water

MDPI

Article

GHDFloodNet: An Advanced Model for Improved Short-Term Flood Forecasting

Mohammad Abdullah-Al-Shafi [1,*], Golam Sorwar [1], Ali Reza Alaei [1] and Masrur Ahmed [2]

[1] Faculty of Science and Engineering, Southern Cross University, Gold Coast 4225, Australia; golam.sorwar@scu.edu.au (G.S.); ali.alaei@scu.edu.au (A.R.A.)
[2] Science and Insights Division, Department of Climate Change, Energy, the Environment and Water, NSW Government, Sydney 2141, Australia; masrur.ahmed@dcceew.nsw.gov.au
* Correspondence: alshafi08@gmail.com, m.shafi.11@student.scu.edu.au

Abstract

Accurate short-term flood forecasting is vital for effective risk management and early warning systems. However, many data-driven models struggle to generalise with limited historical data and fail to consistently capture complex temporal dependencies across varying forecasting horizons. To address these challenges, this study proposes GHDFloodNet (Generalised Hybrid Data-limited Flood Prediction Network), a hybrid deep learning framework designed for robust multi-step-ahead forecasting. GHDFloodNet integrates First-Order Model-Agnostic Meta-Learning (FOMAML) with a Temporal Fusion Transformer (TFT) to enable rapid task adaptation and effectively capture long-range temporal dependencies and variable interactions. To further enhance predictive consistency, the framework incorporates a bidirectional Long Short-Term Memory (BiLSTM) network augmented with an additive attention mechanism and static feature fusion as a core learner within a meta-ensemble architecture. Bayesian hyperparameter optimisation within an AutoML framework identifies optimal model configurations, while a dedicated data handling layer with real-time augmentation improves stability under non-stationary conditions. The framework was evaluated for multi-horizon water level forecasting across four lead time ranges (1–6 h, 6–12 h, 12–24 h, and 24–48 h) using rainfall and lagged water level observations as primary inputs. Experimental results demonstrate that GHDFloodNet achieves robust, nearly invariant error distributions across the full 1–48 h forecast window, reporting an MSE of 0.53–0.55, RMSE of 0.72–0.74, and MAE of 0.35–0.36. Furthermore, the model exhibits stable goodness-of-fit, with R^2 and NSE values consistently ranging from 0.44 to 0.47 across all lead times, significantly outperforming conventional baselines, which typically exhibit pronounced error escalation at longer horizons. Overall, GHDFloodNet demonstrates that horizon-independent forecast reliability can be architecturally engineered, offering critical value for operational flood forecasting where consistent performance across all lead times outweighs peak short-range precision.

Keywords: short-term flood forecasting; GHDFloodNet; meta-learning; transformer-based models; ensemble learning

Check for updates

Academic Editor: Gwo-Fong Lin

Received: 9 May 2026
Revised: 15 June 2026
Accepted: 23 June 2026
Published: 28 June 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) license.

1. Introduction

Flooding is one of the most destructive natural hazards worldwide, posing severe threats to human life, infrastructure, ecosystems, and socio-economic stability [1]. The increasing frequency and intensity of flood events, driven by climate change, land-use

change, and rapid urbanisation, have amplified the need for accurate and timely flood forecasting systems [2]. Reliable short-term flood forecasts are essential for disaster risk reduction, enabling early warning dissemination, emergency response planning, reservoir operation, and real-time water resource management [3]. Forecasts at sub-daily to multi-day horizons are particularly important for reducing loss of life and infrastructure damage, especially in small to medium-sized catchments where hydrological responses are rapid and highly nonlinear [4]. Despite its importance, short-term flood forecasting remains a challenging task due to the complex, dynamic, and non-stationary nature of hydrological processes [3,5]. These characteristics require forecasting frameworks to effectively capture nonlinear catchment responses and rapidly evolving hydrometeorological conditions [6]. Consequently, the choice of modelling approach becomes a central issue in the development of reliable short-term flood forecasting systems [7]. Traditional flood forecasting methods are commonly grouped into physically based and stochastic models; however, their limited capacity to consistently represent nonlinear hydrological dynamics and rapidly evolving conditions constrains their predictive reliability, thereby motivating the exploration of more advanced data-driven and hybrid modelling approaches [8].

Physical models, such as HEC-HMS, SWAT, and distributed hydrological models, aim to represent the processes of the hydrological cycle through mathematical formulations. Although these models offer physical interpretability, they suffer from several important limitations [3]. These models require extensive high-resolution spatial and hydrometeorological data for calibration and validation, and their performance is highly sensitive to the accuracy of physical parameters, which are often difficult to measure [9]. Model performance can degrade significantly when applied to ungauged or data-scarce basins, and parameter transferability across catchments and flood events remains a persistent challenge [9]. Moreover, the computational complexity of physically based models can hinder their real-time applicability for short-term forecasting [9]. In parallel, stochastic models, including autoregressive and regression-based approaches, have been widely used for flood forecasting due to their simplicity and computational efficiency. These methods generally rely on assumptions of linearity and stationarity, restricting their ability to represent the nonlinear, dynamic, and rapidly evolving nature of flood processes [3]. Consequently, their predictive skill often declines during extreme events and at longer lead times, reducing their effectiveness for operational flood early warning.

Recent advances in machine learning (ML) models have shown considerable potential for overcoming some of these limitations by learning complex nonlinear relationships directly from data. In particular, models such as long short-term memory (LSTM) networks, convolutional neural networks (CNNs), and Transformer-based architectures have demonstrated improved performance in flood and runoff forecasting [10]. Nevertheless, conventional deep learning models face critical challenges when applied to real-world hydrological forecasting. They typically require large volumes of historical data for effective training, making them vulnerable in data-scarce environments where flood events are infrequent [11]. Additionally, their performance can be sensitive to non-stationary hydrological conditions, such as changes in climate or land use, and they often exhibit limited generalisation across different basins or flood regimes [11]. To mitigate these challenges, hybrid and ensemble modelling approaches have gained increasing attention in hydrological forecasting [3]. By combining multiple models or learning paradigms, ensemble frameworks can improve and enhance predictive accuracy across varying conditions. Hybrid ensemble models leverage the complementary strengths of different architectures, enabling more effective representation of both short-term dynamics and long-term dependencies. However, existing ensemble approaches often lack systematic mechanisms for rapid adaptation to new tasks, automated optimisation, and real-time data variability,

which are essential for operational short-term flood forecasting [11]. It is important to distinguish short-term flood forecasting from conventional medium- and long-term hydrological forecasting. Models developed for longer forecasting horizons primarily focus on capturing seasonal variability, climatic trends, and gradually evolving hydrological processes over timescales ranging from several days to months [12].

In contrast, short-term flood forecasting requires the accurate representation of rapidly changing rainfall–runoff dynamics, abrupt river-level fluctuations, nonlinear catchment responses, and event-scale hydrological behaviour occurring within minutes to hours [13]. These characteristics introduce substantial challenges related to temporal dependency modelling, data non-stationarity, and forecasting stability under rapidly evolving conditions. Consequently, model architectures that perform effectively for long-term forecasting may not necessarily provide reliable performance for short-term flood prediction, particularly during extreme hydrological events, when hydrological responses can change rapidly [14]. Furthermore, unlike many forecasting frameworks that experience progressive performance degradation as prediction horizons increase, operational short-term flood forecasting requires consistent predictive reliability across successive lead times to support timely early-warning and emergency-response decisions [6,15]. This highlights the need for a unified and adaptive forecasting framework that can address data scarcity, non-stationarity, and rapid flood dynamics while maintaining predictive accuracy across multiple lead times.

Despite substantial advances in hydrological forecasting, the development of models capable of simultaneously achieving high predictive accuracy, cross-catchment adaptability, robustness under data-limited conditions, and stable performance across extended forecasting horizons remains a significant challenge. Existing studies have generally focused on improving individual aspects of forecasting performance, such as feature representation, temporal dependency modelling, or ensemble prediction, often without providing a holistic solution that addresses these challenges collectively. Consequently, a critical assessment of the current state of the literature is necessary to identify remaining methodological gaps and establish the motivation for the proposed GHDFloodNet framework.

The primary goal is to develop a robust, transferable deep learning framework that maintains high temporal invariance and accuracy despite data limitations. To achieve this, we propose GHDFloodNet, a Generalised Hybrid Data-limited Flood Prediction Network designed to forecast water levels across four distinct short-term lead-time ranges: 1–6 h, 6–12 h, 12–24 h, and 24–48 h. The methodology integrates First-Order Model-Agnostic Meta-Learning (FOMAML) for rapid few-shot adaptation across basins, a Temporal Fusion Transformer (TFT) for capturing long-range temporal dependencies, and a Bidirectional Long Short-Term Memory (BiLSTM) meta-ensemble for dynamic feature synthesis, all optimised via an automated Bayesian hyperparameter tuning process. This framework is specifically evaluated on heterogeneous catchments characterised by non-stationary hydrological conditions and limited historical records, utilising rainfall and lagged water level observations as the primary inputs. Aligned with these objectives, the specific contributions of this study are as follows:

- **Unified GHDFloodNet (Generalised Hybrid Data-limited Flood Prediction Network) Architecture**: A comprehensive, end-to-end forecasting framework is proposed that integrates multiple advanced learning paradigms into a cohesive architecture, enabling robust short-term flood prediction across diverse hydrological conditions.
- **Integration of MAML and TFT**: GHDFloodNet combines First-Order MAML with a TFT to enable rapid adaptation to new flood forecasting tasks while effectively capturing complex temporal dependencies, variable interactions, and non-stationary behaviour in hydrological time series.

- **Multi-Horizon BiLSTM with Attention for Enhanced Temporal Encoding**: A BiLSTM network with an additive attention mechanism is integrated as a complementary base learner, providing recurrent temporal processing that captures sequential dependencies and delayed hydrological responses across multiple forecast horizons.

Together, these contributions position GHDFloodNet as a scalable and operationally relevant solution for next-generation short-term flood forecasting and disaster risk reduction. By integrating meta-learning, transformer-based temporal modelling, recurrent sequence learning, attention mechanisms, and automated hyperparameter optimisation within a unified framework, the proposed approach addresses several persistent limitations discussed in Section 2 that continue to hinder the reliability, transferability, and long-term stability of existing flood forecasting systems.

