RESEARCH ARTICLE

Event-Triggered CNN-LSTM with Prediction-Error Closed-Loop Feedback for Rainfall Forecasting

Xiaoqi Yin[1,*], **Jing Zhou**[1], **Ruping Liu**[1], **Chenxi Ma**[1] **and Chen Li**[1]

[1] School of Electronic Information Engineering, Huai'an University, Huai'an 223003, China

Abstract

Intermittent daily rainfall requires temporal pattern extraction and correction of newly observed prediction errors. This paper presents an event-triggered convolutional neural network–long short-term memory model with prediction-error closed-loop feedback (ECF-CNN-LSTM) for seasonal one-day-ahead rainfall forecasting. A 30-day window supplies 31 causal rainfall-history and calendar features. A binary gate controls a zero-initialized, bounded residual and learns from the expected prediction errors of its inactive and active alternatives. A separate chronological controller uses each newly observed error to adjust subsequent forecasts. The single-series Tengzhou, Shandong dataset contains 976 unique June–September dates over 2013–2020. The proposed model and its ablations use 312 training targets from 2013–2016 and 156 validation targets from 2017–2018, with five random seeds. In the fixed-network feedback comparison, the complete model obtains mean absolute error (MAE) of 1.948576 mm and root mean squared error (RMSE) of 7.370497 mm, reducing the backbone errors by 1.15% and 2.43%, respectively. Feedback reduces MAE in all five paired runs for both predictor variants. The complete model's wet-day and dry/trace-day MAEs are 11.049914 and 0.043645 mm; the MAE standard deviation across seeds is 0.104537 mm. The results quantify complementary learned and sequential forecast corrections within the evaluated seasonal setting. The rainfall-intensity analysis covers 27 wet targets, including one above 50 mm.

Keywords: rainfall forecasting, CNN-LSTM, event trigger, prediction-error feedback, closed-loop correction.

1 Introduction

Rainfall forecasting is important for flood prevention, urban drainage, agricultural scheduling, and water resource management. In practical hydrometeorological scenarios, rainfall time series are usually sparse and non-stationary. Long dry periods are interspersed with short but intense rainfall events, and the target distribution is strongly skewed. These properties make rainfall prediction more difficult than ordinary smooth time series forecasting. Recent daily-rainfall studies have explored rainfall-history reconstruction, recurrent prediction and hybrid networks to represent this variability [1–3].

Traditional statistical forecasting methods usually

Academic Editor:
Abdul Rehman

Submitted: 08 July 2026
Revised: 11 September 2026
Accepted: 16 September 2026
Published: 22 September 2026

Vol. 3, **No.** 3, 2026.
10.62762/TIS.2026.531561

***Corresponding author:**
Xiaoqi Yin
hy_xuebao2009@126.com

Citation
Yin, X., Zhou, J., Liu, R., Ma, C., & Li, C. (2026). Event-Triggered CNN-LSTM with Prediction-Error Closed-Loop Feedback for Rainfall Forecasting. *ICCK Transactions on Intelligent Systematics*, 3(3), 172–185.

rely on linear assumptions or fixed distributional forms. They are efficient and interpretable, but they can have difficulty representing nonlinear rainfall evolution and abrupt changes. Recent rainfall-specific approaches combine wavelet decomposition, machine learning and gated convolution with attention [4, 5]. Convolutional neural networks (CNNs) extract local patterns from ordered inputs, while long short-term memory (LSTM) networks represent sequential dependencies through gated memory states [6, 7]. Hybrid CNN-LSTM models combine these complementary forms of temporal representation.

For intermittent rainfall, a useful question is how to balance stable dry-period predictions with responsiveness to recent changes. Gated residual correction [8], multi-regime rainfall prediction [9] and forecast postprocessing [10, 11] offer relevant approaches. Against this background, we investigate a specific proposition: can a forecast-trained residual branch improve a common CNN-LSTM predictor, and can chronological error feedback further reduce its errors when the network is held fixed? This separates learning a forecast correction from applying an online correction state.

To address this question, this paper proposes an event-triggered CNN-LSTM with prediction-error closed-loop feedback (ECF-CNN-LSTM). The perception–prediction–correction structure combines local CNN feature extraction and recurrent prediction with two explicit correction paths. A forecast-utility gate determines whether to activate a bounded residual. Its training objective compares the rainfall errors of inactive and active correction, connecting the gate to the forecasting task. In parallel, prediction errors revealed by newly available observations update a scalar correction for subsequent targets. This separation makes both the gate trajectory and the error-feedback state inspectable.

The contributions concern the correction architecture and its controlled evaluation under a common CNN-LSTM backbone:

- A forecast-utility-driven binary correction branch combines an explicit on/off decision, a zero-initialized bounded residual, and an expected-error objective that connects gate learning directly to the relative errors of the two forecast alternatives.
- A prediction-error closed-loop feedback channel operates chronologically: the current observation updates only subsequent predictions, and the correction state is retained within each annual evaluation sequence.
- A completed four-arm, five-seed ablation evaluates the correction branch and the sequential controller under a common CNN-LSTM backbone. Feedback is compared at identical neural weights, with complete metrics, rainfall-interval counts and chronological state records supporting interpretation of the measured contributions.

The remainder of this paper reviews related methods in Section 2, presents the framework in Section 3, specifies the experiments in Section 4, and analyzes the results in Section 5. Sections 6 and 7 discuss the practical implications and summarize the contributions.

2 Related Work

2.1 Deep Temporal Models

Rainfall forecasting has been studied using statistical models, machine learning regressors, and deep neural networks. Classical autoregressive and regression-based methods capture temporal dependencies through prescribed functional forms. Hybrid deep learning approaches that couple one-dimensional convolutional layers with LSTM have demonstrated improved multi-step-ahead daily rainfall prediction over individual models [12]; lag and rolling features additionally provide historical context for such predictors.

For rainfall-history prediction, Necesito et al. [1] combined discrete wavelet reconstruction and univariate LSTM modeling for daily rainfall in four Philippine subcatchments. Patro and Bartakke [2] also investigated LSTM for daily rainfall prediction. Küllahçı and Altunkaynak [4] combined maximum-overlap discrete wavelet transforms with machine learning using three Turkish rainfall stations and forecast horizons extending to three days. These studies connect recurrent prediction and signal decomposition directly to the daily-rainfall task.

LSTM [6] and its forget-gate extension [7] provide the recurrent foundation used here. Surveys of deep learning for time-series forecasting document how encoder–decoder architectures represent temporal information in both one-step and multi-horizon settings [13], while direct comparisons of PSO-SVR, LSTM and CNN for short-term rainfall illustrate the complementary strengths of these sequence models

[14]. The present task is a single-series daily forecast: its CNN-LSTM encodes local variation before recurrent prediction, with the encoder structure held common across correction settings.

Recent daily-rainfall studies provide more directly comparable task settings. Samson and Aweda [3] compared single and hybrid networks across African cities using reanalysis and meteorological predictors, finding location-dependent performance. Waqas et al. [15] investigated daily precipitation in Thailand with biorthogonal wavelet-based hybrids. These studies motivate controlled comparison of backbone complexity and complementary temporal representations.

Gating and attention also have rainfall-specific precedents. Wu et al. [5] combined gated convolution and attention with variational mode decomposition for daily rainfall forecasting. Seidu et al. [8] applied a gated residual learning framework to daily rainfall forecasting in coastal Ghana, evaluating residual corrections learned from validation data across multiple machine learning backbones. These approaches enrich the backbone or add a post-prediction correction stage. Our contribution instead makes the inactive and active residual forecasts explicit in the training objective and separates that learned correction from an observation-driven output state.

