

Article

An Architecture-Feature-Enhanced Decision Framework for Deep Learning-Based Prediction of Extreme and Imbalanced Precipitation

Wenjiu Yu, Yingna Sun *, Zhicheng Yue, Zhinan Li and Yujia Liu

Institute of Water Conservancy and Electric Power, Heilongjiang University, Harbin 150080, China; jiujiu121829@gmail.com (W.Y.)

* Correspondence: 2006090@hlju.edu.cn

Abstract

Accurate precipitation forecasting is paramount for water security and disaster mitigation, yet it remains formidable due to atmospheric stochasticity and the inherent class imbalance in rainfall datasets. This study proposes an integrated “architecture-feature-augmentation” framework to circumvent these limitations. Through a systematic evaluation of CNN-LSTM and Transformer architectures, we delineate distinct performance profiles: The Transformer model, when coupled with feature engineering and physics-informed augmentation, yields a peak F1-score of 0.1429, marking the optimal configuration for harmonizing precision and recall. Conversely, CNN-LSTM demonstrates superior robustness in extreme event detection, consistently maintaining high recall rates (up to 0.90) across diverse scenarios. We identify feature engineering as a critical performance modulator, substantially bolstering CNN-LSTM’s baseline metrics while enabling the Transformer to realize its maximum predictive capacity. Although synthetic oversampling techniques—such as SMOTE and GAN—effectively extend the detection range for heavy precipitation, physics-informed augmentation provides the most consistent performance gains, particularly in multi-class contexts. We conclude that the Transformer, augmented by physical constraints, is the optimal candidate for high-precision requirements, whereas CNN-LSTM, integrated with synthetic augmentation, offers a more sensitive alternative for early warning systems prioritizing recall. These findings provide empirical guidance for advancing extreme weather preparedness and strategic water resource management.

Keywords: deep learning; class imbalance; data augmentation methods; feature engineering; time-series forecasting



Academic Editor: Paul Kucera

Received: 29 November 2025

Revised: 1 January 2026

Accepted: 5 January 2026

Published: 8 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Accurate precipitation prediction is critical for water resource management, agricultural planning, and early warning disaster systems, an importance further elevated by the increasing frequency of extreme events under climate change [1–3]. However, achieving high-accuracy prediction remains a formidable challenge due to the complex, nonlinear, and non-stationary nature of precipitation mechanisms [4–6].

The methodology for precipitation prediction has evolved from statistical models to machine learning (ML) and recently to deep learning techniques [7]. Early statistical approaches like ARIMA established a foundation for linear time-series forecasting but struggled with nonlinear dynamics and the integration of multi-source meteorological

variables [8,9]. Subsequent ML methods, including Support Vector Machines (SVMs) and the Random Forests (RF), offered powerful nonlinear fitting capabilities [10–13]. For instance, Tang et al. (2022) demonstrated that data augmentation methods like SMOTE could significantly boost the performance of shallow models like XGBoost [14]. Nonetheless, these models rely heavily on manual feature engineering, and their inherent structures (e.g., tree-based architectures) are limited in capturing raw spatiotemporal dependencies [15].

Deep learning has since brought transformative progress. Architectures such as CNNs enabled automatic spatial feature extraction, while the Convolutional LSTM (ConvLSTM) integrated CNNs with LSTMs, establishing a milestone for spatiotemporal sequence forecasting [16,17]. More recently, the Transformer architecture has emerged as a prominent focus due to its powerful global dependency modeling capabilities, leading to advanced frameworks like PrecipNet, SwinNowcast, and TransMambaCNN for various tasks, including downscaling, nowcasting, and extreme event prediction [18,19].

Despite these advances, deep learning—being inherently data-intensive—faces significant challenges under extreme class imbalance, where critical heavy rainfall events may constitute only a small fraction of samples. In such scenarios, models exhibit a pronounced bias toward the majority class, failing to predict rare extremes. While data augmentation is a potential solution, its application in hydrology remains exploratory and often lacks physical consistency constraints [20]. Moreover, augmentation techniques like SMOTE yield markedly lower performance gains in deep learning models (e.g., LSTM) compared to their substantial benefits in shallow ML, a limitation likely more pronounced in parameter-rich architectures like the Transformer [21]. Consequently, effectively addressing class imbalance under data-scarce and physically constrained conditions remains a core bottleneck.

In summary, a critical research gap exists at the intersection of deep learning and daily-scale extreme precipitation prediction, constrained by limited samples and extreme class imbalance. Existing work primarily focuses on architectural refinements, with insufficient systematic investigation into the synergistic effects between data augmentation strategies and deep learning architectures [22]. Developing augmentation frameworks that incorporate physical constraints to ensure data realism is particularly crucial.

To address these challenges, this study constructs a comprehensive synergistic decision-making framework integrating “Architecture-Feature-Augmentation” for the Yongcui River Basin. Our key contributions are as follows:

1. Development of a spatiotemporal-integrated meteorological feature set, incorporating lag processing, rolling window statistics, and physically coupled features, refined via the Random Forest for enhanced interpretability.
2. Systematic evaluation of architecture–augmentation synergy by comprehensively assessing multiple strategies under extreme imbalance conditions.
3. Design of physics-constrained data augmentation by enforcing meteorological evolution laws to ensure physical plausibility.
4. Establishment of a multi-dimensional evaluation system, including recall and the F1-score, to thoroughly assess minority class identification and stability.

2. Materials

2.1. Study Area

This study is conducted in the Yongcui River Basin, a typical high-latitude forest ecosystem located in the Lesser Khingan Mountains of Heilongjiang Province, China (Figure 1). The basin drains an area of approximately 703 km² and is characterized by high vegetation coverage with minimal human disturbance.

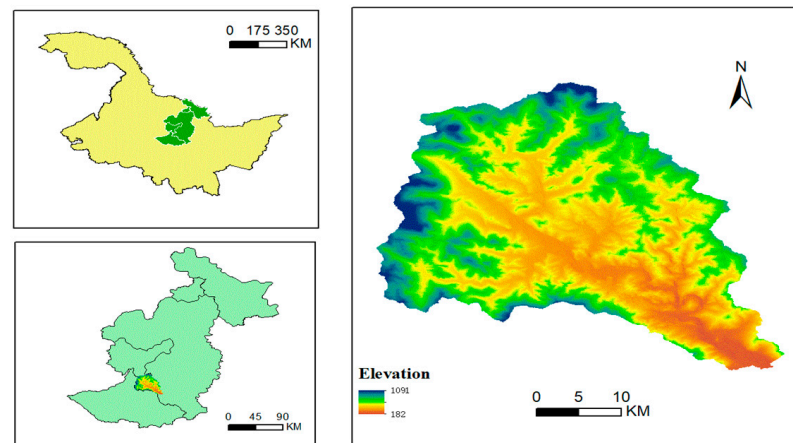


Figure 1. Location map of the study area.

The region experiences a cold-temperate continental monsoon climate, with an average annual precipitation of 650–800 mm that is highly unevenly distributed (Figure 2) [23]. Over 70% of the annual precipitation occurs from June to September, with heavy rainfall events predominantly concentrated in July and August. This temporal concentration, combined with the complex terrain, results in a precipitation dataset with marked seasonal imbalance and a zero-inflated long-tail distribution. These inherent data characteristics make the basin an exemplary case for evaluating deep learning models under extreme class imbalance and sample scarcity.