The remainder of this paper is structured as follows. Section 2 presents the literature review of the study. Section 3 introduces the proposed GHDFloodNet architecture, outlining its overall modular design and the interaction between the meta-learning, transformer, and ensemble layers, followed by in-depth descriptions of the MAML meta-learning layer, the Temporal Fusion Transformer for temporal dependency modelling and interpretability, the meta-ensemble stacking strategy, and the AutoML framework based on Bayesian hyperparameter optimisation. Section 3 also describes the study area and datasets, detailing the hydrological characteristics of the selected catchments, data sources (rainfall, streamflow, and meteorological variables), preprocessing procedures, input-output configurations for short-term forecasting, and evaluation metrics. Section 4 presents the experimental results, including comparisons with benchmark models and an analysis of lead time sensitivity with emphasis on short-term forecasting horizons. Section 5 presents the discussion and limitations of the proposed model. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Literature Review

Recent advancements in flood forecasting reflect a shift from traditional process-based models to sophisticated data-driven architectures. Among these, ensemble methods and deep learning techniques have shown considerable promise, particularly for short-term and multi-step prediction. These models have demonstrated strong predictive capability under basin-specific settings; however, their transferability and robustness remain unresolved challenges [16–19].

Given the broad and rapidly expanding literature, this review adopts a targeted critical synthesis rather than an exhaustive catalogue of all published studies. The reviewed papers were selected based on the following criteria:

(i) direct relevance to short-term flood forecasting;
(ii) methodological relevance to the present work, particularly ensemble, hybrid, encoder–decoder, meta-learning, and deep learning architectures;
(iii) representation of both single-basin and cross-basin modelling paradigms;
(iv) availability of sufficient methodological and performance details to support comparative technical analysis.

Using these criteria, the selected studies are critically synthesised to highlight the dominant methodological directions and persistent limitations in short-term flood forecasting. Particular emphasis is placed on model generalisation, hybridisation depth, sensitivity to data constraints, and forecasting consistency across lead times [16–25], as these aspects directly inform the design of the proposed GHDFloodNet framework.

An intelligent committee ML (ICML-FF) framework integrating radial basis function neural networks (RBF-NN), adaptive neuro-fuzzy inference systems (ANFIS), SVMs, and LSTMs is proposed by [18]. However, their approach primarily relied on simple averaging

or weighted averaging methods (K-Nearest Neighbours, Genetic Algorithm) to combine outputs. Their results demonstrated that ensemble learning can outperform individual models and enhance forecasting accuracy for early warning systems. A key limitation in their work is that simple averaging cannot effectively capture the unique strengths of individual models, often resulting in suboptimal performance when learners are unbalanced. Moreover, in terms of cross-domain performance, most ensemble and deep learning-based flood forecasting models are trained in a basin-specific manner and rely on static model parameters. As a result, their performance often degrades when applied to unseen catchments, extreme events, or changing hydrological regimes. The ensemble framework proposed by [18], while effective within the study basin, does not incorporate explicit mechanisms for rapid adaptation or cross-basin transfer learning, limiting its applicability in data-scarce or ungauged environments.

Regarding hybridisation, existing ensemble approaches primarily focus on combining outputs from multiple base learners using linear or heuristic weighting strategies. Although weighted averaging via Genetic Algorithm or K-Nearest Neighbours improves performance over simple averaging, such methods do not explicitly model higher-order temporal dependencies or interactions among ensemble members. Moreover, these frameworks generally lack attention mechanisms and advanced temporal representation learning, which are critical for capturing long-range dependencies, abrupt flood peaks, and variable importance in hydrological time series. Data limitations remain a persistent challenge. Many data-driven flood forecasting models require long historical records for training and are sensitive to missing data, noise, and non-stationary conditions [26,27]. While ensemble learning can partially mitigate overfitting, existing approaches do not explicitly address real-time data variability or dynamically augment limited datasets, which restricts their stability during rare or extreme flood events. A deterministic LSTM encoder–decoder model is proposed by [17], then extended to a probabilistic version by minimising the pinball loss at three quantiles (0.05, 0.5, 0.95) while keeping the same encoder–decoder architecture; Bayesian optimisation is used once to pick layer sizes, and wavelet coefficients are either linearly or nonlinearly pre-selected. Their approach integrates LSTM networks with dense networks, utilising a sequence-to-sequence architecture to predict future discharge. While the research utilised Bayesian optimisation to tune models for specific basins, their approach required training and optimising distinct models for each catchment. The reported performance showed high variability (KGE ranging from 0.308 to 0.979) across the three selected basins. This indicates that models optimised for one specific hydrological characteristic (e.g., high snowmelt) may not transfer effectively to others (e.g., rainfall-dominated). The reliance on basin-specific optimisation creates a computational bottleneck and limits the model's ability to generalise to unseen or data-scarce basins (few-shot learning) without extensive retraining. Architecturally, the study primarily relies on the LSTM-DN encoder–decoder setup. LSTMs, while powerful for modelling temporal dependencies, can struggle with extremely long sequences and may struggle to capture complex, global attention patterns compared to newer architectures.

Furthermore, the study does not explore deep architectural ensembles (e.g., combining diverse deep learning paradigms), instead relying on a single dominant architecture type per configuration. The study utilised a static partition of the CAMELS dataset. While they simulated synthetic precipitation forecasts to mimic uncertainty, the core data handling relied on fixed training sets without real-time augmentation strategies to handle non-stationary distributions or shifting climate patterns.

A hybrid deep learning SA-CNN-LSTM model is proposed by Zhou, Chen, and Liu [16]. The model integrates three distinct components: a CNN for extracting local features, an LSTM for modelling temporal dependencies, and a Self-Attention (SA) mecha-

nism to weigh the importance of different timesteps and input features. Despite its strong predictive performance, several limitations regarding cross-domain performance can be identified. The SA-CNN-LSTM model is trained and evaluated on a single basin with dense rainfall station coverage and relatively homogeneous hydro-climatic conditions. Model parameters, including the input window length and network architecture, are optimised specifically for this basin, limiting transferability to ungauged or data-sparse catchments. While the authors demonstrate robustness across different flood magnitudes and lead times, cross-basin or cross-climate generalisation is not explored. As a result, the model's ability to adapt to unseen hydrological regimes, non-stationary climate conditions, or basins with different response times remains uncertain. Moreover, performance degrades markedly after a 5 h lead time (NSE falls from 0.99 to 0.77), indicating limited extrapolation capability when meteorological forcing changes. Although the study attempts a deep architectural hybrid, the results indicate instability in this integration. Specifically, the SA-CNN component performed worse than the standard CNN at longer lead times (3–7 h). The interactions between components are predefined rather than dynamically learned at a meta-level. The self-attention mechanism improves feature and timestep weighting. However, it does not address structural limitations, such as error accumulation in multi-step forecasting or adaptive model selection across varying hydrological states. The static normalisation strategy assumes stationarity in data distributions, which may not hold under real-time operational forecasting or climate-induced variability.

A Transformer-based meta-learning framework (MetaTrans-FSTSF) is proposed by [19]. From a generalisation perspective, although MetaTrans-FSTSF improves adaptability across tasks, its performance still depends on the diversity and representativeness of tasks used during meta-training. Extreme climatic or hydrological conditions that are poorly represented in the meta-training distribution may reduce transfer effectiveness, limit reliability when applied to fundamentally different flood regimes. Moreover, the model's reliance on a complex Transformer architecture, while powerful, introduces interpretability challenges. The few-shot paradigm implies that the model's upper bound of accuracy is constrained by the very small number of samples available for adaptation; if the few available samples contain noise or do not represent the full spectrum of flood events, the model's predictive capability is inherently limited. Additionally, the study does not explicitly detail robust data handling layers for real-time noise filtering or augmentation.

Wavelet-based ANN and LSTM (WANN, WLSTM) models are proposed by [23] for river discharge prediction by decomposing monsoon-season discharge data using wavelet transforms and feeding the reconstructed sub-series into neural networks. While wavelet preprocessing improved long-lead forecasting accuracy, the approach is strongly basin-specific and season-dependent, limiting transferability to other climatic regimes or non-monsoon periods. Furthermore, hybridisation is confined to sequential preprocessing rather than end-to-end learning, and the reliance on discharge-only data restricts consistency under data scarcity.

The research by [20] integrates geographic information and the SCS-CN hydrological concept with LSTM networks to enhance runoff forecasting in a complex mountainous basin using satellite precipitation. Although feature engineering improved short-term accuracy, the model's effectiveness depends heavily on the quality of empirical SCS-CN parameters and satellite rainfall products. This constrains transferability to basins with different geomorphology. Moreover, the model relies on GSMaP-NRT satellite precipitation, which has a 5 h latency and a short historical record.

An ensemble machine learning framework combined with hydrological modelling and remote sensing (Sentinel-1 SAR) for flood prediction and damage assessment is proposed in [21]. Ensemble Prediction Systems (EPS) improved stability and accuracy; however, the

framework is computationally intensive due to coupling with the HEC-HMS model and is highly dependent on high-resolution remote sensing and historical flood inventories. Hybridisation here is largely at the decision level rather than through unified learning architectures, limiting scalability and real-time applicability in data-poor regions.

Another study by [25] investigated the use of CHIRPS satellite precipitation, combined with intrinsic time-scale decomposition (ITD) and ML models (RF, CHAID, GMDH) for multi-step streamflow forecasting. The hybrid approach improved prediction accuracy, particularly at longer lead times. However, its decomposition-based formulation requires long, continuous historical records, which are often unavailable in data-scarce or ungauged basins. As a result, its applicability and transferability across different catchments may be limited.

A pre-training enhanced deep learning framework was introduced by Xia and Lu [22] for short-term flood prediction, employing masked pre-training to better capture historical temporal patterns without extending the input window. While this improves temporal representation learning, the model is primarily validated on small and medium basins and requires extensive historical data for effective pre-training.