2.2 Abrupt Changes and Imbalanced Rainfall

Weighted deep learning models for heavy rainfall [16] and multi-step deep learning forecasters for sequential rainfall prediction [17] illustrate complementary directions in deep learning approaches to rainfall forecasting, although their task settings differ from single-series daily forecasting. Customized loss functions and multitask learning objectives have been applied to deep learning models for precipitation bias correction and downscaling, improving the representation of heavy rainfall events [18], whereas a binary gate makes an explicit path-selection decision. Farman et al. [9] studied rainfall prediction across multiple cities and climatic regimes using tree-based and deep learning approaches with multiple lead times and random seeds. Rainfall-amount regression across diverse climatic regimes and forecast-path selection therefore address different aspects of the prediction problem. Here a correction event denotes activation of an additional residual from the preceding rainfall window. Its score is optimized through rainfall prediction error, so the training signal directly concerns correction utility.

Dry-day prevalence also affects the evaluation objective: low aggregate absolute error can coexist with large errors on rare wet days. You et al. [19] studied how Dice loss and weighted cross-entropy mitigate sample imbalance in deep learning correction of heavy precipitation forecasts, showing that loss-function constraints direct training toward rare large-rainfall events. Intensity-weighted losses change the relative cost of wet days, while our binary gate makes an explicit path-selection decision whose training signal directly concerns correction utility. Our analysis makes the rainfall distribution visible through wet/dry errors, interval counts, a zero-rainfall reference and individual high-rainfall cases.

2.3 Error Correction

Rojas-Campos et al. [10] used probabilistic neural networks to postprocess numerical-weather-prediction precipitation forecasts at four observation locations, separating occurrence probability from hourly rainfall amount. Miri et al. [11] combined neighboring-station observations, meteorological delta features, LSTM attention and probabilistic postprocessing in an hourly forecasting setting. These studies establish forecast correction as a relevant rainfall-modeling component while also showing how correction depends on the available information and target timescale.

The present design uses a bounded gated residual and a fixed-step chronological output controller for daily rainfall. The residual's zero initialization preserves the initial backbone forecast, and the feedback update can be checked against observation timing. Unlike attention-weighted continuous representations, the gate selects a discrete forecast-correction path. Unlike a static reweighted loss, the feedback state changes after each newly observed outcome. An always-active residual is the limiting case of the gated path; activation frequencies are therefore reported alongside forecast errors.

3 Proposed Method

The proposed method follows a perception–prediction–correction route. Historical rainfall is encoded into a multi-dimensional temporal feature matrix, CNN-LSTM extracts the predictive representation, a forecast-utility branch supplies a gated residual, and the chronological error controller adjusts subsequent forecasts. Figure 1 shows the overall architecture, and Figure 2 details the event and feedback paths.

The method comprises four steps. **S1** constructs standardized features within each annual season. **S2** extracts local CNN features and a standard LSTM representation. **S3** computes the correction decision and its bounded forecast residual. **S4** issues the daily prediction and updates the feedback state after the observation arrives. The two correction branches act on the forecast output; the standard LSTM memory equations are unchanged.

The backbone contains two one-dimensional convolution (Conv1D) blocks with batch normalization (BN) [20], rectified linear units (ReLU) and max pooling, a two-layer LSTM, and three fully connected (FC) output layers. Dropout provides regularization [21]. The event MLP reads the last pooled CNN position and uses three affine layers. A separate zero-initialized residual head and a scalar output-feedback state complete the architecture.

3.1 Problem Definition and Input Window

Let the original rainfall sequence be

$$Y = \{y_1, y_2, \ldots, y_T\}, \tag{1}$$

where y_t is the observed daily rainfall at time step t. Because rainfall data contain many zero values and a small number of extreme values, the target is transformed by

$$\ell_t = \log(1 + y_t). \tag{2}$$

The inverse transformation $\hat{y}_t = \exp(\widehat{\ell}_t) - 1$ recovers the predicted rainfall in millimeters for evaluation. Observed ℓ_t, unprojected network output q_t and final predicted $\widehat{\ell}_t$ are distinguished throughout.

For each historical day, a feature vector is formed from its observed log rainfall, wet indicator, log-rainfall change, calendar variables, lagged rainfall, rolling statistics and trends:

$$x_t = [x_{t,1}, \ldots, x_{t,F}] \in \mathbb{R}^F, \qquad F = 31, \tag{3}$$

with the feature groups listed in Table 1. Each feature dimension is standardized as

$$z_{t,j} = \frac{x_{t,j} - \mu_j}{s_j}. \tag{4}$$

Given a sliding window length T_w, the input sample for predicting time step t is

$$X_t = [z_{t-T_w}, z_{t-T_w+1}, \ldots, z_{t-1}] \in \mathbb{R}^{T_w \times F}. \tag{5}$$

Here $T_w = 30$. Means and population standard deviations are fitted only to the 432 valid daily feature rows in the 2013–2016 training seasons, before window repetition. Constant-feature scales are set to one. The input uses ℓ_i, $\mathbf{1}\{y_i > 0.1\}$ and $\ell_i - \ell_{i-1}$ in place of the two constant identifiers and absolute year, giving 31 channels. The identifiers are retained only as dataset metadata. Features and windows are constructed separately within each June–September season, with no off-season bridge. A row may include rainfall observed on that historical day, but prediction of t uses rows only through $t-1$. The direct rainfall channels expose information also represented indirectly by the rolling statistics.

3.2 CNN-Based Local Temporal Feature Extraction

The standardized input window is passed to a one-dimensional CNN to extract local temporal patterns. For a convolution kernel of width k, the local feature at position i can be written as

$$v_i = \phi\left(\sum_{m=0}^{k-1} W_m z_{i+m} + b_{\mathrm{c}}\right), \tag{6}$$

where W_m and b_{c} are trainable parameters and $\phi(\cdot)$ is a nonlinear activation function. The implementation uses two Conv1D blocks. Each block contains convolution, batch normalization, ReLU activation, and max pooling. Dropout is applied after the convolutional blocks. The output sequence is denoted as

$$H_t = [v_1, v_2, \ldots, v_M], \tag{7}$$

where $M = 7$ is the reduced length after two pooling operations. For batch size K, the tensor path is $K \times 30 \times 31$ to $K \times 64 \times 15$ and then $K \times 128 \times 7$. Both convolution kernels have width 3, stride 1 and padding 1; pooling has width and stride 2. Dropout is 0.3. This representation captures local increasing, decreasing and stationary patterns within the rainfall window.

3.3 Event-Feedback Gate

The core module is the event-feedback gate shown in Figure 2. It contains an event-triggered path and an error-feedback path.