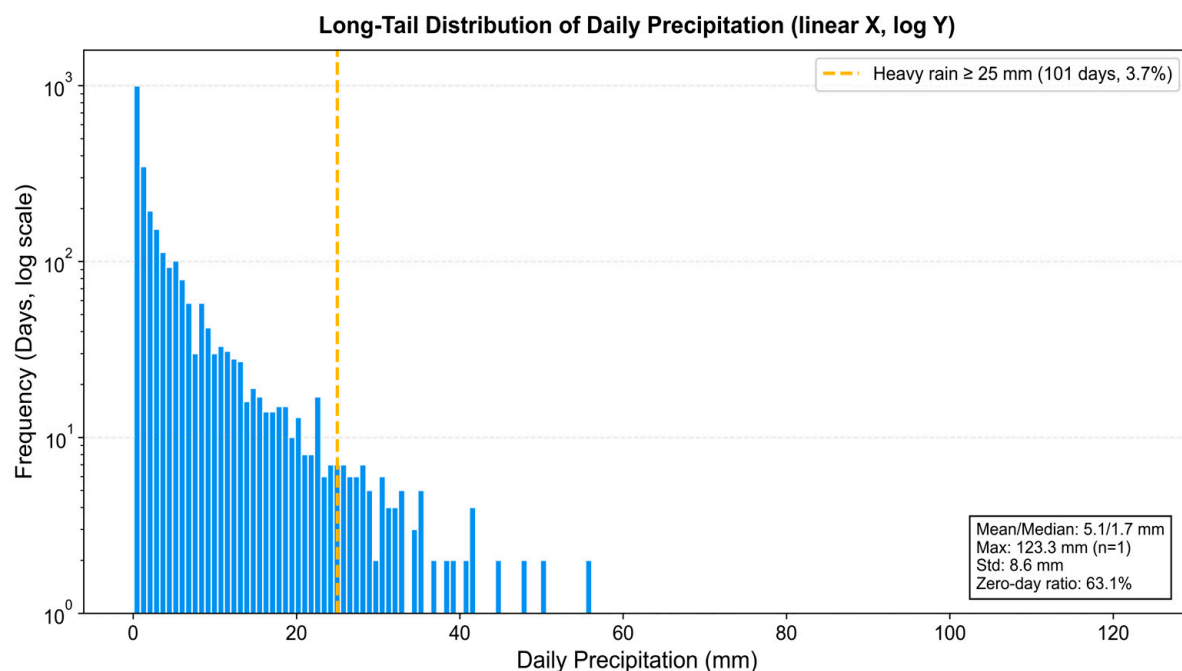


Figure 2. Long-tail distribution of daily precipitation with heavy rainfall events (≥ 25 mm) highlighted.

2.2. Study Data

This study utilized daily meteorological and precipitation data from 1980 to 1999 for the Yongcui River Basin. Meteorological variables, including temperature (mean, max, and min), pressure, relative humidity, wind speed, and sunshine duration, were sourced from the Yichun National Reference Climatological Station. Daily precipitation data were obtained from the Heilongjiang Hydrological Yearbook. The high-quality 20-year series (7300 samples) underwent linear interpolation to handle minimal missing values.

The precipitation data exhibit a pronounced “zero-inflated” and right-skewed distribution (Figure 2). Events were classified into three categories: no rain (0 mm), general precipitation (0.1–24.9 mm), and heavy rainfall (≥ 25 mm). A 25 mm threshold was adopted instead of the common 50 mm standard to better identify hydrologically significant events in this region (Figure 3).

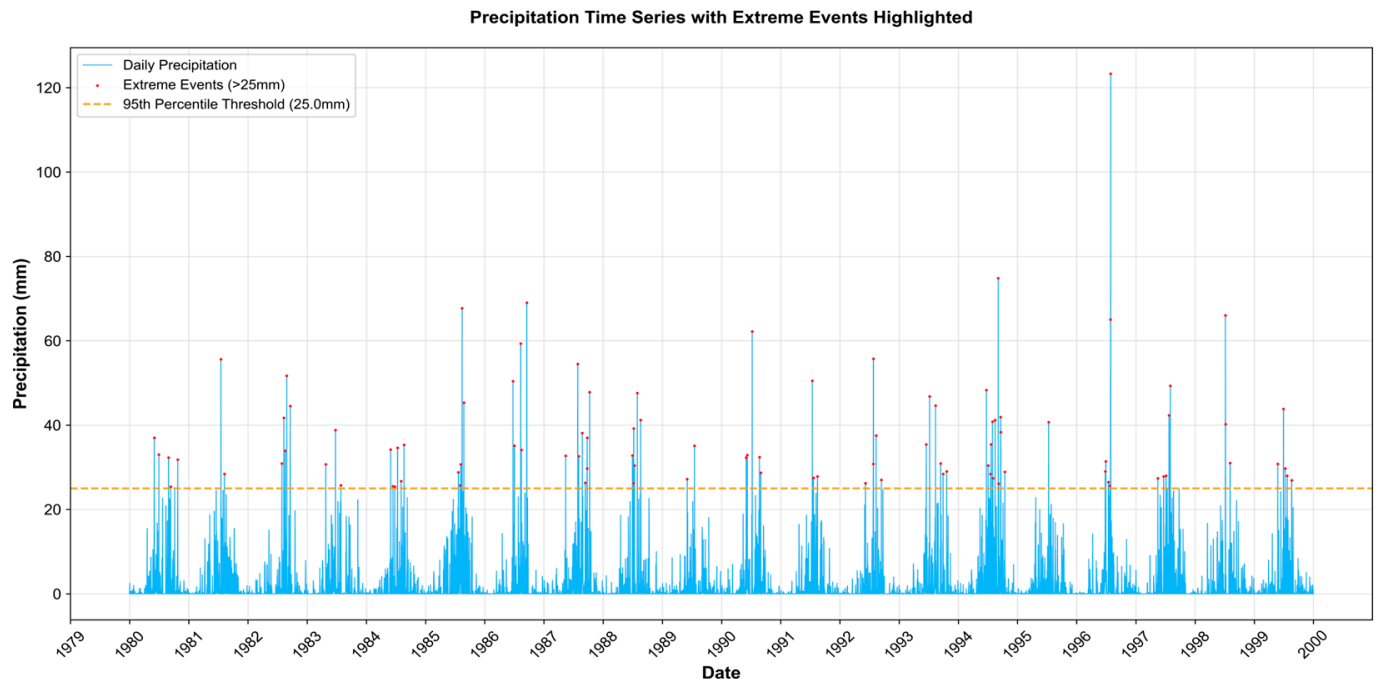


Figure 3. Daily precipitation time series with extreme events highlighted.

The 25 mm threshold for defining heavy precipitation events was determined through a balanced consideration of national standards and local climatic characteristics. While the national meteorological guideline in China classifies heavy rainfall as ≥ 50 mm/24 h, regional precipitation patterns exhibit significant spatial heterogeneity. Our study area demonstrates a lower precipitation intensity distribution, with the 95th percentile of daily rainfall measuring 11.92 mm. The selected 25 mm threshold represents an optimal compromise that maintains alignment with national standards while ensuring sensitivity to local extreme events. This approach balances statistical representativeness (being approximately twice the 95th percentile) with operational practicality for regional forecasting applications. The threshold ensures adequate event capture while minimizing false alarms in this specific climatic context.

A systematic feature engineering pipeline was implemented, encompassing raw data preprocessing, temporal feature construction, and the design of physically coupled features, followed by feature selection to enhance model performance.

Critically, the dataset exhibits severe class imbalance (Figure 4) [24]. The training set only contains 81 heavy rainfall samples (vs. 3610 no-rain and 2063 general precipitation), and the test set merely contains 20 samples (vs. 934 and 507). This scarcity of heavy rainfall events substantially impedes model training, leading to low F1-scores and posing a core challenge for accurate identification of these critical events [25].

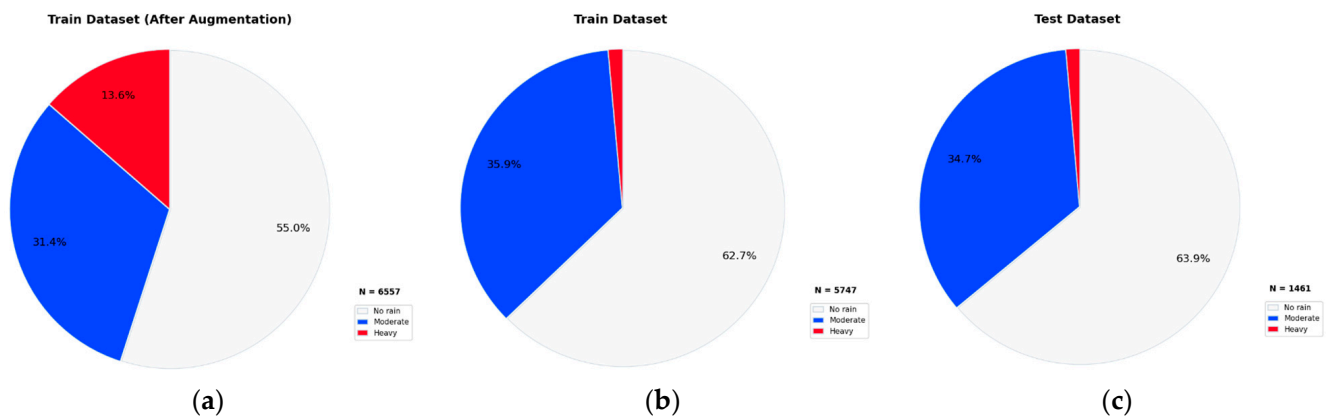


Figure 4. Class distribution comparison of precipitation datasets: (a) training set (before augmentation), (b) training set (after augmentation), and (c) test set.