An encoder–decoder LSTM model is examined by [24]. Their results highlight that model performance strongly depends on the diversity and representativeness of the training dataset. Although regional ED-LSTM models outperform local models, their reliability deteriorates in hydro-climatically heterogeneous or arid regions, underscoring persistent data dependency and limited consistency in extreme or unseen conditions. Based on the preceding review, existing short-term flood forecasting models exhibit four critical limitations:

- Poor cross-basin generalisation: Most models rely on basin-specific, static parameters that fail to transfer to ungauged or data-scarce environments.
- Shallow hybridisation and temporal representation: Existing ensembles rely on simple linear weighting and lack advanced attention mechanisms to capture long-range dependencies and abrupt flood peaks.
- Vulnerability to data scarcity and non-stationarity: Models largely lack real-time data augmentation or dynamic adaptation to distributional shifts caused by extreme events.
- Manual tuning and computational bottlenecks: Basin-specific hyperparameter optimisation limits scalability and operational applicability.

To address the above limitations in existing systems, this study proposes a data-driven approach—GHDFloodNet. GHDFloodNet learns basin-invariant initialisations during meta-training across heterogeneous catchments, enabling rapid adaptation to new basins using only a small number of samples. This few-shot learning capability directly addresses the transferability limitations observed in LSTM encoder–decoder models, probabilistic LSTM frameworks, and basin-specific hybrid architectures, making the framework more suitable for ungauged or data-scarce environments.

3. Materials and Methods

3.1. Overall Architecture Design

The GHDFloodNet architecture is developed as a modular, multi-layered framework that systematically addresses the four key limitations identified in Section 2. Rather than relying on a single predictive paradigm, GHDFloodNet integrates meta-learning, deep temporal representation, ensemble intelligence, automated optimisation, and adaptive data handling into a unified pipeline. A schematic overview is presented in Figure 1.

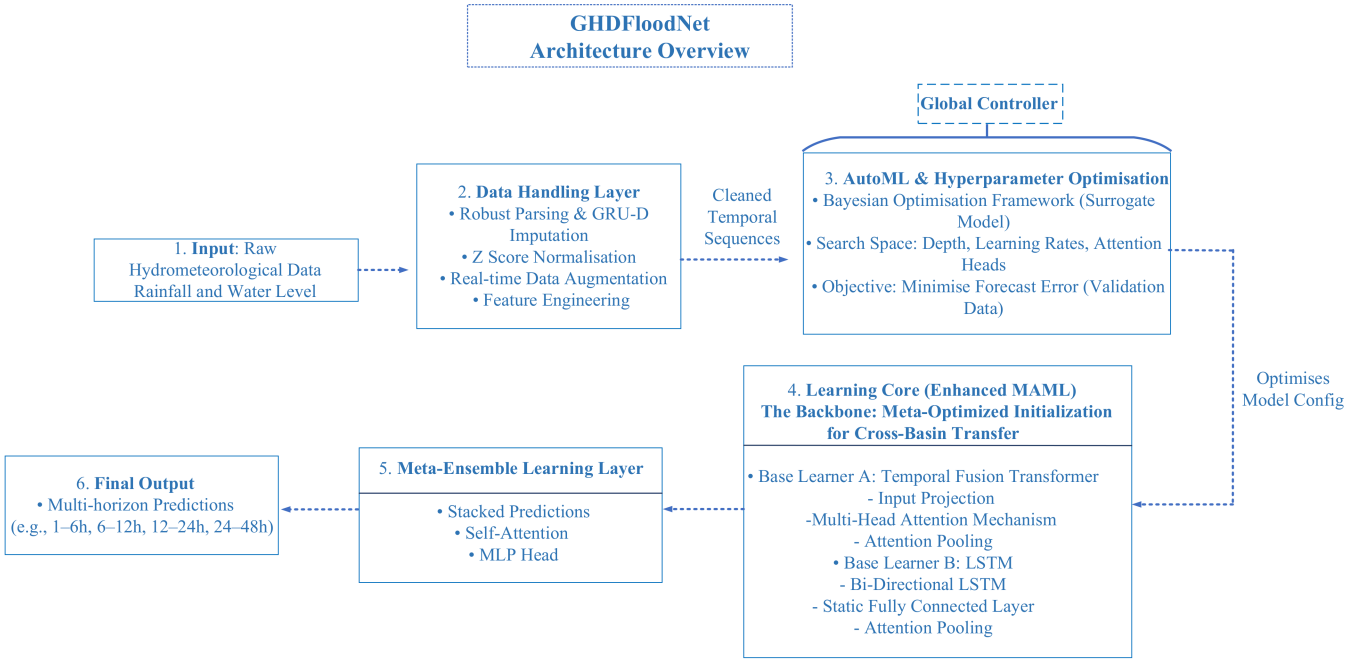


Figure 1. GHDFloodNet architecture.

The Data Handling Layer forms the foundation by performing real-time data augmentation and preprocessing to reduce noise and manage missing values, thereby addressing data scarcity and non-stationarity (Limitation 3). Building on this, the MAML layer optimises transferable initial parameters across diverse catchments, enabling rapid adaptation to new basins with limited data and resolving poor cross-basin performance (Limitation 1). To overcome shallow temporal representation (Limitation 2), the TFT is integrated to learn intricate temporal patterns, long-range interactions, and variable importance. These outputs are further processed through an end-to-end trainable Meta-Ensemble stacking strategy, where TFT and BiLSTM predictions are horizontally stacked and processed via horizon-specific meta-learner networks. This replaces static linear weighting with learned temporally aware interactions. Finally, AutoML-driven Bayesian Hyperparameter Optimisation eliminates the computational bottlenecks of manual tuning (Limitation 4).

3.1.1. MAML Task Setup

The Learning Core utilises an Enhanced MAML strategy (*EnhancedMAML* class) to optimise the initial weights of the model for improved cross-basin transferability.

- **Task Formulation**: In this context, a task T_i is defined as a regression problem on a specific batch of sequential data $D_i = \{(X_{seq}, X_{static}), Y\}$. The distribution of tasks $p(T)$ is implicitly defined by the data loader's sampling strategy across the temporal domain.
- **Inner Loop (Adaptation)**: For a given batch, the model parameters θ are cloned to create task-specific parameters θ'_i. The adaptation rule follows: $\theta'_i \leftarrow \theta - \alpha \nabla_\theta L_{Ti}(f_\theta)$, where α is the inner learning rate. The implementation performs N = 2 gradient steps using the *AdamW* optimiser with *CosineAnnealing* scheduling to ensure stable adaptation within the task.
- **Outer Loop (Meta-Update)**: The meta-parameters θ are updated by backpropagating through the inner loop updates. The meta-objective minimises the loss of the adapted parameters θ'_i on the same task (support set), effectively optimising for a parameter initialisation that is sensitive to hydrological gradients: $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{Ti} L_{Ti}(f_{\theta' i})$. The framework utilises *torch.cuda.amp* (Automatic Mixed Precision) for the inner loop to

enhance memory efficiency. Gradient clipping (max norm = 1.0) is applied to mitigate exploding gradients, a common issue in recurrent architectures.

3.1.2. TFT Architecture

The TFT variant implemented here is optimised for computational efficiency while preserving the ability to model long-range dependencies.

- **Input Processing**: Sequential inputs $X \in R^{B \times T \times F}$ (Batch, Time, Features) are projected into a high-dimensional space d_{model} via a linear layer followed by Batch Normalisation.
- **Static Covariate Enrichment**: Static features $S \in R^{B \times D}$ are projected to d_{model} and added element-wise to the sequential embeddings, ensuring context infusion at every time step.
- **Temporal Encoder**: The core utilises a stack of $L = 3$ *TransformerEncoderLayer* blocks with Pre-Layer Normalisation (*norm_first* = *True*) to stabilise deep network training.

  Attention Mechanism: Multi-head attention with $H = 8$ heads.
  Feed-Forward Network (FFN): The dimension is optimised to d_{model} to reduce parameter count.
  Activation: Gaussian Error Linear Unit (GELU).

- **Sequence Aggregation**: Instead of a variable selection network, a simplified Attention Pooling mechanism is employed to aggregate the temporal dimension: $Context = \sum_{t=1}^{T} Softmax\ (W_2 \cdot Tanh(W_1 h_t))\ h_t$, where h_t is the transformer output at time t.

3.1.3. Ensemble Fusion Formula

The fusion strategy moves beyond simple weighted averaging by employing an attention-based meta-learner. This allows the model to dynamically weigh the contributions of the TFT and BiLSTM base learners based on the input context. Let P_{TFT} and P_{LSTM} denote the prediction tensors from the base models for a specific lead time range. The fusion process is defined as follows:

- **Stacking**: Predictions are concatenated to form a feature matrix Z: Z = Concat (P_{TFT}, P_{LSTM}) $\in R^{B \times 2L}$, where L is the number of lead time steps in the specific range.
- **Attention Context**: An attention mechanism computes context weights α based on the relevance of the base predictions: $\alpha = Softmax\ (W_{att} \cdot Z)$ $C = \alpha \cdot Z$.
- **Meta Regression**: A Multi-Layer Perceptron (MLP) maps the context to the final prediction $\hat{Y}$: $\hat{Y} = MLP\ (C)$. This MLP consists of a linear layer that reduces the dimension by half, Batch Normalisation, ReLU activation, Dropout, and a final projection layer.

3.1.4. Data Augmentation Parameters

To enhance model cross-domain performance and robustness against noise inherent in environmental sensors, a stochastic augmentation pipeline is integrated into the *FloodDataset*. The parameters are adaptively scaled based on the dataset size.

- **Gaussian Noise Injection**: Applied with probability $p = 0.3$. This simulates instrumentation errors and noise inherent in hydrological monitoring networks.

Standard deviation $\sigma_{noise} = 0.003$ (standard) or 0.005 (small datasets). These values are deliberately constrained to the order of millimetres in the normalised space to preserve the macroscopic hydrograph structure while perturbing exact values. The slightly elevated variance in smaller datasets acts as a stronger regulariser, preventing the model from memorising sparse observations and forcing it to learn broader temporal trends [28].