3.3.1 Forecast-Utility Event Correction

A correction event denotes activation of the residual path for a particular forecast window. The last pooled CNN position, $H_t[:, M]$, is read by an MLP with three affine layers of widths $128 \to 64 \to 32 \to 1$, ReLU

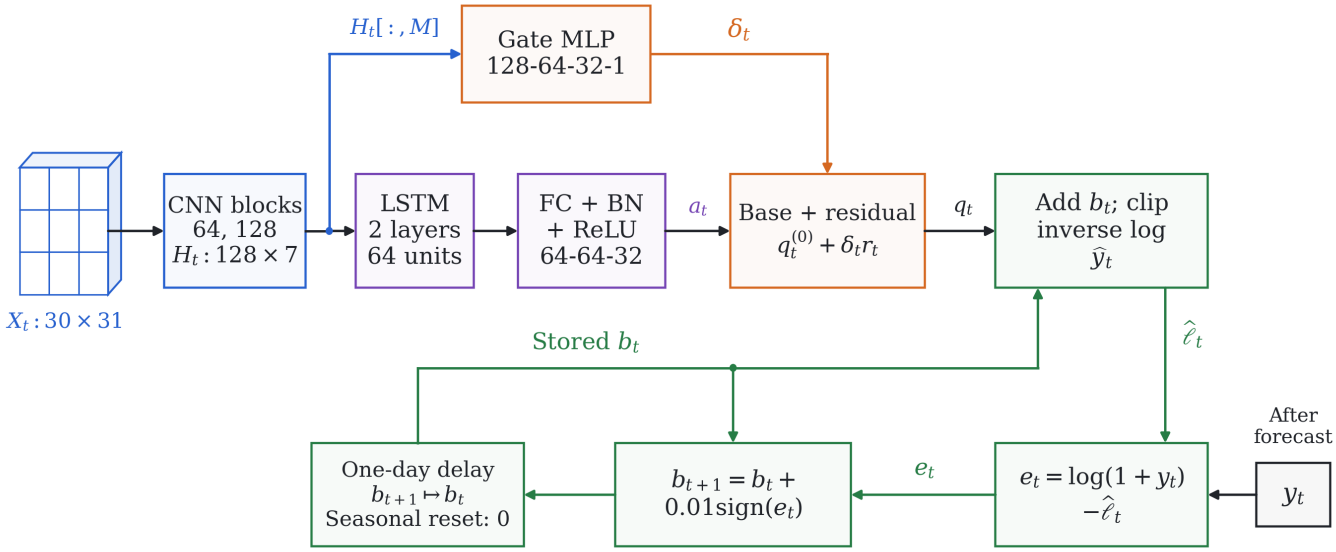

Figure 1. Overall ECF-CNN-LSTM architecture. The CNN output feeds both the LSTM backbone and the event detector. The shared FC representation supplies the two forecast heads detailed in Figure 2. Separate forecast and observation inputs determine the error; the one-day delay maps updated b_{t+1} to the stored bias b_t used at the next forecast step.

Table 1. The 31 daily features. Rainfall inputs in historical row i use observations no later than i; the forecast window for t ends at $t - 1$.

Group	Count	Definition
Direct rainfall	3	Log rainfall ℓ_i, wet indicator $\mathbf{1}\{y_i > 0.1\}$, and within-season log change $\ell_i - \ell_{i-1}$.
Calendar	5	Month, weekday, day of year, season index $\lfloor(\text{month} \bmod 12)/3\rfloor + 1$, and June–September indicator.
Lagged rainfall	7	Log rainfall ℓ_{i-d} for $d = 1, \ldots, 7$.
Rolling statistics	9	Trailing mean, maximum and sum of log rainfall over 7, 14 and 30 days, including i, with at least one available row.
Cyclic calendar	4	Sine and cosine of day of year (period 365) and month (period 12).
Trends	2	Differences of the 7- and 14-day rolling means separated by 7 and 14 days.
Interaction	1	Month multiplied by the season index.

between hidden layers and a sigmoid output:

$$\xi_t = \text{MLP}(H_t[:, M]), \quad p_t = \sigma(\xi_t), \quad \delta_t = \mathbf{1}\{p_t > 0.5\}. \tag{8}$$

The score parameterizes the choice of a forecast correction, rather than a calibrated rainfall-occurrence or flood probability. Training uses the prediction losses of both alternatives, as defined below; it does not differentiate the hard threshold or require a separate historical-event classification label. The deployed gate remains binary.

The cutoff 0.5 gives a symmetric decision between the inactive and active alternatives. It is an architectural setting for forecast-path selection, not a physical rainfall threshold. A continuous score could instead blend the two forecasts; the binary form makes the realized correction decision directly inspectable.

Let $a_t \in \mathbb{R}^{32}$ be the penultimate prediction-head representation and $q_t^{(0)}$ the backbone forecast. A separate correction head supplies

$$r_t = 0.3 \tanh(w_r^\top a_t + b_r), \qquad q_t = q_t^{(0)} + I_E \delta_t r_t, \tag{9}$$

where I_E enables the event branch. Both correction parameters are initialized to zero, preserving the initial backbone output. The residual bound is in log-rainfall units, not a fixed millimeter bound. This provides an identity-initialized correction path whose activation and magnitude can be inspected separately.

The bound 0.3 constrains the event branch's multiplicative change in one-plus-rainfall to $[\exp(-0.3), \exp(0.3)]$, approximately $[0.74, 1.35]$, before the feedback adjustment. It therefore permits an interpretable correction while limiting the branch's intervention. This bound and the gate cutoff are fixed across the reported ablations.

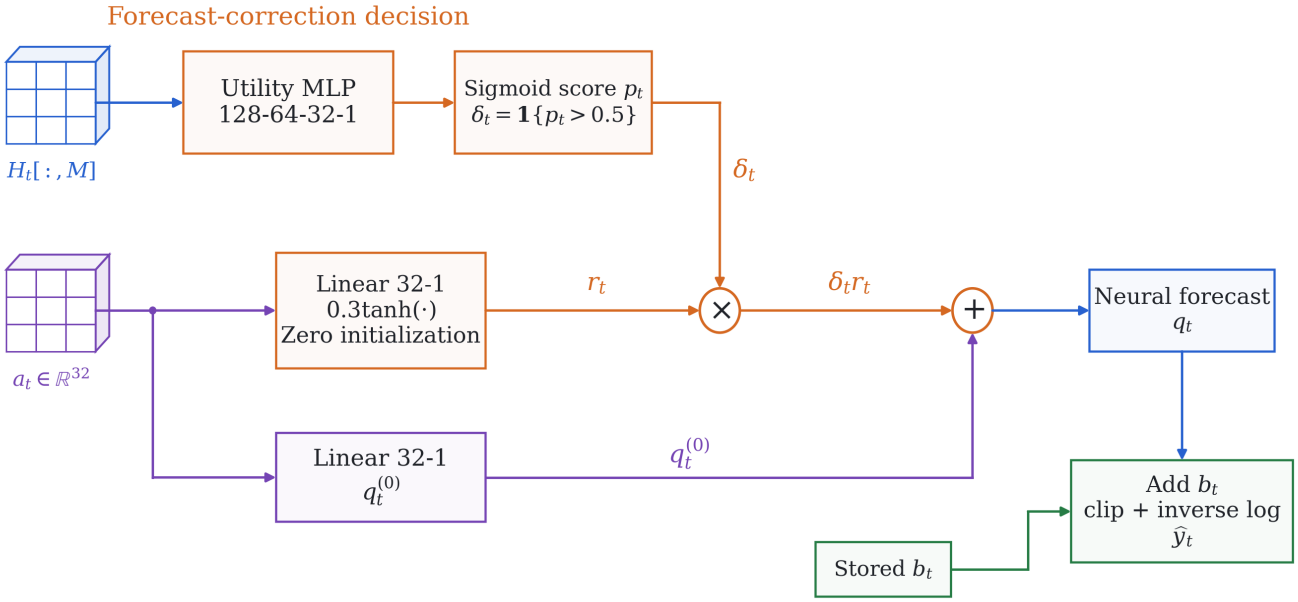

Figure 2. Event-gated heads and output fusion. The same 32-dimensional vector a_t feeds the base head and zero-initialized residual head. Only the residual is multiplied by δ_t before addition to $q_t^{(0)}$. The stored bias b_t is then applied before nonnegative projection and inverse transformation; its chronological update is shown in Figure 1.