3. Methods

3.1. Feature Selection

To mitigate the risks of dimensionality inflation and overfitting caused by redundant features, this study established a three-tier feature selection pipeline [26].

1. Standardization: The feature matrix $X \in \mathbb{R}^{n \times d}$ was standardized using Z-score normalization, which is calculated as follows:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (1)$$

In this equation, μ_j and σ_j represent the mean and standard deviation of the j -th feature.

2. Redundant Feature Elimination: Features exhibiting a high linear correlation ($|r| > 0.8$) were removed, and low-variance features ($\sigma^2 < 0.01$) were filtered out. This step aimed to reduce inter-feature redundancy and enhance feature independence.
3. Feature Selection: Based on the feature importance ranking derived from the Random Forest algorithm (see Figure 5), the top 30 most discriminative features were selected. This method evaluates the average contribution of a feature during node splits in the tree models, thereby measuring its impact on the classification outcome [27,28].

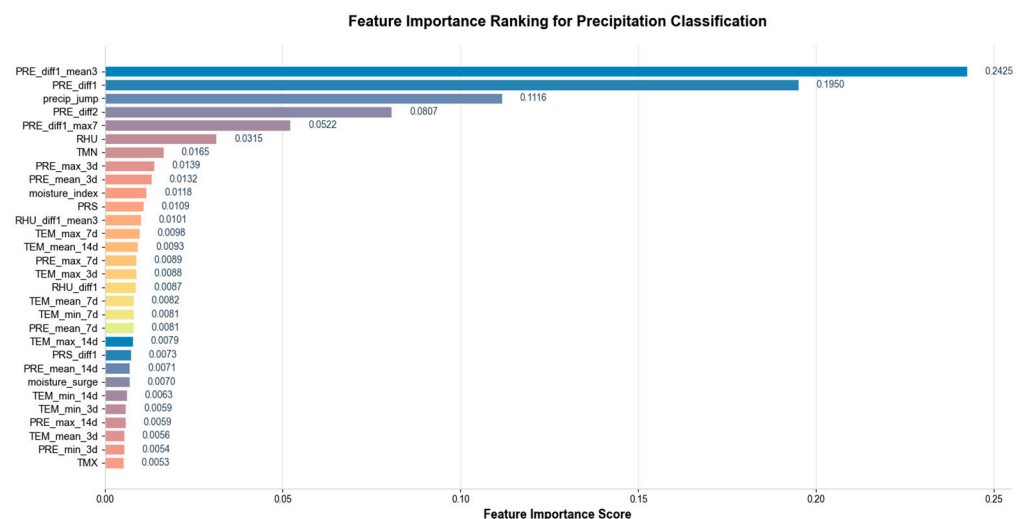


Figure 5. Feature importance ranking for three-class heavy rainfall prediction.

$$F_{\text{selected}} = \underset{F' \subseteq F, |F'|=30}{\operatorname{argmax}} \sum_{f \in F'} I(f) \quad (2)$$

The importance score for a feature f , denoted as $I(f)$, quantifies its contribution to the model's decision-making process, typically using Gini importance or the mean decrease in impurity. This pipeline effectively improved the utilization efficiency of the feature space, reduced the adverse impact of redundant information on model performance, and simultaneously enhanced the model's interpretability. Figure 5 presents the feature importance rankings derived from the binary and ternary classification models.

The final selection of the top 30 features by importance ranking achieved a cumulative contribution rate of 85.3%. Precipitation variation features (e.g., PRE_diff1_mean3, PRE_diff1, etc.) were dominant, demonstrating consistency with the abruptness characteristic of heavy rainfall events.

Our feature importance analysis revealed that temporal precipitation patterns—specifically short-term precipitation changes (PRE_diff1), rolling mean anomalies (PRE_diff1_mean3), and abrupt precipitation jumps (precip_jump)—are the most discriminative predictors for extreme rainfall events. To fully leverage these temporal dynamics, our Transformer architecture incorporates dedicated temporal convolution modules prior to the self-attention layers, explicitly capturing local multi-scale precipitation trends and abrupt transitions. The encoder's 4 attention heads then learn to attend to both short-term meteorological shocks and longer-term antecedent moisture conditions (e.g., RHU and PRE_max_3d), while the learnable positional encoding adapts to the specific temporal scale of convective processes. The model is trained with AdamW ($\text{lr} = 1 \times 10^{-4}$), a 0.3 dropout rate for robust generalization, and a focal loss that emphasizes the heavy-rain class. This integrated design—where feature-informed preprocessing complements the Transformer's global attention—ensures that the performance gains stem from a principled architectural synergy tailored to precipitation physics, rather than opaque training dynamics.

3.2. Sample Augmentation Strategy

The extreme class imbalance, with heavy rainfall samples constituting only 1.4% of the dataset, biases deep learning models toward the majority class, severely limiting their ability to identify critical precipitation events. To mitigate this, we implemented and evaluated four distinct augmentation strategies, which are categorized into interpolation-based and generative approaches, to systematically explore the synergy between data augmentation and model architecture.

(1) Interpolation-based Oversampling: SMOTE-TS and ADASYN.

We employed two variants of the Synthetic Minority Over-sampling Technique (SMOTE) designed for time-series data. The SMOTE-TS algorithm generates synthetic samples via linear interpolation between temporally adjacent minority class instances [29,30].

$$x_{\text{new}} = x_i + \lambda(x_{i+1} - x_i), \lambda \sim U(0, 1) \quad (3)$$

Building on this, the Adaptive Synthetic Sampling (ADASYN) algorithm incorporates an adaptive mechanism to preferentially generate samples near the feature space boundaries of difficult-to-learn minority instances [31]. While both methods effectively increase the minority sample size, they do not explicitly enforce physical consistency among the generated meteorological variables, potentially introducing implausible data points.

$$x_{\text{new}} = x_i + \lambda(x_z - x_i), x_z \in \text{KNN}(x_i), \lambda \sim U(0, 1) \quad (4)$$

(2) Physics-Constrained Perturbation Augmentation.

To address the physical inconsistency limitation of interpolation methods, we developed a novel physics-constrained perturbation strategy [32,33]. This method applies controlled random perturbations ($\pm 5\%$) to the temperature and relative humidity of existing heavy rainfall samples. Crucially, it enforces a monotonic pressure decrease constraint ($PRS(t+1) < PRS(t)$) to maintain physical consistency with atmospheric instability mechanisms preceding heavy rainfall. This approach enhances model robustness while ensuring the augmented data remains physically plausible.

(3) Generative Adversarial Network (TimeGAN and Light-TimeGAN).

As a generative alternative; we implemented a Time-Series Generative Adversarial Network (TimeGAN) and its lightweight variant (Light-TimeGAN) to capture underlying temporal distributions [34]. The Light-TimeGAN architecture employs single-layer LSTM networks for both the generator and the discriminator, which are trained on all available heavy rainfall samples ($N = 81$) using the Adam optimizer. To ensure meteorological plausibility, we integrated verification mechanisms including variable range validation, metadata inheritance, and statistical feature constraints via a feature matching loss. These mechanisms enhance the physical realism and stability of the generated sequences.

All augmentation strategies were implemented through a unified Python3.9 framework. The heavy rainfall augmentation factor was controlled by the parameter `heavy_rain_aug_factor` (set to 10) to ensure consistent and comparable experimental conditions across all models.

3.3. Data Partitioning Strategy

To prevent data leakage and preserve temporal coherence, we partitioned the 20-year dataset (1980–1999) using an annual stratified time-series split. This approach ensures that the model is trained on past data and evaluated on future data, maintaining a realistic forecasting scenario and providing a robust assessment of generalization capability [35,36]. A three-fold validation scheme was employed, with each fold containing non-overlapping blocks of years for training and validation, thus eliminating cross-year boundary leakage. Table 1 depicts the specific temporal distribution of the three-fold splits.

Table 1. Temporal splitting scheme for three-fold cross-validation.