- **Magnitude Warping**: Distorts the magnitude of the signal via multiplication.

Warping factor drawn from $N\ (1, \sigma_{warp})$ where $\sigma_{warp} = 0.03$. Unlike global scaling, magnitude warping introduces localised, non-uniform distortions across timesteps. This

effect emulates the spatial heterogeneity of rainfall and variable catchment response rates that a single point gauge cannot capture. By introducing micro-variations in the rising and falling limbs, this forces the attention mechanisms and recurrent layers to rely on the overall temporal pattern rather than overfitting to isolated, precise spike magnitudes [28].

- **Scaling**: Global scaling of the input sequence.

Scale factor $s \sim u$ (0.995, 1.005). This emulates systematic sensor bias (e.g., a gauge consistently reading slightly high or low) or marginal shifts in overall catchment antecedent wetness. The tight bound ensures the scaled hydrograph remains physically plausible while encouraging the model to learn scale-invariant features of the flood dynamics [28].

- **Mixup Augmentation**: Linear interpolation between two samples (x_i, x_j).

Probability $p = 0.2$, $\alpha_{mixup} = 0.01$. The very low α value ensures that the synthetic sample is heavily dominated by a single primary hydrograph, with only a marginal blend of a secondary event. This generates physically plausible transitional states between distinct flood events, smoothing the decision boundaries of the meta-learner and significantly improving resilience against overfitting during few-shot adaptation tasks in ungauged basins [29].

3.1.5. Experimental Protocol

The experimental design ensures statistical rigour and reproducibility, adhering to best practices in time-series forecasting evaluation.

- **Data Preprocessing and Splitting**:

Imputation: Missing values are handled via a simplified GRU-D-inspired mechanism that utilises time-decay factors based on the interval since the last observation, combined with forward/backward filling.

Cross-Validation: A 5-Fold Cross-Validation strategy is employed to assess model reliability. The dataset is shuffled with a fixed random seed (42) before splitting.

Validation Holdout: Within each training fold, the data is further split into 70% for training and 30% for validation for hyperparameter tuning and early stopping.

- **Hyperparameter Optimisation:**

Framework: Optuna with the Tree-structured Parzen Estimator sampler.

Pruning: The *SuccessiveHalvingPruner* is utilised to terminate unpromising trials early, optimising computational resources.

Search Space:

Model Dimension (d_{model}): {128, 256}.

Number of Heads (*nhead*): {4, 8}.

Dropout Rate: Uniform search space [0.15, 0.25].

Learning Rate: Log-uniform search space [10^{-4}, 10^{-3}].

- **Training Configuration:**

Loss Function: A custom *ContextAwarePeakLoss* is used, combining Base MSE, Asymmetric Peak Loss (penalising underestimation 2× more than overestimation), and Shape Similarity Loss.

Regularisation: L_1 and L_2 penalties are applied ($\lambda \approx 10^{-5}$ to 10^{-4}).

Optimisation: *AdamW* optimiser with *CosineAnnealing* learning rate scheduling. Mixed precision training (*autocast*) is enabled on CUDA devices.

Early Stopping: Monitoring validation loss with a patience of five epochs.

- **Evaluation Metrics:**

The model is evaluated on standard metrics such as MSE, RMSE, MAE, R^2, and NSE.

3.2. Data Preprocessing

To ensure data quality and consistency across catchments and lead times, a comprehensive data preprocessing strategy is applied prior to model training and evaluation. The preprocessing pipeline addresses missing data, scale disparities, temporal structure, and non-stationarity through a combination of imputation, normalisation, and multi-stage transformation techniques designed specifically for short-term flood forecasting [30].

Missing data is handled via GRU-D (Gated Recurrent Unit with Decay) based imputation, designed for irregular multivariate time series. GRU-D parameterises temporal decay to modulate hidden states and inputs as a function of time gaps, thereby adaptively weighting historical observations [31]. The approach embeds masking and time-interval encoding within the recurrent dynamics, preserving temporal structure without imposing an increase in interpolation bias [31]. This is particularly suitable for hydrological datasets with frequent sensor and transmission gaps, yielding statistically consistent imputations and reduced bias in rainfall and water-level records.

Following imputation, all continuous variables are normalised using Z-score standardisation, which rescales each feature to have a mean of 0 and unit variance. This normalisation strategy ensures predictive consistency during training and prevents features with larger magnitudes from dominating the learning process [32]. Z-score standardisation is especially appropriate for deep learning models, as it accelerates convergence and improves cross-domain performance when training across multiple basins with differing hydrological scales and rainfall intensities [33].

Beyond imputation and normalisation, a series of transformation techniques is applied at three distinct stages: preprocessing, runtime data augmentation, and post-processing, to enhance temporal representation, prevent overfitting, and improve predictive reliability [34]. During preprocessing, rainfall measurements are converted from millimetres to metres to maintain unit consistency. The hydrometeorological observations used in this study were obtained at sub-hourly temporal resolution. Therefore, the modelling framework was developed from high-resolution time series data rather than from daily aggregated observations. To capture diurnal patterns, the hour-of-day feature is encoded using sinusoidal functions, allowing the model to represent temporally cyclicality so that late-night and early-morning hours remain close in feature space. Temporal dynamics are further enriched through first-order differencing, which captures rates of change, and through rolling statistics, including rolling means and cumulative sums computed over 6 h and 24 h windows to reflect short-term persistence and longer-term catchment memory. In addition, an interaction term between rainfall and water level is introduced to explicitly represent their coupled influence on flood evolution.

To improve model stability and mitigate overfitting, data augmentation transformations are applied dynamically during training [29]. As parameterised in Section 3.1.4, these transformations are specifically tailored to mirror operational hydrological uncertainties. The injection of Gaussian noise forces the network to generalise beyond precise gauge readings, simulating telemetry errors. Magnitude warping introduces local temporal distortions that reflect variable runoff response times and the spatial heterogeneity of precipitation, ensuring the attention weights are distributed across the sequence rather than localised to single spikes. Random global scaling emulates systematic sensor bias and intensity variations in catchment response [35]. Finally, a low-alpha mixup strategy is employed, where pairs of training samples are linearly combined to synthesise plausible intermediate hydrograph states. This encourages the model to learn smoother, more continuous decision boundaries, which is particularly critical for improving few-shot transferability under limited data conditions.

Finally, post-processing transformations are applied to refine model outputs and enhance operational reliability [36]. Dynamic bias correction adjusts predictions based on recently observed systematic errors, defined as the residuals between model predictions and corresponding ground-truth observations over a rolling time window [37]. Only past (observed) errors are utilised, ensuring that no future information is introduced during correction. This mechanism enables the model to compensate for short-term prediction drift and non-stationary dynamics by adaptively updating the bias term in response to evolving hydrological conditions. Temporal smoothing using a centred moving average filter reduces high-frequency noise in forecasted water levels, producing more stable predictions suitable for early warning applications [38]. To ensure accurate representation of extreme events, a peak enhancement step selectively amplifies predicted local maxima, improving the detection and magnitude estimation of flood peaks [39]. Together, these preprocessing and transformation strategies ensure that the forecasting framework remains resilient to data imperfections, captures critical hydrological dynamics, and delivers reliable short-term flood predictions in multi-horizon forecasting.

3.3. Input-Output Configuration

The input-output configuration of the forecasting framework is designed to capture effectively short-term hydrological dynamics while maintaining flexibility across varying forecast horizons. Both rainfall and water level are treated as continuous time-series variables, reflecting their naturally evolving behaviour and preserving temporal dependencies throughout the modelling process. The selection of rainfall and lagged water level as input variables was guided by both hydrological relevance and operational applicability. Rainfall represents the primary external forcing responsible for runoff generation and flood initiation, while water level reflects the current hydrological state of the catchment and implicitly captures the cumulative effects of antecedent rainfall, catchment storage conditions, routing processes, and other basin-scale influences [40,41]. Together, these variables provide the most direct and immediately available information governing short-term flood evolution. Although additional hydrometeorological variables such as temperature, soil moisture, evapotranspiration, humidity, wind speed, and atmospheric pressure may influence hydrological processes, their impact on flood dynamics within short forecasting horizons (1–48 h) is generally indirect. It often occurs through their influence on rainfall-runoff interactions [3,13,42]. Furthermore, such variables are not consistently available across many operational monitoring networks and data-scarce catchments [13,43]. Therefore, the proposed framework prioritises rainfall and lagged water level to maintain model transferability, reduce input dimensionality, minimise data dependency, and ensure practical applicability for real-time flood forecasting across diverse basin conditions.

Lag selection plays a critical role in short-term flood forecasting, as recent hydrometeorological conditions strongly influence near-future river response. In this study, lagged sequences of rainfall and water level are used as model inputs to capture catchment memory, flow persistence, and delayed runoff processes. Rainfall lags provide information on antecedent precipitation and storm evolution, which are essential for identifying rising limbs and imminent peak formation. Water level lags, on the other hand, represent the internal state of the river system, allowing the model to learn inertia effects, recession behaviour, and feedback from upstream routing. By jointly incorporating lagged rainfall and water level inputs, the framework captures both external forcing and internal system dynamics that govern short-term flood evolution.

The forecasting task focuses exclusively on short-term lead times, which are critical for operational flood early warning and emergency response. Four forecast horizons are considered: 1–6 h, 6–12 h, 12–24 h, and 24–48 h. These lead time windows are chosen to

represent progressively increasing levels of forecasting difficulty, ranging from near-real-time prediction driven by strong temporal persistence to extended short-term forecasts that require effective learning of delayed hydrological responses. Shorter lead times primarily rely on recent water level trends, while longer lead times increasingly depend on rainfall-driven dynamics and the model's ability to retain and propagate temporal information.

The problem is formulated as a supervised sequence-to-sequence learning task, where lagged rainfall and water level sequences form input features, and future water levels over selected forecast windows are the outputs. This formulation enables unified learning of multi-step temporal dependencies and supports stable forecasting across all short-term horizons. Maintaining a consistent input-output configuration across lead times also facilitates fair performance comparison and consistent assessment of model generalisation under varying forecast lengths.

