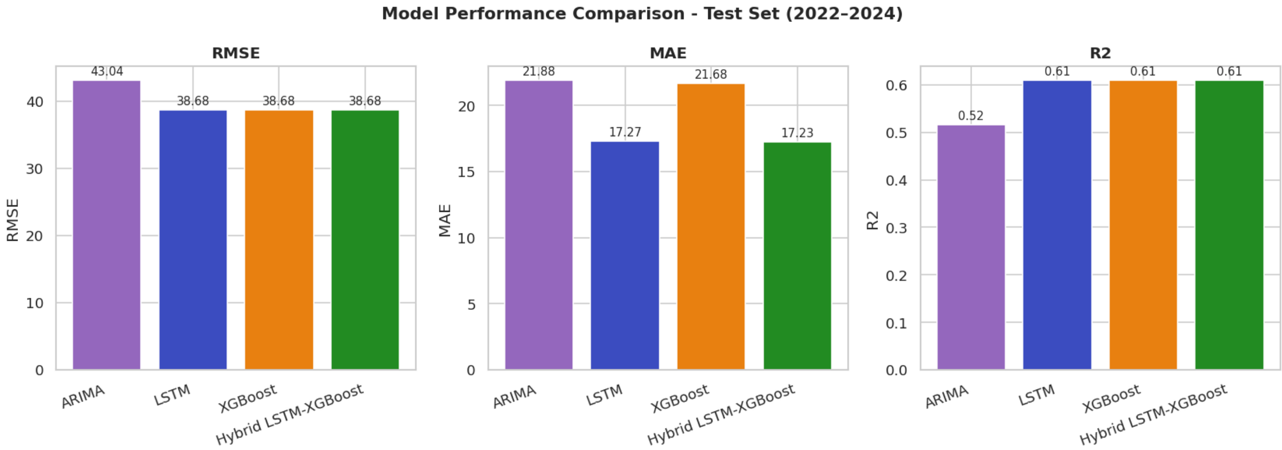

Figure 11. *Test-set (2022–2024) RMSE, MAE and R^2 for the ARIMA, LSTM, XGBoost and hybrid models.*

Table 1 and Figure 11 summarise the test set performance of the four models. The machine learning models LSTM, XGBoost and the Hybrid LSTM-XGBoost, outperform the ARIMA baseline. On RMSE and R^2, the models are effectively identical, each achieving an RMSE of 38.68 mm and an R^2 of 0.609 despite their different structures. This agreement across independent methods suggests a stable result that is not dependent on a single modeling approach. Differences appear mainly in Mean Absolute Error, where the hybrid LSTM-XGBoost model and LSTM record the lowest values (17.23 mm and 17.27 mm). The XGBoost is higher at 21.68 mm close to the Seasonal ARIMA value of 21.88 mm. This indicates that the LSTM based models are more reliable for typical monthly errors.

The Diebold Mariano test using both absolute error and squared error loss with the Harvey Leybourne Newbold correction, provided statistical support for these comparisons. Under absolute error loss, the hybrid significantly outperforms XGBoost (Diebold Mariano statistic, DM = -2.47, $p = 0.019$), and both the hybrid and LSTM models significantly outperform SARIMA ($p = 0.047$ and $p = 0.049$). However, the hybrid model and LSTM model are not significantly different ($p = 0.285$). Under RMSE, no significant differences are observed between the machine learning models (all $p > 0.31$).

The accuracy of the three models, around $R^2 = 0.609$, is best understood in relation to both the climate of the study area and the nature of the data. This value meets the commonly cited threshold of $R^2 > 0.60$ for satisfactory hydrological performance [24]. Admittedly, it is lower than values reported for more humid catchment regions. Ouma et al. [18], for example, reported an R^2 of 0.861 in the humid Lake Victoria region of Kenya. This difference is expected for two main reasons. First, semi-arid rainfall is inherently more variable (intra-seasonally) and intermittent compared to rainfall in humid regions, making it harder to model [2, 23]. Secondly, studies reporting very high accuracy often use multivariate inputs such as temperature, humidity and solar radiation [8]. The models used in this study relied only on rainfall history. The fact that all three models converge to a similar level of accuracy suggests that the result reflects the inherent predictability of the data rather than the limitations of any single method.

5. Conclusions

This study compared the performance of a Seasonal ARIMA model, LSTM, XGBoost and a hybrid LSTM-XGBoost model for monthly rainfall forecasting in Tana River County using CHIRPS rainfall data. The results demonstrate that machine learning methods consistently outperformed the statistical baseline (SARIMA model). This highlights their ability to capture nonlinear temporal relationships in rainfall time series.