Fold	Training Year Range	Validation Year Range	Training Data Coverage
Fold 1	1980–1989	1990–1992	Approx 65%
Fold 2	1980–1991	1992–1994	Approx 70%
Fold 3	1980–1993	1994–1995	Approx 75%

3.4. Evaluation Metrics

A multi-tiered evaluation framework was adopted to rigorously assess model performance, with a primary focus on the detection of rare heavy rainfall events. The selected metrics are as follows:

Overall Performance: Accuracy and the Area Under the ROC Curve (AUC-ROC) were used to evaluate general discriminative capability.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d\text{FPR} \quad (6)$$

Heavy Rainfall Detection: Recall and the F1-score for the heavy rainfall class served as the primary metrics to quantify the model's sensitivity and precision in identifying critical events.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

$$Heavy - Recall = \frac{TP_{heavy}}{TP_{heavy} + FN_{heavy}} \quad (8)$$

Imbalance Adjustment: Balanced accuracy was incorporated to mitigate interpretive biases caused by class imbalance.

$$Balanced\ accuracy = \frac{1}{2} \times \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (9)$$

This framework ensures a comprehensive assessment aligned with the operational priority of maximizing extreme weather event detection.

$$Brier\ Score = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2 \quad (10)$$

3.5. Threshold Decision Mechanism

To enhance the stability and generalization of the classification threshold, we implemented a three-fold averaging methodology. The final decision threshold (τ) was set as the arithmetic mean of the three optimal thresholds, each identified as the point maximizing the F1-score for the heavy rainfall category within a validation fold:

$$\tau = \frac{1}{3} \sum_{k=1}^3 \tau_k \quad (11)$$

This strategy mitigates threshold fluctuations caused by data partitioning randomness, thereby enhancing predictive robustness.

3.6. Monsoon Feature Enhancer

The month variable ($m \in [1, 12]$) was processed through an Enhanced Monsoon Embedder to capture seasonal climate patterns. To avoid imposing ordinal bias, we first applied a discrete sinusoidal embedding. This was subsequently transformed by a two-layer nonlinear network into an optimized 64-dimensional representation:

$$e_{month} = f_{embed}(m) = \text{Linear}_2(\text{ReLU}(\text{Linear}_1(m))) \quad (12)$$

The resulting temporal embedding e_{month} was then concatenated with the CNN-encoded meteorological features, followed by unified normalization. This design enables seasonally adaptive feature weighting, enhancing the model's capacity to capture regional precipitation regimes through dynamically modulated representations.

3.7. Input Windowing Strategy

To capture medium-term and long-term dependencies in precipitation patterns, this study constructs input samples using a sliding time window approach. Each sample consists of consecutive meteorological observations from the preceding 90 days, with the objective of predicting the precipitation category of the target day (day 91). The 90-day window length was empirically determined to be optimal through the performance of the validation set, effectively balancing the coverage of characteristic seasonal variation cycles against potential noise introduction and computational burden associated with longer

sequences. The 90-day window was selected based on meteorological considerations of seasonal precipitation patterns. This duration corresponds to a full season (approximately 3 months), allowing the model to capture complete seasonal transitions and antecedent conditions that influence precipitation extremes. The window size balances sufficient temporal context for pattern recognition while maintaining computational efficiency.

3.8. Bootstrap Resampling Methodology

It is important to acknowledge that the preliminary results for our baseline models—specifically the CNN-LSTM and Transformer configurations without optimization—exhibit certain performance limitations, particularly regarding predictive precision. However, these initial tests were primarily conducted as a proof of concept to validate the statistical stability and structural integrity of our evaluation framework. Through a rigorous Bootstrap resampling analysis ($N = 1000$), we observed that while the numerical metrics remain at a foundational level, the consistency of the results is noteworthy. For instance, both baselines yielded a stable 95% confidence interval (CI) width of 0.0419 for the heavy rainfall F1-score, with the CNN-LSTM and Transformer maintaining highly consistent mean accuracies of 0.3223 and 0.2899, respectively. These results, despite their current modest scale, successfully demonstrate that our experimental pipeline is statistically robust and free from extreme stochastic fluctuations. This stability provides a reliable, albeit preliminary, foundation upon which more sophisticated feature engineering and augmentation strategies can be systematically evaluated and improved.

4. Model Architecture

4.1. 1D CNN

Convolutional Neural Networks (CNNs) form a foundational deep learning architecture. While their two-dimensional (2D CNN) and three-dimensional (3D CNN) variants are commonly employed in image and video recognition, one-dimensional convolutional networks (1D CNNs) are particularly suited for time-series analysis. This study utilizes a 1D CNN to model local temporal characteristics within meteorological sequences, with a focus on capturing short-term fluctuation patterns in variables such as temperature, humidity, and atmospheric pressure preceding heavy rainfall events. The input data is structured as sequential tensors with dimensions (B, T, and F), representing batch size, time steps, and feature dimension, respectively [7]. Initially, input features are projected into a 64-dimensional latent space through linear transformation to standardize feature scales and enhance representation capacity:

$$h_t = \text{ReLU}(X \cdot W + B) \quad (13)$$

The temporal embedding vector is then concatenated with the transformed features to create an enhanced feature representation, where $z_t = [h_t; e_{\text{month}}]$. This composite feature is subsequently processed through convolutional layers (kernel size = 3 and stride = 1), batch normalization, and ReLU activation, followed by max-pooling operations (window size = 2) to extract abstract representations of local temporal patterns. These processing steps systematically compress the temporal dimension from T to T/2, effectively smoothing high-frequency noise while amplifying subtle pressure fluctuation signals that typically precede rainfall events. The resulting refined features provide optimized inputs for subsequent temporal dependency modeling stages. Figure 6 illustrates the architecture of a one-dimensional Convolutional Neural Network (1D CNN).

$$H_{\text{cnn}} = \text{MaPool}(\text{ReLU}(\text{BN}(\text{Conv1D}(z_t))))$$

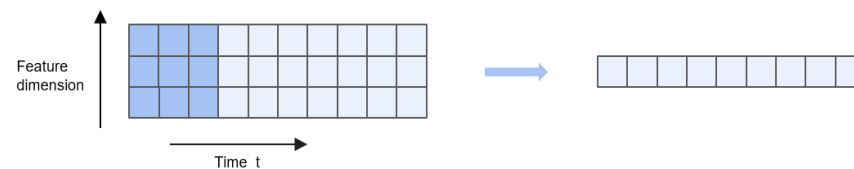


Figure 6. A one-dimensional CNN with a single convolution kernel with a size of 3×3 .

4.2. BiLSTM

Long Short-Term Memory (LSTM) networks effectively capture dynamic temporal dependencies through specialized gating mechanisms, overcoming the vanishing and exploding gradient problems associated with traditional recurrent neural networks. However, conventional LSTM networks process sequences solely in the forward temporal direction and are unable to leverage potentially informative reverse dependencies.

Bidirectional LSTM (BiLSTM) architectures address this limitation by incorporating both forward and backward processing units, enabling simultaneous modeling of temporal dependencies in both directions. The implemented network employs hidden layers with 128 units, substantially enhancing its capacity for comprehensive temporal feature representation. The update process can be formally expressed as

$$\vec{h}_t = LSTM_f(H_{cnn}[t]) \quad \overleftarrow{h}_t = LSTM_b(H_{cnn}[t]) \quad (14)$$

The final output is obtained by concatenating the hidden state vectors from both directions:

$$\hat{h}_t = \begin{bmatrix} \vec{h}_t; \overleftarrow{h}_t \end{bmatrix} \quad (15)$$

This architecture facilitates bidirectional feature propagation across the entire temporal sequence, which significantly enhances the model's capacity to represent the dynamic evolution patterns characteristic of heavy rainfall events. Through BiLSTM's dual-directional modeling mechanism, the framework concurrently captures both gradual seasonal precipitation trends and abrupt heavy rainfall signals, thereby establishing a robust foundation for comprehensive temporal dependency modeling within the prediction system.

4.3. One-Dimensional CNN-BiLSTM

This study develops a CNN-BiLSTM hybrid model that synergistically combines 1D-CNNs for local pattern extraction with bidirectional LSTM networks for capturing long-range, bidirectional temporal dependencies. The model processes input sequences of dimensions (N, T, and D) as follows:

First, raw meteorological features are enhanced by concatenation with 16-dimensional seasonal embeddings generated from an Enhanced Monsoon Embedder. The composite sequence is then fed into a 1D CNN module to extract salient short-term fluctuation patterns and compress the temporal dimension. The resulting local features are subsequently processed by a two-layer bidirectional LSTM network (256 hidden units per layer) to model the long-range evolution of precipitation systems. Finally, the output from the last temporal step is passed through fully connected layers to generate probability distributions over the precipitation categories. To mitigate class imbalance, the model is trained using focal loss, with optimization details provided in Section 4.5. The model architecture diagram is shown in Figure 7.