3.3.2 *Sequential Error Feedback*

Let b_t be the bias available before issuing target t. The deployed forecast is

$$\widehat{\ell}_t = \max(0, q_t + I_F b_t), \qquad \widehat{y}_t = \exp(\widehat{\ell}_t) - 1, \quad (10)$$

where I_F enables feedback. Once y_t becomes available, the controller updates

$$e_t = \ell_t - \widehat{\ell}_t, \qquad b_{t+1} = I_F\big[b_t + 0.01\ \mathrm{sign}(e_t)\big], \quad (11)$$

with $\mathrm{sign}(0) = 0$. The bias starts at zero at the first evaluation target of each year and persists within that sequence. The fixed step limits the increment, not the cumulative bias.

The 0.01 log-unit step produces a gradual adjustment: one positive increment changes one-plus-rainfall by at most a factor $\exp(0.01) \approx 1.0101$ before projection. The sign update gives each newly observed error the same increment magnitude and uses its direction to correct subsequent outputs. The gain is shared by all feedback comparisons.

Raw neural outputs can be computed in evaluation batches, but the controller is applied sequentially in chronological order, with no state reset between batches. Current observations affect only subsequent forecasts. This requires the previous day's rainfall to be available before the next forecast; neural weights remain fixed during this sequence. The output controller is distinct from error-feature concatenation or memory-state injection. Algorithm 1 summarizes the deployed procedure.

3.4 LSTM Prediction Head

The transposed CNN sequence is used by a standard two-layer LSTM with input width 128 and hidden width 64. With v_s denoting its input at pooled position $s = 1, \ldots, M$, the recurrent equations are

$$i_s = \sigma(W_i v_s + U_i h_{s-1} + b_i), \quad (12)$$
$$f_s = \sigma(W_f v_s + U_f h_{s-1} + b_f), \quad (13)$$
$$o_s = \sigma(W_o v_s + U_o h_{s-1} + b_o), \quad (14)$$
$$\tilde{c}_s = \tanh(W_c v_s + U_c h_{s-1} + b_c), \quad (15)$$
$$c_s = f_s \odot c_{s-1} + i_s \odot \tilde{c}_s, \quad (16)$$
$$h_s = o_s \odot \tanh(c_s), \quad (17)$$

where hidden and cell states are initialized separately for each input window. Only the scalar output bias, not recurrent memory, is carried between target days. Dropout is 0.3 between recurrent layers and before the prediction head.

The final hidden state passes through FC layers of widths $64 \to 64 \to 32 \to 1$, with BN and ReLU after the first two:

$$q_t^{(0)} = f_{\mathrm{FC}}(h_M). \quad (18)$$

The event residual and chronological bias are applied to this output before nonnegative projection and inverse transformation.

3.5 Training Objective

To connect the gate directly to forecast utility, define the two millimeter-scale errors

$$L_t^{(0)} = |\exp(q_t^{(0)}) - 1 - y_t|, \tag{19}$$

$$L_t^{(1)} = |\exp(q_t^{(0)} + r_t) - 1 - y_t|. \tag{20}$$

For the event-enabled predictor, the objective is

$$\mathcal{L}_E = \frac{1}{N m_{\mathrm{tr}}} \sum_{t=1}^{N} \left[(1 - p_t) L_t^{(0)} + p_t L_t^{(1)} \right], \tag{21}$$

where N is the number of training windows and $m_{\mathrm{tr}} = 3.145833$ mm is the training-target mean. For a single normalized loss contribution, the direct derivative with respect to the gate logit is

$$\frac{\partial \mathcal{L}_{E,t}}{\partial \xi_t} = \frac{p_t(1 - p_t)}{m_{\mathrm{tr}}} \left(L_t^{(1)} - L_t^{(0)} \right). \tag{22}$$

Thus, the direct gate update favors activation when the corrected alternative has lower error and discourages it when correction increases error. The residual and shared representation are optimized jointly. For the backbone without this branch, $\mathcal{L}_0 = (N m_{\mathrm{tr}})^{-1} \sum_t L_t^{(0)}$. The event ablation therefore evaluates the residual branch together with its forecast-utility training strategy.

Training losses use unprojected neural outputs; all reported metrics use the actual binary gate, nonnegative projection and millimeter-scale forecast. The expected loss is a differentiable training objective, not a substitute for deployed evaluation. Adam with decoupled weight decay (AdamW) [22], gradient clipping and validation-based early stopping are used. The feedback step is fixed and has no neural gradient.

Algorithm 1: Event-feedback rainfall forecasting with causal correction

Data: Chronological 30-day windows, fixed neural weights, branch switches I_E, I_F
Result: One-day-ahead rainfall predictions
foreach *annual target sequence* **do**
  Initialize $b_t = 0$ at its first target;
  foreach *target t in chronological order* **do**
    Extract CNN features and compute backbone forecast $q_t^{(0)}$;
    Compute the forecast-correction trigger and residual when enabled;
    Set $q_t = q_t^{(0)} + I_E \delta_t r_t$;
    Issue $\widehat{y}_t = \exp(\max(0, q_t + I_F b_t)) - 1$;
    After observing y_t, compute e_t and update b_{t+1};
  end
end

4 Experimental Setup

4.1 Dataset

The study uses daily rainfall observations for Tengzhou City, Shandong Province, China, obtained from a government website and supplied as a comma-separated values (CSV) file. The file contains a single June–September rainfall series for 2013–2020, with date, county identifier, station identifier and daily rainfall. Table 2 summarizes the study area, recorded identifiers and audited observations. The identifiers describe the data source and are not used as model input features.

The June–September scope reflects the study's focus on summer and flood-season rainfall for flood-risk analysis. Selecting these months concentrates the present investigation on that seasonal application rather than year-round rainfall modeling. Annual seasons are processed separately; the selection does not assume that rainfall or flood risk is absent in other months.

Table 2. Summary of the rainfall dataset.

Item	Value
Study area	Tengzhou, Shandong, China
Source description	Government website download
Recorded station code	51224000
Recorded county code	370481
Available record period	June–September, 2013–2020
Raw records	1082
Exact duplicate excess rows	106
Unique retained dates	976
Explicit missing entries	0
Conflicting duplicate dates	0
Mean retained rainfall	3.164959 mm
Median rainfall	0 mm
Maximum rainfall	155 mm
Retained zero-rainfall ratio	76.54%

Removing 106 completely duplicate excess rows leaves 976 dates, with 122 continuous daily records per annual June–September season. No rainfall values are averaged, reassigned or newly imputed. Months outside this season are not filled with zeros. The numeric identifiers are retained exactly as supplied. The study-area description follows the acquisition

information associated with the data; numerical station codes are not used to infer a different geographical location or basin.

The experiment uses the 2013–2016 seasons for training and the 2017–2018 seasons for validation. After 14 days of feature warm-up and a 30-day input window, each year yields 78 target days, July 15–September 30, giving 312 training and 156 validation targets. Validation contains 129 dry/trace and 27 wet targets at 0.1 mm. All four arms in Section 5 are evaluated on these same dates, with the same causal features and five random seeds.

4.2 Baselines and Training Protocol

Four matched-backbone arms are considered:

- **A, CNN-LSTM**: backbone alone, $I_E = 0$, $I_F = 0$.
- **B, event branch**: backbone with utility-trained gated residual, $I_E = 1$, $I_F = 0$.
- **C, feedback**: backbone with chronological output correction, $I_E = 0$, $I_F = 1$.
- **D, ECF-CNN-LSTM**: both branches enabled, $I_E = 1$, $I_F = 1$.