3.4. Meta-Learning Layer: First-Order MAML

Flood forecasting models are traditionally trained in a basin-specific manner, assuming that sufficient historical data are available and that hydrological conditions remain relatively stationary [44]. In practice, however, river basins exhibit substantial variability in geomorphology, climate forcing, and response time, while many catchments suffer from limited or incomplete data records [45]. Models optimised for one basin often perform poorly when transferred to another, and retraining from scratch for each new catchment is both computationally expensive and impractical for real-time or large-scale deployment [46]. We have included MAML in the proposed model. The architectural design of GHDFloodNet and its integration with First-Order MAML are structured to mitigate the identified limitations through specific algorithmic mechanisms. The fundamental bottleneck in cross-basin transfer is that standard backpropagation converges on parameter spaces that are highly localised to the physics and noise distribution of a single catchment. The *EnhancedMAML* class explicitly avoids learning static, basin-specific weights. Instead, the meta_update function computes gradients across a distributed *task_batch* (representing heterogeneous basins). By executing *self.inner_update*, which clones the global model and performs a localised forward pass and backward pass (*loss.backward()*) using a small subset of basin-specific data, the algorithm calculates second-order gradients (or first-order approximations) to optimise the *initialisation point (θ)* of the network rather than the final converged weights. Because the meta-optimisation objective seeks an initialisation where a minimal number of gradient steps yields a low loss on a new basin, the resulting parameters represent basin-invariant hydrological priors. When deployed to an ungauged or data-scarce environment, the model does not require the large volumes of data typically needed to escape local minima; it merely requires a few fast adaptation steps to specialise the shared initial representation to the local catchment dynamics.

MAML Formulation and Optimisation

To directly address Limitation 1 (poor cross-basin performance) outlined in Section 2, GHDFloodNet operationalises First-Order MAML through a strict bi-level optimisation architecture. Within this formulation, each distinct river catchment is structured as an independent meta-learning task, replacing conventional single-basin parameter convergence with a task-distribution learning paradigm.

During the inner-loop adaptation, the model is fine-tuned on task-specific data drawn from an individual catchment [47]. Starting from the shared meta-initialisation, a small number of gradient descent steps are performed using recent rainfall and water level observations. This allows the model to quickly specialise in local hydrological characteristics, such as response time, runoff generation mechanisms, and flow regulation effects, without

requiring extensive retraining. The inner-loop updates are intentionally lightweight to ensure fast adaptation, which is critical for operational forecasting scenarios [48]. Specifically, the meta-learning procedure follows a strictly nested optimisation framework to prevent information leakage between adaptation and evaluation stages [49]. In this setup, the inner-loop updates (task-specific adaptation) are performed exclusively on support sets constructed from historical observations within each training task. Importantly, temporal ordering is strictly preserved, ensuring that future information is never introduced into the adaptation phase. Furthermore, during evaluation, the model is adapted using only past observations available up to the forecast issuance time, thereby mimicking real-world deployment conditions [48]. This nested training and evaluation protocol ensures that no data leakage occurs between training, validation, and testing phases and aligns with best practices for meta-learning in time-series forecasting.

The outer-loop meta-optimisation operates across a distribution of tasks representing multiple catchments and hydrological regimes. After inner-loop adaptation, the performance of the adapted model is evaluated on separate validation data from the same task [47]. Gradients from these evaluations are then aggregated to update the shared meta-parameters, with the objective of improving the model's ability to adapt efficiently across all tasks. Through repeated exposure to diverse catchments during meta-training, the model learns parameter initialisations that facilitate cross-catchment transferability, as demonstrated by consistent multi-metric performance across the Richmond and Tweed datasets, maintaining nearly invariant error distributions (MSE $\approx$ 0.53–0.55, RMSE $\approx$ 0.72–0.74, MAE $\approx$ 0.35–0.36) and stable goodness-of-fit (R^2/NSE $\approx$ 0.44–0.47) across all lead times (1–48 h), in contrast to benchmark models that exhibit pronounced error escalation and metric degradation with increasing forecast horizons, thereby capturing recurrent hydrological response behaviours observable across heterogeneous basins, while remaining flexible enough to accommodate local variability. Although the Richmond catchment contains a longer historical record and consequently yields a larger pool of potential meta-learning tasks than the Tweed catchment, task imbalance during training is mitigated through stratified episodic sampling. Specifically, each meta-training iteration draws an equal number of adaptation tasks from both catchments, ensuring that the outer-loop meta-update is not disproportionately influenced by the data-rich basin.

Furthermore, regularisation and stochastic augmentation are applied to enhance the diversity of tasks generated from the shorter Tweed record. Nevertheless, the present study acknowledges that incorporating a larger number of catchments with comparable data lengths would provide a more representative task distribution and may further improve cross-basin generalisation. Details of the meta-learning layer, the predictive core (base learners), are presented in Figure 2.

3.5. Temporal Representation Layer: TFT

3.5.1. Temporal Dependency Modelling

To capture the complex, nonlinear temporal dynamics in flood time series, the proposed framework adopts a dual-branch temporal representation architecture comprising (i) a stacked BiLSTM encoder and (ii) a lightweight TFT encoder. The two branches operate in parallel on the same hourly lookback window and are trained to produce complementary multi-horizon predictions. Their outputs are subsequently fused by an attention-based meta-ensemble that performs learned nonlinear combination separately for each lead time range (1–6 h, 6–12 h, 12–24 h, and 24–48 h).

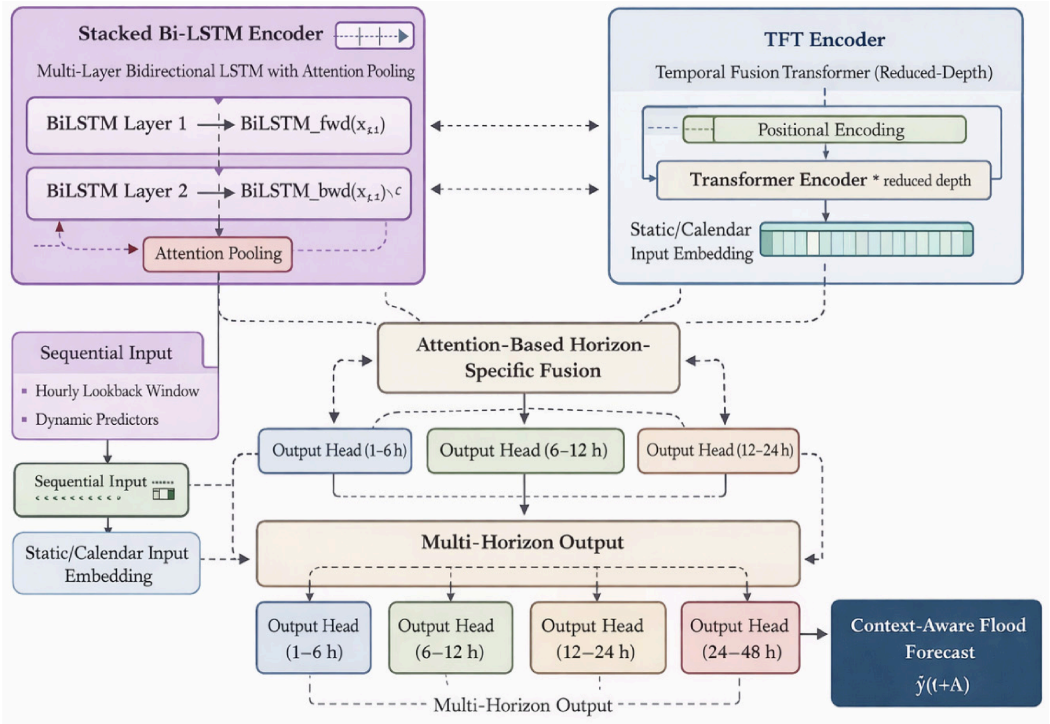


Figure 2. Architecture of BiLSTM and TFT encoder in GHDFloodNet.

- **Multi-Layer BiLSTM**

The recurrent branch is implemented in the *LSTMModel* module as a stacked BiLSTM with dropout regularisation. The hidden dimensionality, number of stacked layers, and dropout rate are treated as tunable hyperparameters (set via the selected configuration *best_params* during training), allowing the model capacity to adapt to the dataset size and complexity. Bidirectional processing enables the encoder to exploit contextual information from both earlier and later positions within the input window. After recurrent encoding, a feed-forward attention pooling mechanism is applied to the full sequence of LSTM hidden states. Specifically, attention weights are computed using a small multilayer perceptron with Tanh activation followed by Softmax normalisation across timesteps, and the final context vector is obtained as the weighted sum of hidden states. This attention-derived context vector is then passed to horizon-specific output heads (MLPs with batch normalisation and dropout) to generate multi-step predictions for each forecast range. A detailed workflow is presented in the next subsection.

- **TFT encoder**

The second branch, implemented in *TemporalFusionTransformer*, models temporal dependencies using self-attention to capture global interactions and delayed hydrological responses across the lookback window. Input features at each timestep are projected into a latent space of dimension d_{model} (tuned during optimisation), and static/calendar descriptors are projected to the same dimension and additively fused with the temporal embeddings by broadcasting across timesteps. Because self-attention is permutation-invariant, sinusoidal positional encoding is added to preserve temporal order. The resulting sequence is processed by a reduced-depth *NN.TransformerEncoder* with (i) pre-normalisation (*norm_first = True*) for improved training stability, (ii) GELU activation in the feed-forward sublayers, and (iii) an efficiency-oriented feed-forward width (*dim_feedforward = d_model*). The encoder outputs are aggregated via the same attention pooling mechanism used in the LSTM branch to obtain a compact context representation. This representation is then

passed through a sigmoid-activated gating block (Linear → Sigmoid → Dropout) before being mapped to lead time-specific predictions through horizon-specific output heads.

For each lead time range, predictions from the BiLSTM and Transformer branches are concatenated and fused using an attention-based meta-learner in the *MetaEnsemble* module. The fusion stage learns nonlinear combinations of the base-model outputs, producing final horizon-specific multi-step forecasts.