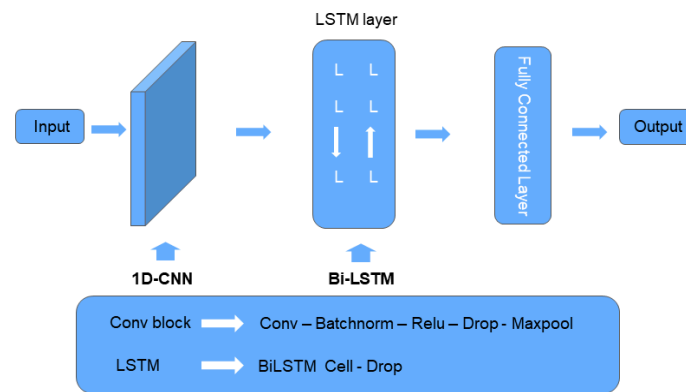


Figure 7. Multi-level precipitation forecast network.

4.4. Improved Transformer

While the standard Transformer excels in global dependency modeling via multi-head self-attention, its application to non-stationary meteorological sequences often results in over-smoothed representations and diminished sensitivity to critical local variations [37]. To address this, we introduce an improved precipitation Transformer.

Our architecture adapts the standard Transformer for precipitation classification through three key modifications: (1) the decoder is removed as the task requires classification, not generation; (2) a seasonal embedding module is incorporated to enhance temporal awareness of monthly variations; and (3) an attention-based learnable pooling mechanism replaces standard pooling, using trainable query vectors to generate a globally contextualized representation via weighted feature aggregation. These enhancements collectively enable the model to more effectively capture the complex, evolving patterns in meteorological data. Figure 8 presents the model architecture diagram.

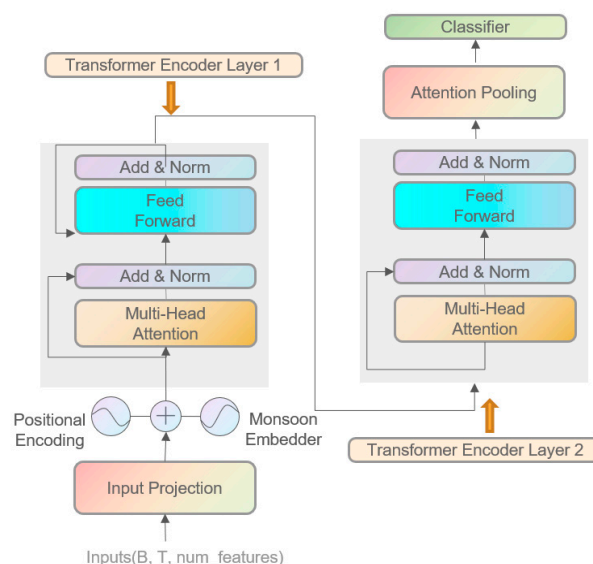


Figure 8. Schematic diagram of the enhanced Transformer architecture.

We developed a specialized Transformer architecture optimized for extreme precipitation prediction, featuring significant modifications from standard designs. Key improvements include a seasonal embedding module for meteorological pattern recognition, attention-based temporal pooling for dynamic feature weighting, and a compact two-layer structure with 128 hidden dimensions to prevent overfitting. The architecture replaces standard components with domain-specific elements: GELU activations for smoother gradient

flow, learnable positional encodings adapted to weather sequences, and enhanced dropout regularization (0.3) for extreme event prediction robustness (Table 2).

Table 2. Configuration of hyperparameters.

Hyperparameter Category	Parameter Name	Value
Model Architecture	Hidden Dimension	128
	Number of Layers	2
	Attention Heads	4
	Window Size	90
Training Configuration	Dropout Rate	0.3
	Batch Size	64
	Learning Rate	1.00×10^{-4}
Task-Specific Setting	Focal Loss Gamma	2
	Number of Classes	3

4.5. Training Configuration

To address the extreme class imbalance, we employed focal loss ($\alpha = 0.25$, $\gamma = 2$) as the training objective, which dynamically scales the cross-entropy loss to focus learning on hard, minority-class examples. All models were optimized using Adam with a consistent learning rate of 1×10^{-3} , 1×10^{-4} , 5×10^{-4} . Training incorporated early stopping to prevent overfitting. Detailed hyperparameters (e.g., scheduler and batch size) are provided in Supplementary Material. This unified setup ensures fair comparability across all architectural and augmentation experiments.

5. Results and Discussion

5.1. The Impact of Data Augmentation Methods on Model Performance

5.1.1. Comparison of Data Augmentation Under the Three-Class Classification Task

Figure 9 tells an interesting story: just how well data augmentation works depends a great deal on the model architecture you are using. Take the CNN-LSTM—here, physics-perturbed augmentation consistently comes out on top. The heavy rainfall F1-score climbs from 0.1130 at baseline to 0.1349, all while recall holds steady at a respectable 0.8500. Furthermore, the confidence interval tightens to 0.0335, a good sign that predictions are becoming more stable. It seems that by respecting the natural continuity of weather data, physics-aware perturbations help the CNN-LSTM model handle extreme class imbalance better. We compare that with purely mathematical fixes like SMOTE, which barely move the needle for this architecture. Apparently, when it comes to catching rare heavy rainfall events, SMOTE does not offer much help.

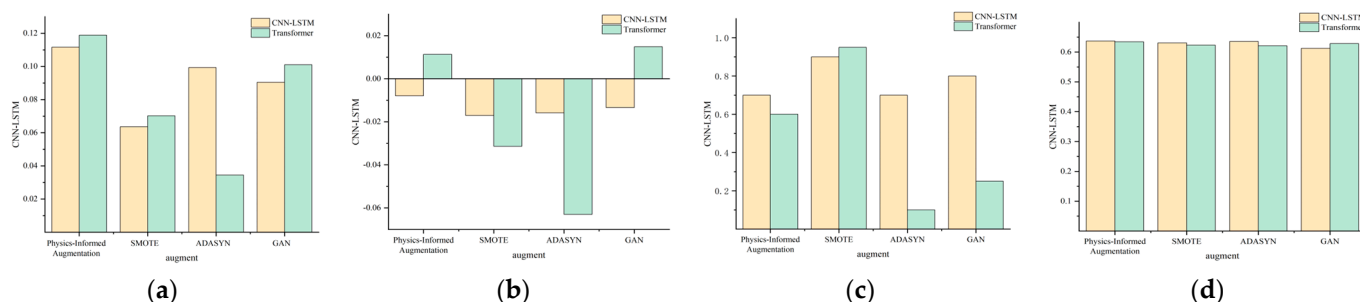


Figure 9. Comparative results of the three-class forecasting task. Figure (a) displays the F1-score on the test set, Figure (b) illustrates the generalization gap between the validation set and the test set, Figure (c) presents the AUC score on the test set, and Figure (d) shows the recall on the test set.

After switching to the Transformer, the picture changes entirely. We pair it with ADASYN, and recall for heavy rainfall jumps to 0.9500—the model suddenly detects far more minority-class events. That is the self-attention mechanism for you: it really leans into enriched minority samples. But there is a trade-off. Precision takes a hit, and decision boundaries become less stable across the three classes. So even though ADASYN pushes recall higher, the overall F1-score does not improve much. And SMOTE? It hardly makes a dent here either; recall stays flat at 0.1000. Simple linear interpolations just do not seem to give Transformers what they need.

Stepping back, a pattern emerges. With aggressive augmentation—think ADASYN- or GAN-generated samples—the Transformer can stretch its sensitivity to minority classes further. CNN-LSTMs, on the other hand, prefer a gentler, physics-informed touch that preserves temporal flow. In practice, that means that if you are chasing maximum recall—say, for early warning systems—a Transformer with adaptive augmentation might be worth the trade-offs. But if you care about steady, reliable performance, you will probably want the CNN-LSTM model paired with physics-aware perturbations. It is less about which is universally better and more about what each architecture brings to the problem in front of you.

5.1.2. Comparison of Data Augmentation Under the Binary Classification Task

When we shift to a simpler “heavy rain or not” setup, the choice of model starts to matter much more—and so does how you augment your data. Without any augmentation, the CNN-LSTM already performs a decent job, catching about two-thirds of heavy rainfall events with a fairly low threshold. It is a reminder that convolutional and recurrent layers can still pull useful local patterns from thin data. The baseline Transformer, by comparison, is surprisingly cautious, recalling only 30% of events despite its reputation for modeling global relationships.