Two predictors are fitted for each seed: the backbone A using $\mathcal{L}_0$ and the event-correction predictor B using $\mathcal{L}_E$. They share the 64-unit LSTM backbone design, feature preparation, optimizer and early-stopping rule. Common layers are initialized identically within each seed, and the residual head starts at zero. A/C contain 120,513 neural parameters and B/D contain 130,915. Seeds 0–4 are included without filtering. The ten fitted predictors yield twenty arm–seed forecast sequences when each is evaluated with feedback disabled and enabled. Table 3 lists the settings.

Training windows are shuffled in batches of 32, while evaluation uses chronological target order. An improvement must exceed 10^{-6}. At least 80 epochs are run; thereafter training stops when 30 or more consecutive epochs have not improved validation MAE, with a maximum of 300. The non-improvement counter and best-checkpoint search begin at epoch one, so an earlier checkpoint can still be retained. ReduceLROnPlateau uses patience 20 and a factor of 0.5. Checkpoint states are copied by value and reloaded to verify predictions.

The study uses a *fixed-network feedback ablation*. A and B are selected using their deployed validation MAE without feedback. C applies the chronological controller to the selected A checkpoint, and D to the selected B checkpoint, without refitting or reselecting either network. Thus A–C and B–D measure feedback at identical neural weights, while A–B measures the jointly trained residual branch and its forecast-utility objective. Backbone dimensions are matched; the detector and residual account for the additional B/D parameters. All reported comparisons retain the full five-seed set.

Table 3. Main experimental settings.

Parameter	Value
Input window length	30 days
Training / validation years	2013–2016 / 2017–2018
Minimum / maximum epochs	80 / 300
Learning rate	0.001
Batch size	32
LSTM hidden size	64
LSTM layers	2
Dropout	0.3
CNN channels; kernel	64, 128; width 3
Pooling width / stride	2 / 2
FC widths	$64 \to 64 \to 32 \to 1$
Event score threshold	0.5
Residual log bound	0.3
Gate training	Expected forecast MAE
Feedback step	0.01
AdamW (β_1, β_2)	$(0.9, 0.999)$
Weight decay; gradient clip	10^{-5}; norm 1
Early-stop patience	30 epochs
Random seeds	0, 1, 2, 3, 4

Every forecast uses a window ending on the preceding day. The current observation updates only later forecasts, and the controller resets at annual boundaries, not evaluation batches. This is rolling one-day prediction with newly available observations. Reproduced A/B training controls and reloaded-checkpoint checks establish consistency between the fitted networks and saved forecasts.

The reported runs use an AMD Ryzen 7 8845H CPU, one PyTorch thread and Windows, with Python 3.11.9, PyTorch 2.4.1, NumPy 1.25.1, pandas 2.0.3, scikit-learn 1.5.2 [23] and Matplotlib 3.9.3. No GPU training is used.

4.3 Evaluation Metrics

Predictions are transformed back to the original rainfall scale before evaluation. The main metrics are mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), rain/no-rain classification accuracy, and threshold accuracy under absolute error bounds of 0.1 mm, 1

mm, 5 mm, and 10 mm. Forecasting studies commonly compare absolute-error and squared-error measures [24]. This paper focuses on MAE because it is directly interpretable in millimeters and is less dominated by a small number of extreme deviations than RMSE [25]:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{23}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{24}$$

The coefficient of determination is $R^2 = 1 - \sum_t (y_t - \widehat{y}_t)^2 / \sum_t (y_t - \overline{y})^2$. Rain/no-rain accuracy applies the threshold 0.1 mm to both truth and forecast. Absolute-error coverage P_ϵ is the fraction of targets with $|y_t - \widehat{y}_t| \le \epsilon$, for $\epsilon = 0.1, 1, 5, 10$ mm.

The five rainfall-analysis bins are $[0, 0.1]$, $(0.1, 10]$, $(10, 25]$, $(25, 50]$ and $(50, \infty)$ mm. These are explicitly defined numerical analysis intervals, not official categorical rainfall labels; endpoints are retained consistently in all calculations and figures. Overall MAE is the count-weighted mean of interval MAEs, calculated before display rounding. Reported means and sample standard deviations are across seeds. The validation period contains 156 distinct dates; its 780 seed–date predictions are not treated as 780 independent observations.

In performance tables, boldface marks the best mean among the compared entries: lower errors and higher R^2, accuracy or coverage are preferred. Ranking uses unrounded values, and tied best results are all highlighted. Standard deviations and descriptive counts are reported without an optimization ranking.

Exploratory paired intervals use 2,000 bootstrap replicates, paired seed resampling and year-stratified moving time blocks. Within each year, sampled blocks are concatenated and truncated to 78 targets. The main block length is 30 days, with a seven-day sensitivity analysis. Per-seed metrics are computed before averaging. The primary intervals are reported in Table A1; they describe variation within the evaluated development setting.

5 Results and Analysis

5.1 Fixed-Network Feedback Ablation

Table 4 reports the main four-arm comparison on all 156 validation targets, with five seeds per arm. The complete ECF-CNN-LSTM model D has the lowest mean MAE and RMSE, 1.948576 and 7.370497 mm. Relative to the backbone A, these correspond to reductions of approximately 1.15% and 2.43%. The result evaluates the proposed combination of learned residual correction and chronological feedback under the stated fixed-network protocol.

Table 4. Fixed-weight feedback isolation on 156 validation targets. Mean errors in mm; seed standard deviations in parentheses. C reuses A, and D reuses B.

Arm	MAE	RMSE
A	1.971149 (0.054279)	7.553681 (0.525628)
B	1.951598 (0.100978)	7.370557 (0.912438)
C	1.964594 (0.062240)	7.558982 (0.525967)
D	**1.948576** (0.104537)	**7.370497** (0.917324)

The ablation makes each contribution explicit. The residual model B achieves MAE 1.951598 mm and RMSE 7.370557 mm, reducing A's mean errors by approximately 0.99% and 2.42%. Its mean wet-day MAE decreases from 11.213348 to 11.052910 mm. Applying feedback to the same B checkpoint then lowers MAE in all five seeds, by 0.003022 mm on average, producing D. Applying feedback to the same A checkpoint also lowers MAE in all five seeds, by 0.006555 mm on average. These same-weight pairs identify the sequential controller's contribution independently of changes in network weights.

The event-branch comparison improves MAE in three of five paired seeds and RMSE in three of five; the corresponding seeds need not be identical. For D versus B, the mean RMSE change is approximately -0.000060 mm, so the additional feedback effect is most clearly expressed by MAE. C also illustrates the metric distinction: its MAE improves over A while its mean RMSE increases. Standard deviations in Table 4 retain run-to-run variation. The five-of-five feedback MAE result describes paired seed outcomes; Appendix A separately quantifies temporal uncertainty.

5.2 Complete Metrics and Forecast Sequences

Table 5 reports all declared evaluation metrics. Alongside its lowest aggregate MAE and RMSE, D achieves wet-day MAE 11.049914 mm and dry/trace-day MAE 0.043645 mm. Aggregate MAE and RMSE summarize rainfall-amount accuracy, while conditional errors, occurrence accuracy and error coverage characterize complementary aspects of the forecasts.

Table 5. Complete mean validation metrics for the fixed-weight comparison. MAE/RMSE are in mm; P_ϵ is coverage at ϵ mm.