3.5.2. BiLSTM Architecture in GHDFloodNet

In the ensemble fusion mechanism, the BiLSTM receives sequences of predictions generated by the base learners and learns nonlinear relationships among them over time [50]. Rather than assigning static weights, the BiLSTM dynamically adjusts the contribution of each base model based on temporal context and learned dependencies [50]. This results in context-aware fusion, where models that perform better under certain conditions, such as extreme rainfall or stable flow periods, are automatically prioritised. By integrating forward-backward temporal dependency capture with adaptive ensemble weighting, the meta-ensemble layer significantly improves forecast reliability and consistency across all short-term lead times [51]. Figure 3 shows the structural diagram of BiLSTM.

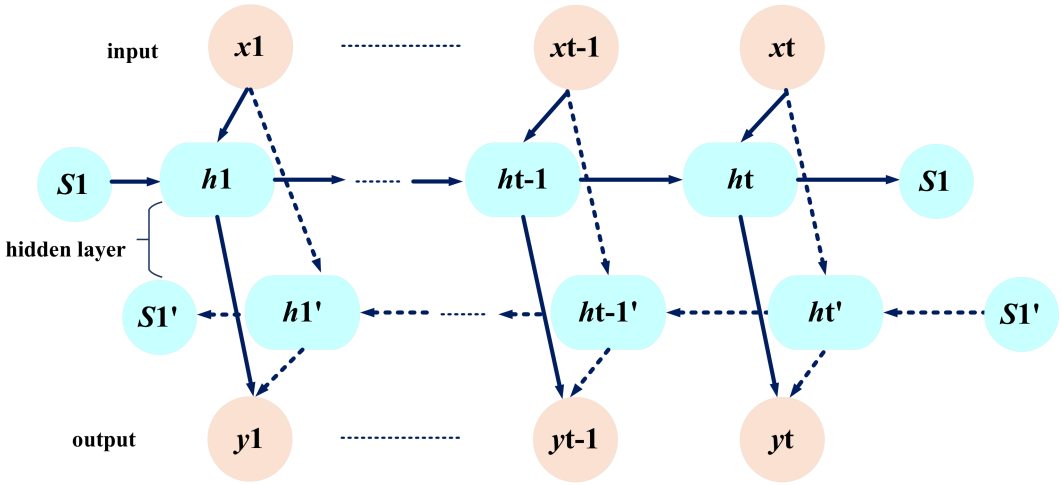


Figure 3. BiLSTM structure.

The First BiLSTM Layer: This is the entry point for the data.

Input: It takes the raw time-series sequences (e.g., rainfall, water level, lags) defined by x in the forward method.

Operation:

Forward Pass: Reads the sequence from time t_1 to t_{end} (past → future). It learns how previous water levels lead up to the current moment.

Backward Pass: Reads the sequence from time t_{end} to t_1 (future → past). It learns the context of what is coming based on future data points in the input window.

Output: It produces a set of hidden states (features) that capture immediate temporal patterns. It passes this entire processed sequence up to the second layer.

The Second BiLSTM Layer: This layer does not look at the raw data; it looks at the output of the first layer.

Input: It receives the sequence of hidden states from the 1st BiLSTM Layer.

Operation:

Forward Pass: It scans the "summarised" features from Layer 1 from start to end.

Backward Pass: It scans the "summarised" features from Layer 1 from end to start.

Purpose: While the 1st layer detects simple patterns (e.g., "rain is increasing"), the 2nd layer combines these to detect complex, higher-level relationships (e.g., "heavy rain + rising water level + soil saturation = high flood risk"). It abstracts the data further.

How they connect:

Data Flow: The input data flows through Layer 1. Layer 1 processes it and passes its internal state (hidden and cell states) to Layer 2.

Hierarchical Learning: Layer 2 operates on Layer 1′s representation. The model has a 3rd layer; it would operate on Layer 2′s representation, and so on.

Final Output: After passing through all layers, the final output tensor out contains the processed information from the topmost layer.

Layer 1 BiLSTM reads x $\rightarrow$ produces hidden states h_1

Layer 2 BiLSTM reads h_1 $\rightarrow$ produces hidden states h_2

Layer 3 BiLSTM reads h_2 $\rightarrow$ produces final hidden states out

3.5.3. Attention Mechanisms for Hydrological Time Series

The framework incorporates attention to improve event sensitivity and temporal summarisation in flood forecasting. Given hidden states $\{h_t\}_{t=1}^{L}$, the module computes timestep relevance scores using an MLP (Linear $\rightarrow$ Tanh $\rightarrow$ Linear), normalises them with a Softmax over time, and forms a context representation $c = \sum_t \alpha_t h_t$. This mechanism is applied after both the BiLSTM encoder and the Transformer encoder, potentially enabling the model to emphasise hydrologically informative intervals. However, the interpretability and physical meaning of the attention weights (e.g., alignment with rainfall peaks or rapid stage rise) have not been empirically validated through analysis or visualisation of the attention distributions in this study and represent a direction for future work.

Although the same attention pooling formulation is employed across both branches, it serves complementary, non-redundant roles. In the BiLSTM branch, it selectively integrates sequential memory representations, mitigating information decay over extended temporal horizons. In contrast, in the Transformer branch, where self-attention already captures global dependencies, the pooling layer primarily acts as a dimensionality reduction and summarisation operator, compressing multi-head representations into a compact context for downstream fusion.

In the Transformer pathway, sinusoidal positional encoding is introduced to preserve temporal order prior to self-attention, and the pooled context is further processed through a sigmoid-activated transformation before horizon-specific prediction heads. At the fusion stage, base-model outputs are concatenated and combined via a nonlinear meta-learner; while an attention block is included at this level, the final predictive mapping is predominantly governed by the subsequent MLP, thereby minimising redundancy in attention utilisation across the architecture.

3.6. Stacking-Based Meta-Ensemble Learning Layer

To synthesise complementary predictive behaviours from heterogeneous temporal encoders, the proposed framework adopts a stacked generalisation (neural stacking) strategy implemented in the *MetaEnsemble* module. At the level-0 layer, two base learners, a BiLSTM-based sequence encoder (*LSTMModel*) and a Transformer-based temporal encoder (*TemporalFusionTransformer*, simplified encoder-style variant), are trained to produce multi-step forecasts for predefined lead time ranges $\tau \in \{1–6\text{ h}, 6–12\text{ h}, 12–24\text{ h}, 24–48\text{ h}\}$. For a given input (x, s), each base model M_k outputs a vector $\hat{y}_\tau^{(k)} \in \mathbb{R}^{L_\tau}$, where L_τ is the number of forecast steps in range τ. Rather than using a single global combiner, the architecture employs range-specific meta-learners to reflect the fact that short-lead and long-lead forecasts typically exhibit different error structures. An attention-shaped

block (*meta_attention[range_name]*) is included before the meta-learner for architectural consistency with the temporal attention pooling used in the base encoders. In the current implementation, this block is applied to the concatenated prediction vector as a length-one token; consequently, it does not perform substantive reweighting across multiple elements, and the primary fusion capacity is delivered by the subsequent MLP combiner.

3.7. AutoML and Bayesian Hyperparameter Optimisation

3.7.1. Hyperparameter Search Space

The hyperparameter search space in GHDFloodNet spans multiple architectural and optimisation dimensions. This includes network depth, such as the number of layers in recurrent, transformer, and ensemble components, which controls model capacity and representation power [52]. Learning rates and related optimisation parameters are explored to ensure stable convergence and effective adaptation during both meta-learning and fine-tuning phases [52]. For transformer-based components, key architectural parameters, such as the number of attention heads and hidden dimensions, are also included, allowing the model to balance expressiveness with computational efficiency [52]. By jointly optimising these parameters, the framework avoids over- or under-parameterisation and ensures consistent performance across lead times and catchments.

3.7.2. Bayesian Optimisation Framework

To efficiently navigate the hyperparameter search space, GHDFloodNet employs a Bayesian optimisation framework within an AutoML setting. The optimisation process is guided by an objective function defined in terms of forecasting accuracy on validation data, using performance metrics relevant to short-term flood prediction. Compared to exhaustive grid search or uninformed random search, Bayesian optimisation is substantially more efficient, requiring fewer model evaluations to converge toward optimal or near-optimal solutions [53]. This efficiency is particularly important given the computational cost of training deep architectures and meta-learning models [53].

3.8. Ablation Analysis and Justification of Core Architectural Components

An ablation-oriented perspective is adopted to justify each architectural choice in relation to the challenges of short-term flood forecasting, including cross-basin performance, temporal dependency modelling, robustness under limited data, and operational consistency.

- **Effect of removing Meta-learning layer:** MAML is adopted as it introduces a series of optimisation refinements, including improved gradient estimation, better inner-loop learning-rate handling, and enhanced stability during meta-training [54]. Ablation experiments showed that replacing First-Order MAML with standard MAML or conventional transfer learning led to slower convergence and reduced consistency across lead times, justifying the selection of First-Order MAML.
- **Effect of replacing the TFT module:** TFT provides improved interpretability through its variable selection networks and attention weights, which are particularly valuable in hydrological applications where understanding model behaviour is important for trust and operational use [55,56]. Ablation results indicated that replacing TFT with standard recurrent networks reduced performance, especially for longer lead times, supporting the choice of TFT as the core temporal representation layer.
- **Effect of replacing the BiLSTM:** Alternative fusion approaches, such as simple averaging, weighted linear combinations, were found to be less effective in capturing temporal coherence across lead times [57,58]. The BiLSTM consistently improved peak prediction and reduced error variance, justifying its inclusion [59].

- **Effect of disabling Bayesian hyperparameter optimisation:** Ablation analysis demonstrated that disabling Bayesian HPO led to suboptimal and inconsistent performance across basins, confirming its role in ensuring stable and near-optimal model configurations.
- **Effect of removing Runtime Data Augmentation:** Ablation results showed that removing this layer degraded transferability performance, particularly during high-flow events, highlighting its importance for reliable flood forecasting.

3.9. Study Area and Data Description

This study focuses on two flood-prone river systems in the Northern Rivers region of New South Wales, Australia: the Richmond and Tweed River catchments. These catchments were selected to represent contrasting hydrological regimes, characterised by differences in runoff response, peak flow behaviour, and rainfall-runoff dynamics.