Once we start adding augmented data, the two architectures clearly go their separate ways. For the CNN-LSTM, only ADASYN makes a real difference, boosting recall to 0.70 and lifting the AUC to 0.7534. However, for GAN-generated samples, performance actually drops. It is as if the recurrent setup becomes distracted when synthetic sequences do not “feel” physically realistic.

The Transformer, on the other hand, seems far more adaptable. Both ADASYN and SMOTE push its recall up to 0.90—the self-attention mechanism apparently thrives on enriched minority examples. But what is really interesting is how it handles physics-informed perturbations (Exp. 34). That configuration achieves the best balanced accuracy (0.8376), almost as if the physical constraints help steer the model’s attention toward what meteorologically matters. Even GAN-based augmentation (Exp. 40) works here, delivering the highest F1-score (0.1366) while keeping false alarms in check.

In this binary setting, the Transformer, when paired with the right augmentation, simply mines extreme events more effectively. If you cannot afford to miss a heavy rainfall signal, a Transformer fed with ADASYN or SMOTE samples might be your best bet. But if you value consistency and stability across different conditions, you will likely lean on GAN-based or physics-guided augmentation—approaches that help the model generalize, not just memorize.

5.2. Performance Comparison Between Model Architectures and Task Types

5.2.1. Comparison of Model Architectures Under the Three-Class Classification Task

Figure 9 reveals a clear trade-off between catching every possible event and trusting the predictions you obtain. For the unaugmented models, the Transformer pulls ahead in heavy rainfall recall (0.75 vs. CNN-LSTM’s 0.15)—it is simply better at spotting the rare

cases. But that sensitivity comes at a cost. Its predictions are less certain, with a calibration error (ECE) nearly double that of CNN-LSTM.

Stability, though, is where CNN-LSTM quietly shines. Its recall might be lower, but it manages a slightly better F1-score and AUC right out of the gate. It is not chasing every outlier; it is aiming for a balanced result. And with a little help from something like SMOTE, its recall improves without throwing that balance off—the model assimilates the new samples without losing its way.

The Transformer's relationship with augmentation is more complicated. Sure, ADASYN tightens up its calibration error, but at a steep price: precision plummets, and its decision boundaries become shaky. It seems that flooding a self-attention model with oversampled data in a multi-class setting can do more harm than good.

If our goal is to miss as few heavy rainfall events as possible, the Transformer is your tool—just be ready for its predictions to come with more uncertainty. If you need a reliable, well-calibrated forecast you can count on, CNN-LSTM, when gently augmented, is probably the steadier choice. It is less about which model is better and more about what kind of performance you are willing to trade for.

5.2.2. Comparison of Model Architectures Under the Binary Classification Task

Figure 10 makes one thing clear: when you switch to a straight “yes or no” forecast, the right model really depends on what you feed it. With no data augmentation, CNN-LSTM takes the lead—its AUC (0.7532) and recall (0.65) both outpace those of the Transformer. It is a reminder that for spotting local patterns in thin data, convolutional and recurrent layers still have an edge.

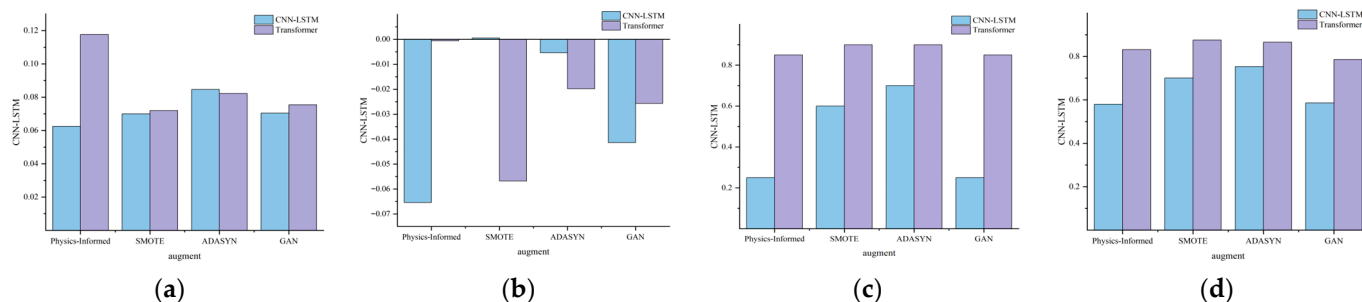


Figure 10. Comparative results of the two-class forecasting task. Figure (a) displays the F1-score on the test set, Figure (b) illustrates the generalization gap between the validation set and the test set, Figure (c) presents the AUC score on the test set, and Figure (d) shows the recall on the test set.

But give both models some augmented data, and their paths split. The Transformer practically thrives on it. Whether through physics-informed tweaks or synthetic samples, its recall jumps to between 0.85 and 0.90, and AUC climbs past 0.86. It is as if the self-attention mechanism just needs enough examples to latch onto—once it has them, it finds the signal.

The CNN-LSTM is not quite as flexible. Push it with aggressive augmentation and, while recall might inch up, overall discrimination suffers. There is even a case where physics-based augmentation seems to backfire, causing recall to drop sharply. It looks like perturbing the data can sometimes confuse the model's more localized decision process.

From a practical standpoint, this shows up in the thresholds you end up using. The CNN-LSTM keeps things steady, its optimal threshold hovering around 0.6. The Transformer, though, often needs to be dialed up to 0.8 or 0.9 to keep false alarms in check—a deliberate swap of sensitivity for precision.

If we need a model that works reliably straight out of the box, the CNN-LSTM holds its ground. But if you are able to carefully prepare your data—adding the right augmented samples—the Transformer can achieve a powerful blend of high recall and

strong discrimination. It is not that one is universally better; it is that the Transformer, given the right support, can be tuned into a remarkably precise tool for spotting extreme rain.

5.3. Comparison of Models with and Without Feature Enhancement

5.3.1. Contribution Evaluation of Feature Engineering Under Three-Class Classification Task

A comprehensive evaluation of the CNN-LSTM and Transformer architectures reveals that advancements in precipitation forecasting are not merely a function of model depth but are profoundly driven by the synergistic coupling of feature engineering and data augmentation. Experimental evidence underscores this: in the non-augmented baseline (Exp. 11), the Transformer yielded a meager heavy rainfall recall of 0.2286, accompanied by significant fluctuations in overall accuracy. However, the integration of feature enhancement catalyzed a paradigm shift from “passive fitting” to “physics-aware perception.” In Exp. 12, even without data augmentation, the recall for heavy rainfall surged to 0.9000—a magnitude of improvement particularly pronounced in the Transformer architecture. By introducing derived variables with clear physical significance, feature engineering enables the self-attention mechanism to precisely anchor precursory signals of extreme weather, effectively addressing the “missed detection” bottleneck for rare events (comprising only 1.41% of the dataset).

Simultaneously, data augmentation and feature engineering exert a powerful complementary effect. Exp. 14 represents the pinnacle of this synergy; under the combined influence of physical perturbations and feature enhancement, the heavy rainfall F1-score reached a high of 0.1429, with recall stabilizing at 0.7000. This marks a fundamental improvement in classification balance compared to architecture-only optimization. Compared to CNN-LSTM, the Transformer, when bolstered by feature engineering, demonstrates superior global feature extraction and statistical stability. For instance, in Exp. 20 (feature engineering + GAN), the model achieved a 0.5000 recall while maintaining a Coefficient of Variation (CV) of only 4.16% and a 95% confidence interval as narrow as 0.0445, categorizing as “highly stable.” The core value of this synergy lies in optimizing the decision boundary, thereby allowing the optimal threshold to be sensitively calibrated between 0.73 and 0.92. Ultimately, while feature engineering overcomes information quality constraints through domain knowledge, data augmentation bridges the gap created by sample scarcity. Together, they form a performance growth pole that far exceeds the marginal architectural gains of transitioning from CNN-LSTM to the Transformer.

5.3.2. Contribution Evaluation of Feature Engineering Under the Binary Classification Task

In the context of binary precipitation forecasting, the efficacy of feature engineering (FE) demonstrates a profound conditionality that is intricately contingent upon the synergistic alignment between model architectures and data augmentation (DA) strategies.