Metric	A	B	C	D
MAE	1.9711	1.9516	1.9646	**1.9486**
RMSE	7.5537	7.3706	7.5590	**7.3705**
R^2	0.0126	**0.0521**	0.0112	0.0520
Wet MAE	11.2133	11.0529	11.2230	**11.0499**
Dry MAE	0.0367	0.0467	**0.0268**	0.0436
Rain acc.	0.7987	0.8192	**0.8282**	0.8179
$P_{0.1}$	0.7628	0.7910	**0.7936**	0.7885
P_1	**0.8590**	0.8564	**0.8590**	0.8564
P_5	**0.9103**	0.9090	**0.9103**	0.9077
P_{10}	**0.9423**	**0.9423**	**0.9423**	**0.9423**

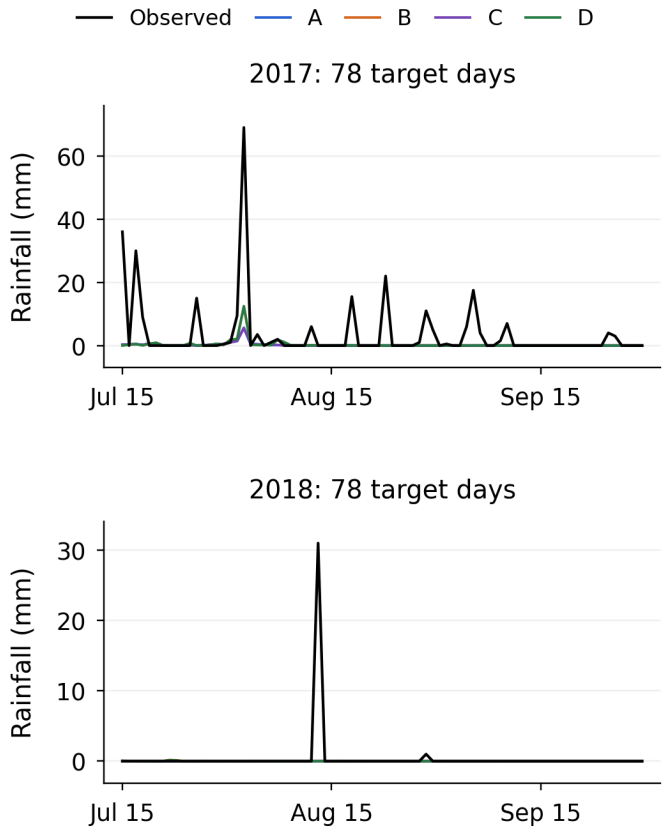

Figure 3. All 156 validation targets in 2017–2018, separated by year, for the fixed-network comparison. Lines show the mean forecast across five seeds for visualization. Tables report means of per-seed metrics, not metrics of the averaged forecast.

Figure 3 presents every target day rather than a selected example. The 2017 and 2018 sequences contain 25 and two wet targets, respectively. This seasonal contrast is retained in evaluation and in year-specific evidence files.

Table 6 adds causal reference forecasts on the same targets. The all-zero reference has MAE 2.003205 mm and RMSE 7.875534 mm; D reduces these to 1.948576 and 7.370497 mm. The previous-day and training-mean references contextualize the neural comparison without requiring information from the target day. The margin over the all-zero reference is small (about 2.7% in MAE), and the coefficients of determination in Table 5 are close to zero (0.011–0.052), so the forecasts explain little of the day-to-day variance of rainfall amounts.

Table 6. Causal reference forecasts on the same validation targets.

Reference	MAE (mm)	RMSE (mm)
Zero	**2.003205**	7.875534
Training mean	4.275641	**7.701741**
Previous day	3.314103	10.055774

5.3 Error Distribution and Training Behavior

Figure 4 shows the concentration of small errors together with the larger rainfall deviations. Feedback changes forecasts through the accumulated sign of past observed errors. Its contribution can therefore be evaluated at identical network weights, independently of changes in the learned CNN-LSTM representation.

Figure 5 shows the histories and selected epochs of the ten fitted A/B predictors used in this ablation. These are deployed validation MAE curves and can fluctuate as the network is updated. The minimum-duration rule runs training for at least 80 epochs, while the best checkpoint is retained from the complete trajectory. C and D reuse these A and B checkpoints, respectively, so their feedback effects do not require separate neural training curves.

5.4 Rainfall-Intensity Analysis

Table 7 gives the observed-rainfall counts and mean errors for every analysis interval. The counts are 129, 18, 5, 3 and one, totaling 156. Their count-weighted MAEs reproduce the overall result before rounding. Dry/trace errors retain nonzero precision rather than being rounded to an apparent perfect zero.

Figure 6 complements the overall comparison by showing where each model's error is concentrated. C

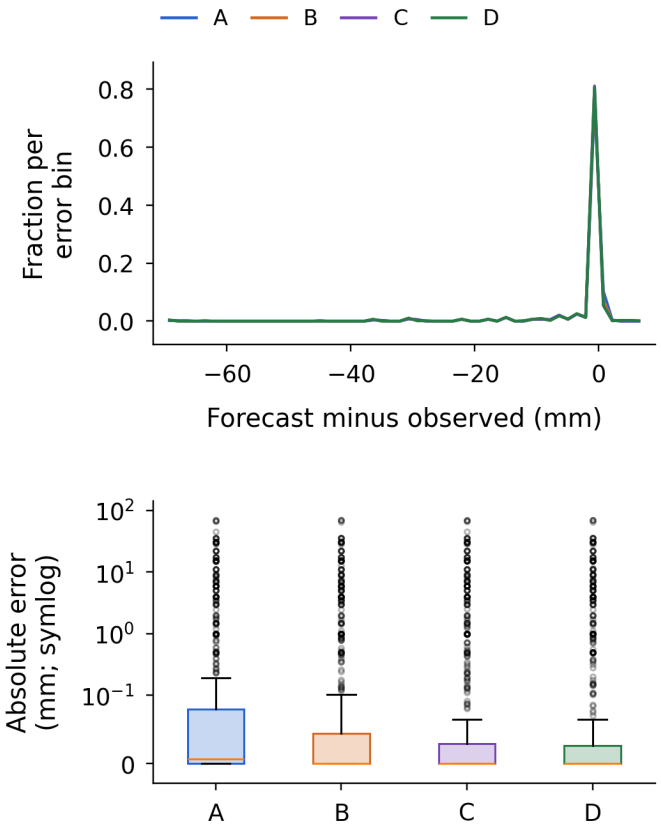

Figure 4. Signed-error frequencies and absolute-error distributions for the fixed-network validation comparison. The 780 seed–date predictions share 156 dates and are shown descriptively. The boxplot uses a symmetric-logarithmic scale with linear threshold 0.1 mm.

Table 7. Validation rainfall-interval counts and mean MAE (mm), fixed-weight protocol.

Interval (mm)	n	A	B	C	D
$[0, 0.1]$	129	0.0367	0.0467	**0.0268**	0.0436
$(0.1, 10]$	18	**3.4589**	3.5721	3.4657	3.5778
$(10, 25]$	5	**16.1847**	16.1852	16.1945	16.1867
$(25, 50]$	3	**32.0701**	32.1611	32.0789	32.1655
$(50, \infty)$	1	63.3661	56.7207	63.4303	**56.5172**

provides the lowest dry/trace error. For the single target above 50 mm, D's mean absolute error is 56.517153 mm, compared with 63.366114 mm for A. This one-target interval is interpreted at the case level, with its sample count stated explicitly. Weighting each interval's MAE difference from A by its share of the 156 targets shows where D's aggregate gain of 0.0226 mm originates: the dry/trace, $(0.1, 10]$, $(10, 25]$, $(25, 50]$ and > 50 mm intervals contribute $+0.0057$, $+0.0137$, $+0.0001$, $+0.0018$ and -0.0439 mm, respectively. The aggregate MAE reduction of the event branch (B and D relative to A) therefore arises from the single 69 mm target on 2017-08-02; in the other four intervals A has the lower MAE.