The Richmond River catchment is represented by the hydrometric station Richmond River at Kyogle (Station No. 203900). The analysed sub-catchment (~1354 km^2) drains the Border Ranges and exhibits a rapid hydrological response due to steep headwaters and dense tributary networks. Key upstream tributaries include Grady's Creek, Roseberry Creek, Findon Creek, and Lynches Creek, which contribute to fast runoff generation during high-intensity rainfall events [60]. Flow regulation is present through structures such as Kyogle Weir, influencing local hydrodynamics. The transition from upland catchment to downstream floodplain introduces both rapid flow peaks and extended inundation behaviour, making it suitable for evaluating short- to medium-range flood forecasting performance.

The Tweed River catchment is represented by the station Tweed River at Uki (Station No. 201900). The catchment covers approximately 1055 km^2 and is characterised by steep terrain, short response times, and high rainfall variability under a humid subtropical climate [61]. Major tributaries, including the Oxley and Rous rivers, amplify peak discharge during storm events. The basin's volcanic geomorphology contributes to heterogeneous infiltration and runoff processes, resulting in highly responsive and flash-flood-prone behaviour. Frequent exposure to extreme weather systems, such as east coast lows and ex-tropical cyclones, further intensifies flood risk.

The contrasting hydrological characteristics of the Richmond and Tweed catchments provide a controlled yet diverse experimental setting, enabling rigorous evaluation of model transferability, multi-horizon forecasting performance, and predictive consistency under varying hydrological conditions. The Richmond and Tweed River catchments are presented in Figure 4a,b.

3.10. Data Sources

This study utilises two primary categories of observational data, namely meteorological data (rainfall) and hydrological data (water level), as predictor variables for short-term flood forecasting. These variables were selected because they represent the dominant hydrological forcing and response mechanisms controlling flood generation in rainfall-driven river systems. The datasets were obtained from the Bureau of Meteorology (BoM), Australia, and associated official water-monitoring archives for the selected stations [62,63]. To improve data transparency and facilitate reproducibility, the key characteristics of the datasets used in this study are summarised in Table 1.

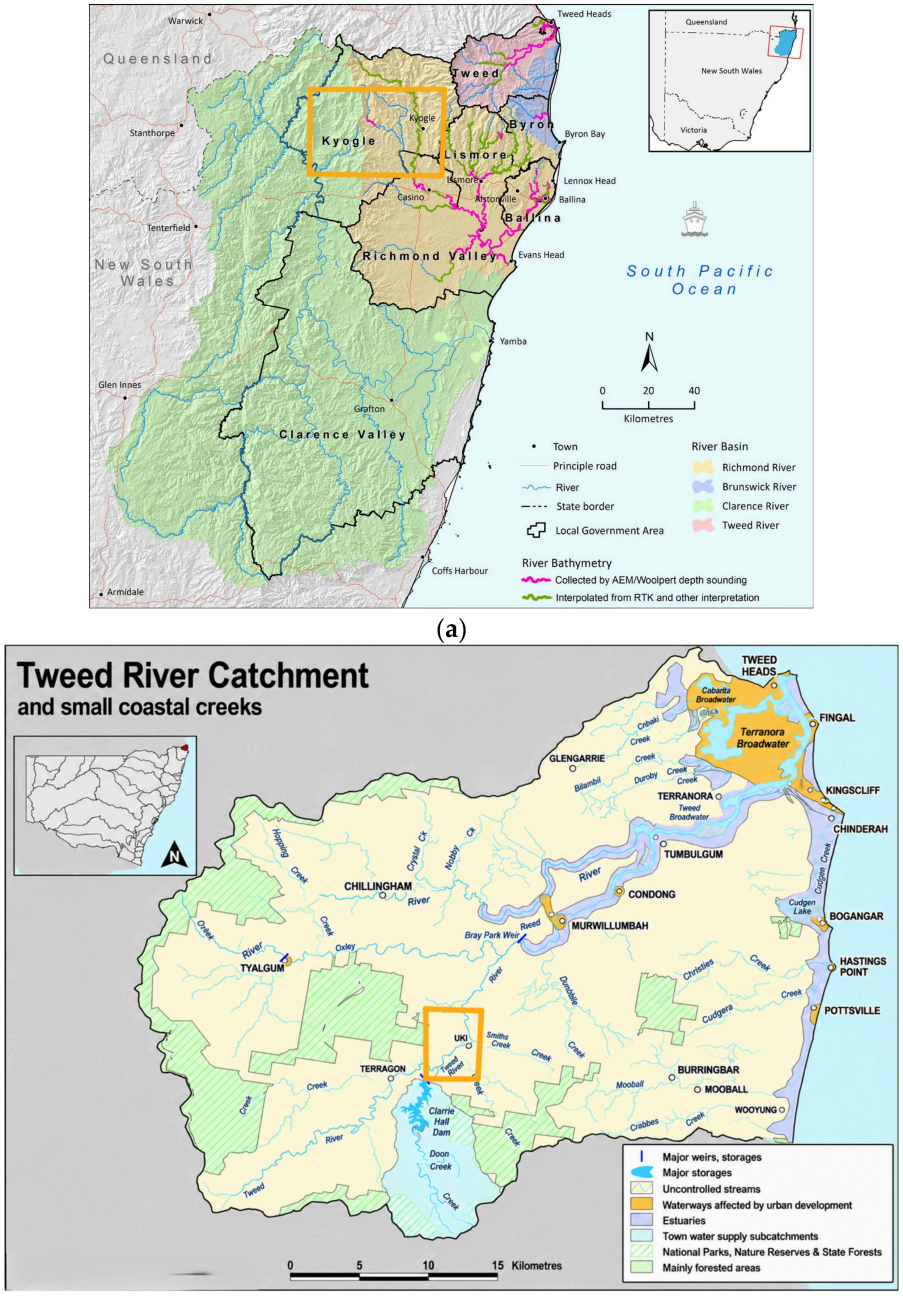


Figure 4. (**a**) Richmond (Kyogle) and (**b**) Tweed River (Uki) catchments.

The water-level observations were recorded at a regular 15 min interval, providing a consistent representation of short-term river-response dynamics. Rainfall observations were available at finer event-based timestamps and were recorded at irregular minute- and second-level intervals depending on rainfall occurrence and sensor logging behaviour. Such high-resolution observations are particularly valuable for flood forecasting because they preserve rapid hydrological transitions that are often obscured by temporally aggregated datasets.

The rainfall records were not uniform across the two study catchments. While rainfall observations were available for approximately ten years in the Richmond catchment, only three years of rainfall records were available for the Tweed catchment. The unequal duration of rainfall records between the Richmond and Tweed catchments has important implications for model training and meta-learning. Specifically, the Richmond catchment

contributes a substantially larger number of training sequences than the Tweed catchment, potentially introducing an imbalance in the distribution of tasks observed during meta-training. Such an imbalance may increase the representation of hydrological patterns characteristic of the Richmond basin and could influence the learned meta-initialisation. However, the proposed GHDFloodNet framework mitigates this effect through its task-based optimisation strategy, where adaptation is performed on individual task batches rather than on the cumulative volume of observations alone.

Table 1. Summary of data sources and characteristics used for flood forecasting.

Catchment	Monitoring Station	Variable	Data Provider	Observation Period	Temporal Resolution	Data Availability
Richmond River	Kyogle	Water Level	BoM and official water-monitoring archives	Sep 2015–Sep 2025	15 min	~10 years
Richmond River	Kyogle	Rainfall	BoM	Sep 2015–Sep 2025	Event-based (minute/second level)	~10 years
Tweed River	Uki	Water Level	BoM and official water-monitoring archives	Sep 2015–Sep 2025	15 min	~10 years
Tweed River	Uki	Rainfall	BoM	Sep 2022–Sep 2025	Event-based (minute/second level)	~3 years

Furthermore, the use of data augmentation, regularisation, and cross-validation reduces the likelihood that the model becomes overly specialised to the longer record. Nevertheless, the difference in record length remains a potential limitation, as a more balanced multi-catchment dataset may further improve the diversity of hydrological conditions available during meta-training and enhance transferability to unseen environments. Future research will investigate the performance of GHDFloodNet using a larger collection of catchments with more uniform observational records. For completeness and transparency, representative examples of the raw observations of the input datasets are presented in Appendix A (Tables A1–A4).

Incorporating rainfall observations enables the forecasting models to capture the causal relationship between precipitation inputs and subsequent river-level responses. Similarly, historical water-level observations provide information regarding hydrological persistence, recession behaviour, and catchment memory, all of which are important for short-term flood prediction. Together, these variables allow the forecasting framework to represent both rapid runoff generation and delayed hydrological responses, thereby improving forecasting consistency across multiple lead times.

Prior to model development, all rainfall and water-level observations were temporally aligned and standardised to a common modelling interval. This preprocessing step was necessary because rainfall data were originally recorded at irregular event-based timestamps, whereas water-level observations followed a regular 15 min measurement schedule. The harmonised dataset ensured temporal consistency across predictor variables and enabled reliable model training and evaluation.

3.11. Evaluation Metrics

To comprehensively assess the predictive performance of the proposed flood forecasting framework, multiple complementary evaluation metrics are employed to capture both

error magnitude and the goodness-of-fit between predicted and observed water levels. Using a combination of absolute, squared, and normalised performance measures allows for a balanced evaluation across different forecast lead times and hydrological conditions. This research used five metrics (MSE, NSE, RMSE, MAE, and R^2), which are widely used to measure the performance in flood forecasting [64].

The Mean Squared Error (MSE) measures the average of the squared differences between predicted values and observed water levels as presented in Equation (1) [64].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(O_i - p_i)^2 \quad (1)$$

The Root Mean Square Error (RMSE) is derived by taking the square root of the MSE, which brings the error metric back to the original units of the predicted variable as presented in Equation (2) [64].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \quad (2)$$

The Mean Absolute Error (MAE) computes the average of the absolute differences between predicted and observed values, ignoring the direction of errors as presented in Equation (3) [64].

$$MAE = \frac{\Sigma_{i=1}^{n}|O_i - P_i|}{n} \quad (3)$$

The coefficient of determination (R^2) measures how well the model explains the variability in observed water level data as presented in Equation (4) [64].