For the CNN-LSTM architecture, FE functions as a “performance stabilizer.” Its impact varies distinctly across different augmentation environments: in scenarios without DA (Exp. 21), while FE marginally moderated the test set’s recall from 0.6500 to 0.6000, it bolstered the AUC from 0.6364 to 0.6760, thereby refining the model’s overall discriminative power. When coupled with GAN-based augmentation (Exp. 29), FE demonstrated exceptional robustness, propelling the heavy rain recall from 0.2500 (without FE, Exp. 30) to 0.6500 while significantly elevating the balanced accuracy from 0.5844 to 0.7591—effectively compensating for the inherent limitations of CNNs in capturing sparse samples.

In stark contrast, the Transformer architecture exhibits a “high-ceiling, high-sensitivity” profile regarding FE. In the absence of any DA, FE (Exp. 31) proved to be transformative, surging the heavy rain recall from 0.3000 (Exp. 32) to 0.8500 and “activating” the AUC from

0.6156 to 0.8489. This underscores the potent physical resonance between the self-attention mechanism and derived feature variables.

However, this sensitivity introduces stability challenges. Following the introduction of oversampling techniques like ADASYN or SMOTE, the Transformer displayed an almost “hyper-responsive” behavior. Although Exps. 36 and 38 (without FE) achieved an elite recall of 0.9000, they suffered from severe predictive distribution imbalance (overestimating heavy rain by over 20 times), causing precision to collapse to approximately 0.04. The reintegration of FE (Exp. 35, 37) acted as a critical corrective, recalibrating the optimal threshold toward 0.92 and suppressing false alarms. Conversely, CNN-LSTM maintained a `cls_f1_norain` range between 0.93 and 0.99 when paired with physical perturbations (Exp. 23) or SMOTE (Exp. 27), exhibiting superior statistical resilience.

The synthesis of these 20 experimental groups reveals that FE serves as a steady-progress tool for CNN-LSTM, optimizing decision boundaries via physical constraints to maintain balanced accuracy above 0.70 despite a 1.41% minority class ratio. For Transformers, FE acts as an explosive gain switch, enabling near-complete detection (recall > 0.85) but remaining susceptible to noise-induced bias. These findings illuminate that FE, DA, and architecture form a tightly coupled system rather than a simple additive relationship. For scenarios prioritizing maximum warning ceilings, the Transformer with FE is the optimal choice; for tasks requiring operational stability, the FE-driven CNN-LSTM approach offers superior practical equilibrium.

5.4. Comprehensive Evaluation: Synergistic Effects and the Optimal Configuration Strategy of Data Augmentation, Feature Engineering, and Model Architecture

Based on the comprehensive experimental data, the core finding of this study is corroborated and refined: peak performance in extreme precipitation prediction is not the product of a singularly superior component, but it emerges from the precise, task-aware alignment of model architecture, feature engineering, and the data augmentation strategy (Tables 3 and 4).

Table 3. Experimental combinations ranked by the test heavy-rain f1-score (three-class).

Experimental Combination	Test_f1_Heavy	ΔF1
Transformer + Feature Engineering + Physics-Informed Augmentation	0.1429	0.0229
CNN-LSTM + Feature Engineering + Physics-Informed Augmentation	0.1351	0.0298
CNN-LSTM + Feature Engineering + SMOTE	0.1296	0.0701
Transformer + No Feature Engineering + No Augmentation	0.1235	0.0128

Table 4. Experimental combinations ranked by the test heavy-rain f1 score (two-class).

Experimental Combination	Test_f1_Heavy	ΔF1
Transformer + No Feature Engineering + GAN	0.1366	0.0381
CNN-LSTM + Feature Engineering + ADASYN	0.1351	0.0493
CNN-LSTM + No Feature Engineering + No Augmentation	0.1296	0.0029
Transformer + No Feature Engineering + Physics-Informed Augmentation	0.1235	−0.0006

The CNN-LSTM and Transformer architectures exhibited fundamentally different interaction patterns with the other components. The CNN-LSTM model demonstrated strong synergistic effects. Feature engineering consistently served as a performance foundation, and its combination with certain augmentations yielded significant gains. For

instance, pairing feature engineering with the ADASYN (Exp. 25, $F1 = 0.1250$) or GAN (Exp. 29, $F1 = 0.1166$) in binary classification tasks produced the top scores for that architecture, enhancing model capability effectively. In contrast, the Transformer's inherent self-attention mechanism showed a more nuanced and sometimes adversarial relationship with feature engineering. While a strong baseline was established with "feature engineering + no augmentation" (Exp. 31, $F1 = 0.1145$), introducing feature engineering alongside augmentations often disrupted this balance. Notably, the optimal configuration for the Transformer involved forgoing feature engineering and applying GAN augmentation (Exp. 40, $F1 = 0.1366$), suggesting that its ability to learn representations directly from raw sequential data is a key strength that can be amplified by the right generative augmentation.

Generalization analysis, measured by the performance gap ($\Delta F1$) between validation and test sets, reveals critical insights for deployment reliability. Configurations employing oversampling methods (ADASYN and SMOTE) were particularly prone to large negative $\Delta F1$ values, indicating significant overfitting where high validation scores masked poor test performance (e.g., Exp. 36, $\Delta F1 = -0.0568$). These configurations, while capable of high test scores in some cases, carry substantial deployment risk. Conversely, combinations involving physics-informed augmentation or no augmentation tended to show smaller $\Delta F1$ magnitudes, signifying more stable and trustworthy models (e.g., Exp. 34, $\Delta F1 = -0.0006$).

Therefore, the optimal model configuration is highly task-specific and objective-driven. We recommend the following:

- For scenarios demanding the highest F1 score with balanced recall and stability, employ the Transformer with feature engineering and physics-informed augmentation in a three-class formulation (Exp. 14, $F1 = 0.1429$ and $\Delta F1 = +0.0229$). This configuration achieved the overall best performance with robust generalization.
- For scenarios prioritizing high recall and model stability in a simpler binary task, the Transformer without feature engineering but with physics-informed augmentation is a robust choice (Exp. 34, $F1 = 0.1176$, recall = 0.85, and $\Delta F1 \approx 0.00$).
- For leveraging CNN-LSTM with an acceptable risk profile, CNN-LSTM with feature engineering and ADASYN/GAN can achieve competitive binary classification performance (Exps. 25 and 29, $F1 \approx 0.12$), but its generalization behavior ($\Delta F1$) should be closely monitored to mitigate overfitting potential.

6. Conclusions

Based on a comprehensive evaluation of 40 controlled experiments for daily-scale, highly imbalanced precipitation forecasting, this study establishes that optimal model performance is not an intrinsic property of any single component but emerges from the deliberate, task-specific synergy among model architecture, feature engineering, and data augmentation. The principal findings and recommendations, grounded in empirical results, are as follows:

Architectural Synergy, Not Superiority: No single architecture was universally superior. Performance was dictated by compatibility. The Transformer achieved the highest absolute F1-score (0.1429) in the three-class task when paired with feature engineering and physics-informed augmentation, leveraging its representational power on enriched data. In contrast, CNN-LSTM excelled in providing robust and generalizable performance in binary classification, with its optimal configuration (no feature engineering + no augmentation) achieving a high F1-score (0.1204) and stable generalization ($\Delta F1 \approx 0$).

Feature Engineering as a Conditional Enhancer: Its efficacy was highly context-dependent. For CNN-LSTM, it served as a consistent foundation, with its most significant gains realized when combined with adaptive augmentations like ADASYN. For the Transformer, however, feature engineering was not always beneficial; its highest F1-score in

binary classification was achieved without engineered features, suggesting that its self-attention mechanism can be impaired by poorly aligned feature spaces.

Augmentation Defines Generalization Risk: The choice of data augmentation strategy was the primary determinant of overfitting. Oversampling methods (SMOTE and ADASYN) frequently led to large negative $\Delta F1$ values, indicating high validation scores that failed to generalize to the test set, thus posing a high deployment risk. Physics-informed augmentation proved to be the most reliable, consistently improving or maintaining performance while demonstrating superior generalization stability (low $|\Delta F1|$). GAN-based augmentation showed potential, producing the top-performing binary classification model, but its stability varied.

Based on the experimental evidence, we propose the following configuration guide:

1. For maximizing predictive performance (F1-score), employ the Transformer with feature engineering and physics-informed augmentation in a three-class formulation.
2. For high-stability and high-recall warnings, use the Transformer without feature engineering but with physics-informed augmentation in a binary task, which offers an excellent recall (0.85) and stable generalization.
3. For a robust and interpretable baseline, CNN-LSTM without feature engineering and without augmentation provides strong, reliable performance in binary classification, serving as a trustworthy benchmark.