The hybrid LSTM-XGBoost model achieved the best overall performance, with an RMSE of 38.68 mm, an MAE of 17.23 mm and an R^2 of 0.609. However, the improvement over the standalone LSTM was marginal. This indicates that the LSTM had already captured most of the predictable structure within the rainfall series. The hybrid residual correction stage therefore provided only modest additional gains, although it remained statistically superior to the standalone XGBoost model under mean absolute-error loss.

All approaches consistently underestimated the peak rainfall events. This finding suggests that historical rainfall alone does not contain sufficient information to fully explain the occurrence of extreme precipitation in the semi-arid environment of Tana River County. Consequently, the forecasting accuracy achieved by all machine learning models likely reflects the inherent predictability limits of the available univariate rainfall data rather than deficiencies in model design.

These findings demonstrate that LSTM-based models provide a practical and effective approach for operational monthly rainfall forecasting in data-scarce semi-arid regions. While hybrid residual correction can offer small improvements, its benefits depend largely on the amount of systematic error remaining after the primary forecasting stage. Future studies should investigate multivariate forecasting frameworks incorporating atmospheric and oceanic predictors such as temperature, humidity, soil moisture, the El Nino -Southern Oscillation (ENSO) and the Indian Ocean Dipole (IOD). Adding such predictors may improve the prediction of extreme rainfall events and further enhance forecasting accuracy for climate adaptation and water resource management in Kenya's arid and semi-arid regions.

ORCID

0009-0006-5251-8188 (Rony Mutugi Muriithi)

Abbreviations

ACF	Autocorrelation Function
ADF	Augmented Dickey–Fuller
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
AR	Autoregression
ARIMA	Autoregressive Integrated Moving Average
ASALs	Arid and Semi-Arid Lands
CHIRPS	Climate Hazards Group InfraRed Precipitation with Station data
CNN	Convolutional Neural Network
DL	Deep Learning
DM	Diebold–Mariano
ENSO	El Niño–Southern Oscillation
FAO GAUL	Food and Agriculture Organization Global Administrative Unit Layers
IOD	Indian Ocean Dipole
IQR	Interquartile Range
LSTM	Long Short-Term Memory
MA	Moving Average
MAE	Mean Absolute Error
ML	Machine Learning
MLR	Multivariate Linear Regression
MSE	Mean Squared Error
NSE	Nash–Sutcliffe Efficiency
PACF	Partial Autocorrelation Function
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SARIMA	Seasonal Autoregressive Integrated Moving Average
SHAP	SHapley Additive exPlanations
WNN	Wavelet Neural Network
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boostings

Author Contributions

Rony Mutugi Muriithi: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Visualization, Writing – original draft

Peter Kinyua Gachoki: Supervision, Validation, Writing – review & editing

Mutua Kilai: Methodology, Supervision, Validation, Writing – review & editing

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] P. I. Palmer, C. M. Wainwright, B. Dong, R. I. Maidment, K. G. Wheeler, N. Gedney, and A. G. Turner. Drivers and impacts of Eastern African rainfall variability. *Nature Reviews Earth & Environment*, 4: 254-270, 2023. https://doi.org/10.1038/s43017-023-00397-x

[2] P. K. Langat, L. Kumar, and R. Koech. Temporal variability and trends of rainfall and streamflow in Tana River Basin, Kenya. *Sustainability*, 9: 1963, 2017. https://doi.org/10.3390/su9111963

[3] T. Dinku, C. Funk, P. Peterson, R. Maidment, T. Tadesse, H. Gadain, and P. Ceccato. Validation of the CHIRPS satellite rainfall estimates over eastern Africa. *Quarterly Journal of the Royal Meteorological Society*, 144(S1): 292-312, 2018. https://doi.org/10.1002/qj.3244

[4] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*, 15(8): 255, 2023. https://doi.org/10.3390/fi15080255

[5] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785-794, 2016. https://doi.org/10.1145/2939672.2939785

[6] M. S. Islam, M. Shafiuzzaman, G. Mahmud, N. Nowshin, P. Reza, J. Hasan, F. M. Ahamed, M. Nahiduzzaman, M. A. Ayari, and A. Khandakar. Explainable deep learning for rainfall prediction: A CNN-XGBoost hybrid approach in the northern region of Bangladesh. *Neural Computing and Applications*, 37: 28125-28160, 2025. https://doi.org/10.1007/s00521-025-11646-z