5.4.1 *Large-Rainfall Cases*

Table 8. Five largest validation targets and mean forecasts (mm), fixed-weight protocol. Bold marks the arm whose mean forecast is closest to the observed value in each row.

Date	y_t	A	B	C	D
2017-08-02	69.0	5.634	12.279	5.570	**12.483**
2017-07-15	36.0	**0.307**	0.033	**0.307**	0.033
2018-08-13	31.0	**0.026**	0.013	0.000	0.000
2017-07-17	30.0	0.457	**0.470**	0.457	**0.470**
2017-08-23	22.0	**0.014**	0.006	0.001	0.004

Table 8 lists the five largest validation targets and all four mean forecasts. For the 69 mm target, the mean forecast increases from approximately 5.634 mm for A to 12.483 mm for D, reducing the difference from the observation. The 31 mm target on 2018-08-13 is predicted near zero by all four arms. The controller updates from errors observed on preceding days, while each new target is predicted from its preceding rainfall window. These cases illustrate the operation of the correction paths and motivate further development with atmospheric covariates and intensity-aware objectives.

5.5 Gate Operating Characteristics

The gate trajectory contains 776 active decisions among 780 seed–window evaluations for B and D. Four seed models activate the residual throughout validation, and the fifth is active on 152 of 156 targets. Thus the learned policy favors broad residual correction in this setting. The gate therefore operates close to the always-active limiting case noted in Section 2, and these records do not show selective triggering. The empirical event-branch contribution is the gated residual and its joint training as a whole; the same-weight D–B comparison separately identifies the output-feedback contribution. The gate records document the realized residual-correction policy.

6 Discussion

The framework's practical contribution is the separation of learned forecast correction from chronological error-state correction. The bounded residual starts from the backbone's initial prediction, and its gate is trained using the prediction costs of the two correction alternatives. The output controller then uses newly available observations without changing neural weights. This yields an auditable perception–prediction–correction process for rolling daily forecasting.

The four-arm ablation supports this cooperative design.

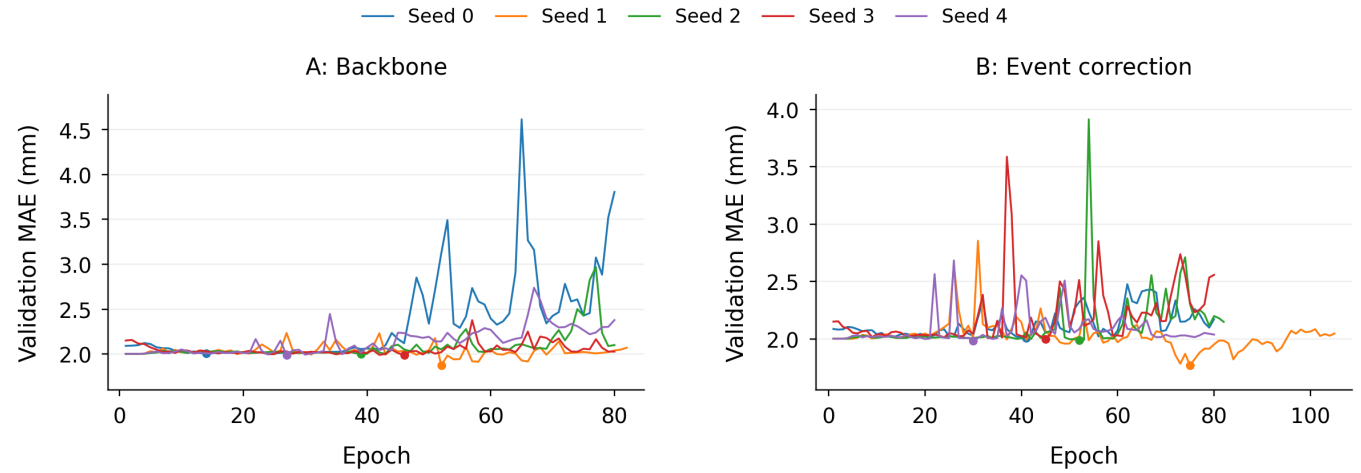

Figure 5. Validation MAE histories for the backbone A and event-correction predictor B, each with five seeds. Markers indicate selected checkpoints. C reuses A and D reuses B with chronological feedback enabled; all curves therefore correspond to the predictors used in Table 4.

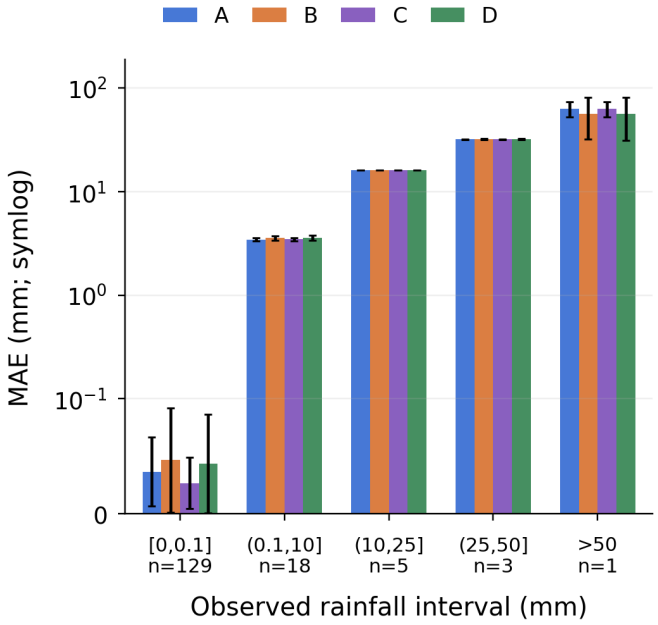

Figure 6. Validation interval MAE with seed standard deviations and unique target counts, fixed-network protocol. A symmetric-logarithmic scale with linear threshold 0.1 mm displays dry/trace and intense-rainfall errors together.

The residual branch reduces the backbone's mean MAE and RMSE, and chronological feedback further reduces MAE for every paired seed at the same network weights. Their combination, D, obtains the lowest mean MAE and RMSE among the four evaluated arms. These differences are small relative to the seed standard deviations in Table 4; the 95% paired intervals for B–A, D–B and D–C include zero (Table A1), and the MAE reduction of B and D relative to A originates from a single 69 mm target (Section 5.4). They are therefore exploratory evidence within this development setting. The architecture combines a standard LSTM encoder with a task-trained forecast correction and an explicit observation-driven output state.

The present contribution is a seasonal, single-series forecast-correction architecture supported by a completed matched ablation, causal state records and intensity-resolved evaluation. Seed means, standard deviations, paired temporal intervals and intensity counts provide the statistical context for the measured gains. The application is rolling daily rainfall prediction in a summer and flood-season setting.

Future research will develop selective and continuous correction policies, integrate meteorological covariates, and examine the framework alongside other forecasting models across broader temporal and spatial rainfall regimes. The present architecture and reproducible evaluation provide a concrete basis for those extensions.