In summary, this research provides a principled, synergistic decision-making framework that moves beyond seeking a universal "best model" and instead guides the construction of task-optimal forecasting systems for extreme hydrometeorological events.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/w18020176/s1>, Table S1: Experimental summary analysis table.

Author Contributions: Conceptualization, W.Y.; methodology, W.Y.; software, W.Y.; validation, Y.S. and Z.Y.; formal analysis, W.Y.; investigation, W.Y. and Y.L.; resources, Y.S.; data curation, W.Y.; writing—original draft preparation, W.Y.; writing—review and editing, Y.S. and Z.Y.; visualization, W.Y. and Z.L.; supervision, Y.S.; project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. (The raw precipitation data are not publicly available due to confidentiality agreements with the data provider, but relevant verification data can be found in the Heilongjiang Provincial Hydrological Yearbook (<https://tjj.hlj.gov.cn/tjjnianjian/2024/zk/indexeh.htm>, accessed on 28 November 2025); the meteorological data are publicly available).

Acknowledgments: The authors sincerely thank everyone for their valuable suggestions and assistance throughout this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kundzewicz, Z.W. Climate change impacts on the hydrological cycle. *Ecolhydrol. Hydrobiol.* **2008**, *8*, 195–203. [[CrossRef](#)]
2. Schmitt, R.W. The Ocean Component of the Global Water Cycle. *Rev. Geophys.* **1995**, *33*, 1395–1409. [[CrossRef](#)]
3. Pei, Y.; Liu, J.; Wang, J.; Mei, C.; Dong, L.; Wang, H. Effects of urbanization on extreme precipitation based on weather research and forecasting model: A case study of heavy rainfall in Beijing. *J. Hydrol. Reg. Stud.* **2024**, *56*, 102078. [[CrossRef](#)]
4. Hou, A.Y.; Kakar, R.K.; Neeck, S.; Azarbarzin, A.A.; Kummerow, C.D.; Kojima, M.; Oki, R.; Nakamura, K.; Iguchi, T. The Global Precipitation Measurement Mission. *Bull. Am. Meteorol. Soc.* **2014**, *95*, 701–722. [[CrossRef](#)]
5. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
6. Sun, A.Y.; Scanlon, B.R.; Save, H.; Rateb, A. Reconstruction of GRACE Total Water Storage Through Automated Machine Learning. *Water Resour. Res.* **2021**, *57*, e2020WR028669. [[CrossRef](#)]

7. Li, W.; Gao, X.; Hao, Z.; Sun, R. Using deep learning for precipitation forecasting based on spatio-temporal information: A case study. *Clim. Dyn.* **2022**, *58*, 443–457. [\[CrossRef\]](#)
8. Wang, H.R.; Wang, C.; Lin, X.; Kang, J. An improved ARIMA model for precipitation simulations. *Nonlinear Process. Geophys.* **2014**, *21*, 1159–1168. [\[CrossRef\]](#)
9. Lai, Y.; Dzombak, D.A. Use of Integrated Global Climate Model Simulations and Statistical Time Series Forecasting to Project Regional Temperature and Precipitation. *J. Appl. Meteorol. Climatol.* **2021**, *60*, 695–710. [\[CrossRef\]](#)
10. Das, S.; Chakraborty, R.; Maitra, A. A random forest algorithm for nowcasting of intense precipitation events. *Adv. Space Res.* **2017**, *60*, 1271–1282. [\[CrossRef\]](#)
11. Zhu, S.; Wei, J.; Zhang, H.; Xu, Y.; Qin, H. Spatiotemporal deep learning rainfall-runoff. forecasting combined with remote sensing precipitation products in large scale basins. *J. Hydrol.* **2023**, *616*, 128727. [\[CrossRef\]](#)
12. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
13. Tripathi, S.; Srinivas, V.V.; Nanjundiah, R.S. Downscaling of precipitation for climate change scenarios: A support vector machine approach. *J. Hydrol.* **2006**, *330*, 621–640. [\[CrossRef\]](#)
14. Tang, T.; Jiao, D.; Chen, T.; Gui, G. Medium-and long-term precipitation forecasting method based on data augmentation and machine learning algorithms. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1000–1011. [\[CrossRef\]](#)
15. Hartigan, J.; MacNamara, S.; Leslie, L.M. Application of machine learning to attribution and prediction of seasonal precipitation and temperature trends in Canberra, Australia. *Climate* **2020**, *8*, 76. [\[CrossRef\]](#)
16. Rasp, S.; Pritchard, M.S.; Gentile, P. Deep learning to represent subgrid processes in climate models. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 9684–9689. [\[CrossRef\]](#)
17. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 802–810.
18. Adibfar, A.; Davani, H. PrecipNet: A transformer-based downscaling framework for improved precipitation prediction in San Diego County. *J. Hydrol. Reg. Stud.* **2025**, *62*, 102738. [\[CrossRef\]](#)
19. Zhang, K.; Zhang, G.; Wang, X. TransMambaCNN: A Spatiotemporal Transformer Network Fusing State-Space Models and CNNs for Short-Term Precipitation Forecasting. *Remote Sens.* **2025**, *17*, 3200. [\[CrossRef\]](#)
20. Reddy, T.; Bhattacharya, S.; Maddikunta, P.K.R.; Hakak, S.; Khan, W.Z.; Bashir, A.K.; Jolfaei, A.; Tariq, U. Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset. *Multimed. Tools Appl.* **2022**, *81*, 41429–41453.
21. Guo, H.; Sun, S.; Zhang, X.; Chen, H.; Li, H. Monthly precipitation prediction based on the EMD–VMD–LSTM coupled model. *Water Supply* **2023**, *23*, 4742–4758. [\[CrossRef\]](#)
22. Sit, M.; Demiray, B.Z.; Demir, I. A systematic review of deep learning applications in streamflow data augmentation and forecasting. *arXiv* **2022**. [\[CrossRef\]](#)
23. Sui, Y.; Jiang, D.; Tian, Z. Latest update of the climatology and changes in the seasonal distribution of precipitation over China. *Theor. Appl. Climatol.* **2013**, *113*, 599–610. [\[CrossRef\]](#)
24. Lee, C.E.; Kim, S.U. Applicability of Zero-Inflated Models to Fit the Torrential Rainfall Data. *Water* **2017**, *9*, 123. [\[CrossRef\]](#)
25. You, X.; Liang, Z.; Wang, Y.; Zhang, H. A study on loss function against data imbalance in deep learning correction of precipitation forecasts. *Atmos. Res.* **2023**, *281*, 106500. [\[CrossRef\]](#)
26. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
27. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
28. Rodriguez-Galiano, V.F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P. An assessment of the effectiveness of a random forest classifier for land-cover classification. *Remote Sens. Environ.* **2012**, *123*, 37–50. [\[CrossRef\]](#)
29. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
30. Lv, Y.; Zhang, L.; Fan, W.; Zhang, Y. Precipitation Retrieval from FY-3G/MWRI-RM Based on SMOTE-LGBM. *Atmosphere* **2024**, *15*, 1268. [\[CrossRef\]](#)
31. Han, H.; Liu, W. Improving Rainfall Prediction Using Adaptive Synthetic Sampling and LSTM Network. *Atmosphere* **2023**, *14*, 932.
32. Karpatne, A.; Watkins, W.; Read, J.; Kumar, V. Physics-guided neural networks (PGNN): Incorporating scientific knowledge into deep learning models. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2351–2365.
33. Shen, W.; Chen, S.; Xu, J.; Zhang, Y.; Liang, X.; Zhang, Y. Enhancing Extreme Precipitation Forecasts through Machine Learning Quality Control of Precipitable Water Data from FengYun-2E. *Remote Sens.* **2024**, *16*, 3104. [\[CrossRef\]](#)
34. Wang, Y.; Zhai, H.; Cao, X.; Geng, X. A Novel Accident Duration Prediction Method Based on a Conditional Table Generative Adversarial Network and Transformer. *Sustainability* **2024**, *16*, 6821. [\[CrossRef\]](#)
35. Wang, G.; Feng, Y.; Dai, Y.; Chen, Z.; Wu, Y. Optimization Design of a Windshield for a Container Ship Based on Support Vector Regression Surrogate Model. *Ocean Eng.* **2024**, *313*, 119405. [\[CrossRef\]](#)

36. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed.; OTexts: Melbourne, Australia, 2018.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.