[7] A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. D. Delgado, and L. A. Akanbi. Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, 7: 100204, 2022. https://doi.org/10.1016/j.mlwa.2021.100204

[8] C. M. Liyew and H. A. Melese. Machine learning techniques to predict daily rainfall amount. *Journal of Big Data*, 8: 153, 2021. https://doi.org/10.1186/s40537-021-00545-4

[9] Z. Xiang, J. Yan, and I. Demir. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resources Research*, 56: e2019WR025326, 2020. https://doi.org/10.1029/2019WR025326

[10] M. Waqas and U. W. Humphries. A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX*, 13: 102946, 2024. https://doi.org/10.1016/j.mex.2024.102946

[11] L. Semmelmann, S. Henni, and C. Weinhardt. Load forecasting for energy communities: A novel LSTM-XGBoost hybrid model based on smart meter data. *Energy Informatics*, 5(Suppl 1): 24, 2022. https://doi.org/10.1186/s42162-022-00212-9

[12] Tripti Dimri, Shamshad Ahmad, and Mohammad Sharif. Time series analysis of climate variables using seasonal arima approach. *Journal of Earth System Science*, 129: 149, 2020. https://doi.org/10.1007/s12040-020-01408-x

[13] Mostafa Dastorani, Mohammad Mirzavand, Dastorani Mohammad Taghi, and Seyyed Javad Sadatinejad. Comparative study among different time series models applied to monthly rainfall forecasting in semi-arid climate condition. *Natural Hazards*, 81: 1811-1827, 2016. https://doi.org/10.1007/s11069-016-2163-x

[14] O. A. Wani, S. S. Mahdi, M. Yeasin, S. S. Kumar, A. S. Gagnon, F. Danish, N. Al-Ansari, S. El-Hendawy, and M. A. Mattar. Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas. *Scientific Reports*, 14, 2024. https://doi.org/10.1038/s41598-024-77687-x

[15] S. Kundu, S. K. Biswas, D. Tripathi, R. Karmakar, S. Majumdar, and S. Mandal. A review on rainfall forecasting using ensemble learning techniques. *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, 6: 100296, 2023. https://doi.org/10.1016/j.prime.2023.100296

[16] M. Niazkar, A. Menapace, B. Brentan, R. Piraei, D. Jimenez, D. Pranav, and M. Righetti. Applications of XGBoost in water resources engineering: A systematic literature review. *Journal of Hydroinformatics*, 26(1), 2024. https://doi.org/10.1016/j.envsoft.2024.105971

[17] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553): 436-444, 2015. https://doi.org/10.1038/nature14539

[18] Y. O. Ouma, R. Cheruyot, and A. N. Wachera. Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: case study of Nzoia River Basin, Lake Victoria, Kenya. *Environmental Science and Pollution Research*, 2021. https://doi.org/10.1007/s40747-021-00365-2

[19] S. Siami-Namini, N. Tavakoli, and A. S. Namin. A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394-1401, 2018. https://doi.org/10.1109/ICMLA.2018.00227

[20] L. J. Slater, L. Arnal, M.-A. Boucher, A. Y.-Y. Chang, S. Moulds, C. Murphy, G. Nearing, G. Shalev, C. Shen, L. Speight, G. Villarini, R. L. Wilby, A. Wood, and M. Zappa. Hybrid forecasting: Blending climate predictions with machine learning. *Hydrology and Earth System Sciences*, 27(9): 1865-1889, 2023. https://doi.org/10.5194/hess-27-1865-2023

[21] J. Saikia, K. Goswami, and S. C. Kakaty. Wavelet-sarima-transformer: A hybrid model for rainfall forecasting. *arXiv*, 2025. https://doi.org/10.48550/arXiv.2509.11903

[22] M. El Hafyani, K. El Himdi, and S.-E. El Adlouni. Improving monthly precipitation prediction accuracy using machine learning models: A multi-view stacking learning technique. *Frontiers in Water*, 6: 1378598, 2024. https://doi.org/10.3389/frwa.2024.1378598

[23] S. D. Latif, D. O. Mohammed, and A. Jaafar. Developing an innovative machine learning model for rainfall prediction in a semi-arid region. *Journal of Hydroinformatics*, 26(4): 904-914, 2024. https://doi.org/10.2166/hydro.2024.014

[24] D. N. Moriasi, M. W. Gitau, N. Pai, and P. Daggupati. Hydrologic and water quality models: Performance measures and evaluation criteria. *Transactions of the ASABE*, 58(6): 1763-1785, 2015. https://doi.org/10.13031/trans.58.10715