7 Conclusion

This paper presented ECF-CNN-LSTM, combining a forecast-utility-trained binary residual branch with chronological prediction-error feedback. The model uses a 30-day causal window, a common CNN-LSTM backbone and a bounded correction initialized to preserve the backbone output. The feedback state is updated only after each observed outcome and acts on subsequent forecasts.

On 156 seasonal validation targets and five random seeds, the complete model D achieves the lowest mean MAE and RMSE in the fixed-network ablation:

1.948576 mm and 7.370497 mm, approximately 1.15% and 2.43% below the backbone. Feedback reduces MAE in all five paired runs for both predictor variants. The results support three connected contributions: a forecast-utility-trained residual branch, a causal sequential controller and a reproducible four-arm analysis of their combined effect. Richer meteorological observations and broader rainfall regimes provide directions for further development.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Necesito, I. V., Kim, D., Bae, Y. H., Kim, K., Kim, S., & Kim, H. S. (2023). Deep learning-based univariate prediction of daily rainfall: application to a flood-prone, data-deficient country. *Atmosphere, 14*(4), 632. [CrossRef]

[2] Patro, B. S., & Bartakke, P. P. (2024). Daily rainfall prediction using long short-term memory (LSTM) algorithm. *Journal of Agrometeorology, 26*(4), 509-511. [CrossRef]

[3] Samson, T. K., & Aweda, F. O. (2025). Comparative study of single and hybrid deep learning models for daily rainfall prediction in selected African cities. *Scientific Reports, 15*(1), 42718. [CrossRef]

[4] Küllahcı, K., & Altunkaynak, A. (2024). Maximizing daily rainfall prediction accuracy with maximum overlap discrete wavelet transform-based machine learning models. *International Journal of Climatology, 44*(10), 3405-3426. [CrossRef]

[5] Wu, H., Du, P., & Heng, J. (2024). Gated convolution with attention mechanism under variational mode decomposition for daily rainfall forecasting. *Measurement, 237*, 115222. [CrossRef]

[6] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780. [CrossRef]

[7] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation, 12*(10), 2451-2471. [CrossRef]

[8] Seidu, J., Ampomah, E. K., Asante, E., & Frimpong, K. A. (2026). Forecasting daily rainfall in Coastal Ghana with machine learning: an evaluation analysis of gated residual learning framework. *Frontiers in Climate, 8*, 1845902. [CrossRef]

[9] Farman, H., Hussain, M. A., Shaikh, S., Hassan, S., & Hassan, S. R. (2026). Dual framework for rainfall prediction: A multi-seed machine and deep learning evaluation across Pakistan's climatic regimes. *Scientific Reports, 16*, 20031. [CrossRef]

[10] Rojas-Campos, A., Wittenbrink, M., Nieters, P., Schaffernicht, E. J., Keller, J. D., & Pipa, G. (2023). Postprocessing of NWP precipitation forecasts using deep learning. *Weather and Forecasting, 38*(3), 487-497. [CrossRef]

[11] Miri, S. M., Kavianpour, M. R., & Alizadeh, M. J. (2026). Delta feature and random forest-enhanced LSTM-attention forecasts with probabilistic postprocessing for rainfall forecasting. *Scientific Reports, 16*, 24208. [CrossRef]

[12] Khan, M. I., & Maity, R. (2020). Hybrid deep learning approach for multi-step-ahead daily rainfall prediction using GCM simulations. *IEEE Access, 8*, 52774-52784. [CrossRef]

[13] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A, 379*(2194), 20200209. [CrossRef]

[14] Aderyani, F. R., Mousavi, S. J., & Jafari, F. (2022). Short-term rainfall forecasting using machine learning-based approaches of PSO-SVR, LSTM and CNN. *Journal of Hydrology, 614*, 128463. [CrossRef]

[15] Waqas, M., Humphries, U. W., Hlaing, P. T., Wangwongchai, A., & Dechpichai, P. (2024). Advancements in daily precipitation forecasting: a deep dive into daily precipitation forecasting hybrid methods in the tropical climate of Thailand. *MethodsX, 12*, 102757. [CrossRef]

[16] Chen, Y., Huang, G., Wang, Y., Tao, W., Tian, Q., Yang, K., ... & He, H. (2023). Improving the heavy rainfall forecasting using a weighted deep learning model. *Frontiers in Environmental Science, 11*, 1116672. [CrossRef]

[17] Narejo, S., Jawaid, M. M., Talpur, S., Baloch, R., & Pasero, E. G. A. (2021). Multi-step rainfall forecasting using deep learning approach. *PeerJ Computer Science, 7*, e514. [CrossRef]

[18] Wang, F., Tian, D., & Carroll, M. (2023). Customized deep learning for precipitation bias correction and downscaling. *Geoscientific Model Development, 16*(2), 535-556. [CrossRef]

[19] You, X. X., Liang, Z. M., Wang, Y. Q., & Zhang, H. (2023). A study on loss function against data imbalance in deep learning correction of precipitation forecasts. *Atmospheric Research, 281*, 106500. [CrossRef]

[20] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning, PMLR 37*, 448-456. https://proceedings.mlr.press/v37/ioffe15.html

[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research, 15*(1), 1929-1958. https://jmlr.org/papers/v15/srivastava14a.html

[22] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. [CrossRef]

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of machine Learning research, 12*, 2825-2830. https://jmlr.org/papers/v12/pedregosa11a.html

[24] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*(4), 679-688. [CrossRef]

[25] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research, 30*, 79-82. [CrossRef]

Appendix

A Paired Temporal Variation

Table A1 reports the exploratory 30-day block intervals for the fixed-network comparison. The same resampled dates and seeds are used for each pair. B–A, D–B and D–C intervals include zero for both metrics. C–A has a negative MAE interval and a positive RMSE interval, reflecting its metric trade-off. Thus the five-of-five feedback MAE outcomes describe consistency across the retained seeds, while the temporal analysis quantifies uncertainty in the size and transfer of those gains. With seven-day blocks, the MAE interval signs are unchanged, while the C–A RMSE interval includes zero. These development-period intervals are exploratory and do not adjust for repeated model selection.

Table A1. Exploratory 95% paired intervals using 30-day seasonal blocks and seed resampling (2,000 replicates). Differences are left arm minus right arm, in mm.

Pair	Δ MAE	Δ RMSE
B-A	$[-0.124415, 0.018977]$	$[-0.958935, 0.019878]$
C-A	$[-0.019410, -0.000706]$	$[0.000203, 0.017337]$
D-B	$[-0.008014, 0.001172]$	$[-0.010503, 0.006773]$
D-C	$[-0.118073, 0.029864]$	$[-0.974874, 0.018767]$

Xiaoqi Yin is the corresponding author, a professor and master's supervisor in Electronic Science and Technology at Huai'an University. Her research focuses on signal processing and measurement technology. (Email: hy_xuebao2009@126.com)

Jing Zhou received a bachelor's degree in Electronic Science and Technology and is currently pursuing a master's degree at Huai'an University. (Email: 2960088350@qq.com)

Ruping Liu is a master's student at Huai'an University. She received her bachelor's degree in Electronic Science and Technology. (Email: 1774179967@qq.com)

Chenxi Ma received a bachelor's degree in Measurement and Control Technology and Instrumentation and is pursuing a master's degree at Huai'an University. (Email: 1564817993@qq.com)

Chen Li received a bachelor's degree in Electronic Information Engineering and is currently pursuing a master's degree at Huai'an University. (Email: 1696915652@qq.com)