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(O_i - P_i)^2}{\Sigma_{i=1}^{n}\left(O_i - \overline{O}_i\right)^2} \quad (4)$$

The Nash-Sutcliffe Efficiency (NSE) is a widely adopted performance metric in hydrology, offering a normalised assessment of model skill relative to a simple baseline model that predicts the mean of the observations as presented in Equation (5) [64].

$$NSE = \left(1 - \frac{\Sigma_{i=1}^{n}(O_i - P_i)^2}{\Sigma_{i=1}^{n}\left(O_i - \overline{O}_i\right)^2}\right) * 100 \quad (5)$$

In the context of the above five equations, O_i refers to the actual value, while P_i implies the predicted value. The symbol $\overline{O}_i$ stands for the mean of the observed values, and n denotes the total number of observations.

3.12. Hyperparameter Tuning

Hyperparameter Tuning plays a critical role in automating the optimisation of the proposed flood forecasting model. It acts as the AutoML component of the model, designed to find the most effective configuration. Specifically, it utilises the Optuna framework [65] to search a defined parameter space, testing combinations of model depth *(num_layers*), dimensionality (*d_model*), learning rate (*lr*), and regularisation weights (*l1_lambda, l2_lambda*). Furthermore, it tunes the specific weights of the custom *ContextAwarePeakLoss* to balance the trade-off between minimising general error and accurately detecting alarming flood peaks.

The role of this tuning process is to maximise predictive performance and computational efficiency. By employing a *SuccessiveHalvingPruner*, the system aggressively cuts off poorly performing trials early, saving significant training time. This ensures the final model uses only the most effective settings, preventing the guesswork often associated

with deep learning. Additionally, the tuning logic acts as a safeguard for data management; if the dataset is too small to support a validation split, it intelligently defaults to established parameters to ensure the model has enough data to train effectively. By systematically tuning these hyperparameters, the model achieves a balanced trade-off between complexity and stability, resulting in more reliable and consistent flood forecasts.

The impact of hyperparameter tuning becomes more pronounced when forecasting across different lead times [66]. Recent observations and local temporal continuity typically dominate short lead-time predictions. As lead time increases, suboptimal configurations tend to amplify prediction uncertainty and error propagation, making careful tuning essential to maintain forecast reliability [67]. By assessing model accuracy and consistency across multiple lead times, lead time analysis reveals whether the tuned hyperparameters support scalable temporal generalisation or only perform well within narrow horizons. The summary of hyperparameters is presented in Table 2.

Table 2. Lists of hyperparameters.

Hyperparameter Types	Hyperparameter	Value
Optuna Hyperparameter Search Space (These parameters are tuned during the optimisation process)	Learning Rate (lr)	1×10^{-4}–1×10^{-3}
	Model Dimension (d_model)	[128, 256]
	Number of Heads (nhead)	[4, 8]
	Number of Layers (num_layers)	2 to 3
	Dropout Rate (dropout)	0.15–0.25
	Batch Size	[16, 32]
	L1 Regularisation (l1_lambda)	1×10^{-6}–1×10^{-4}
	L2 Regularisation (l2_lambda)	1×10^{-6}–1×10^{-4}
	Base Loss Weight	0.6–0.8
	Peak Loss Weight	0.2–0.3
	Shape Loss Weight	0.1–0.2
Fixed Model & Training Configuration (These parameters are defined as constants in the script)	Input Sequence Length	24
	Forecast Horizons (Lead Times)	1–6 h, 6–12 h, 12–24 h, 24–48 h
	Max Lead Time	48
	Important Lags	[1, 6, 24]
	Rolling Windows	[6, 24]
	Feature Selection Count (k)	15
	MAML Inner Steps	2
	Training Epochs (Main)	20
	Optuna Trials	10
	Optuna Timeout	300 s
	Optimiser	AdamW
	Scheduler	OneCycleLR
	Early Stopping Patience	5 (Main)/2 (Optuna)
	Gradient Clipping	1.0

Table 2. *Cont.*

Hyperparameter Types	Hyperparameter	Value
Default Fallback Hyperparameters (If the dataset is too small for optimisation (len(sequences) < 20), these default values are used)	d_model	256
	nhead	8
	num_layers	2
	dropout	0.2
	lr	0.001
	batch_size	16
	l1_lambda	0.0001
	l2_lambda	0.0001
	base_loss_weight	0.7
	peak_loss_weight	0.3
	shape_loss_weight	0.2

3.13. Experimental Setup

The experimental framework begins with the ingestion of rainfall and water level time-series data from CSV files. The preprocessing pipeline involves several critical steps to ensure data quality and feature richness.

- **Parsing and Cleaning**: A custom *robust_csv_load* function handles file inconsistencies by detecting header rows, skipping comments, and managing malformed lines. Timestamp and value columns are automatically identified and renamed.
- **Unit Conversion**: Rainfall values are converted from millimetres to metres to maintain consistency with water level units.
- **Resampling and Imputation**: Data is resampled to an hourly frequency. Missing values are addressed using a simplified implementation of GRU-D, which utilises forward/backward filling and exponential decay based on the time interval since the last observation.
- **Feature Engineering**:
  - i. Time-based Features: Hour, Day of Week, and Month are extracted and encoded cyclically using Sine and Cosine transformations.
  - ii. Lag Features: Significant lags (1, 6, 24 h) are created for both rainfall and water level.
  - iii. Rolling Statistics: Rolling means (windows of 6 and 24 h) and accumulated rainfall are calculated.
  - iv. Dynamic Features: Rate of change (diff) and interaction terms (Rainfall × Water Level) are generated.
- **Feature Selection**: To reduce dimensionality and computational load, SelectKBest with the *f_regression* scoring function selects the top k = 15 features (or fewer if data is limited).
- **Normalisation**: All numerical features are normalised using Z-Score standardisation.
- **Sequence Construction and Datasets:** The data is organised into supervised learning samples using a sliding window approach, such as Sequence Length, Lead Times and Augmentation.
- **Hardware/Software:** The framework is implemented in PyTorch (Version 2.7.1+cu118). Details of hardware and software lists are presented in Table 3.

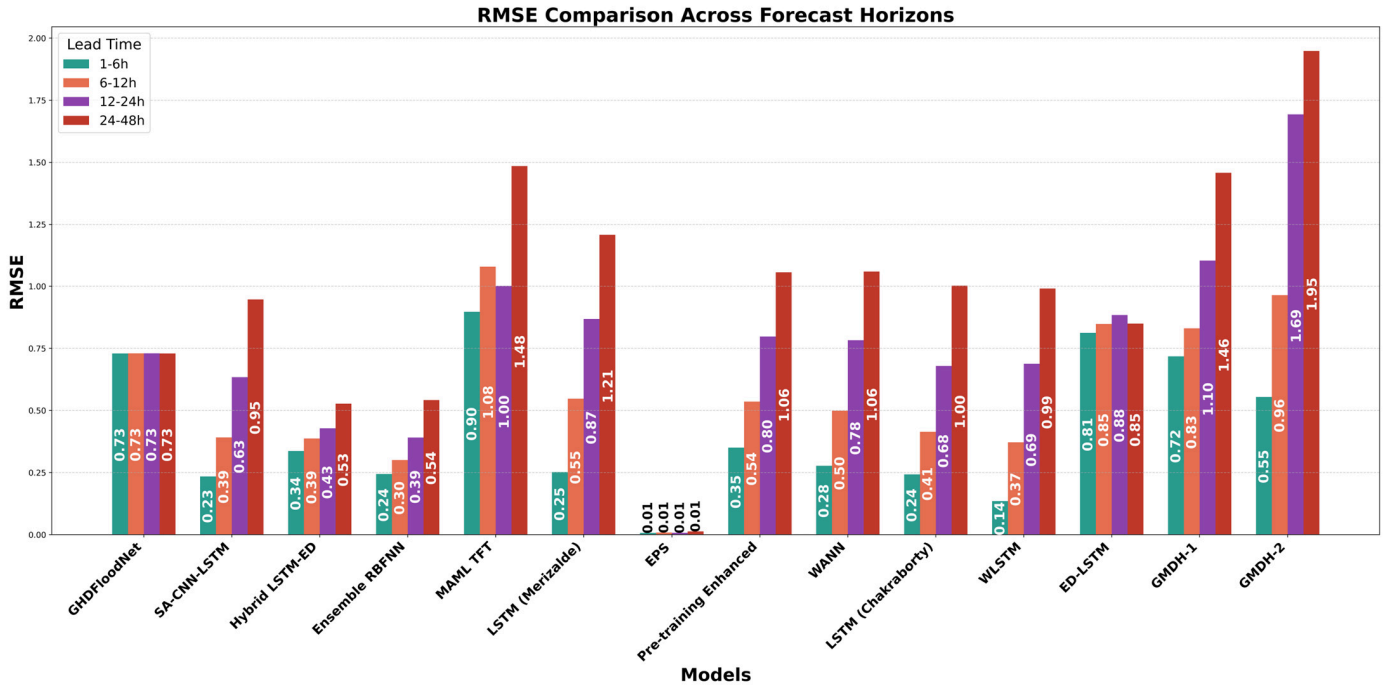
Table 3. Component and specification of the proposed model.

Component	Specification
Deep learning framework	PyTorch (version: 2.7.1+cu118)
Hyperparameter optimisation	Optuna (version: 4.5.0)
Feature selection	scikit-learn (*SelectKBest*, *f_regression*) (version: 1.6.1)
Normalisation	scikit-learn (*StandardScaler*) (version: 1.6.1)
Device	CUDA GPU (if available), otherwise CPU (CUDA version: 11.8, GPU: NVIDIA GeForce RTX 2050)
Reproducibility	Random seeds fixed at 42 (*torch*, *numpy*, *random*); CuDNN deterministic mode enabled; CuDNN benchmark disabled.
Mixed precision	PyTorch AMP (*autocast* + *GradScaler*) on CUDA

4. Results

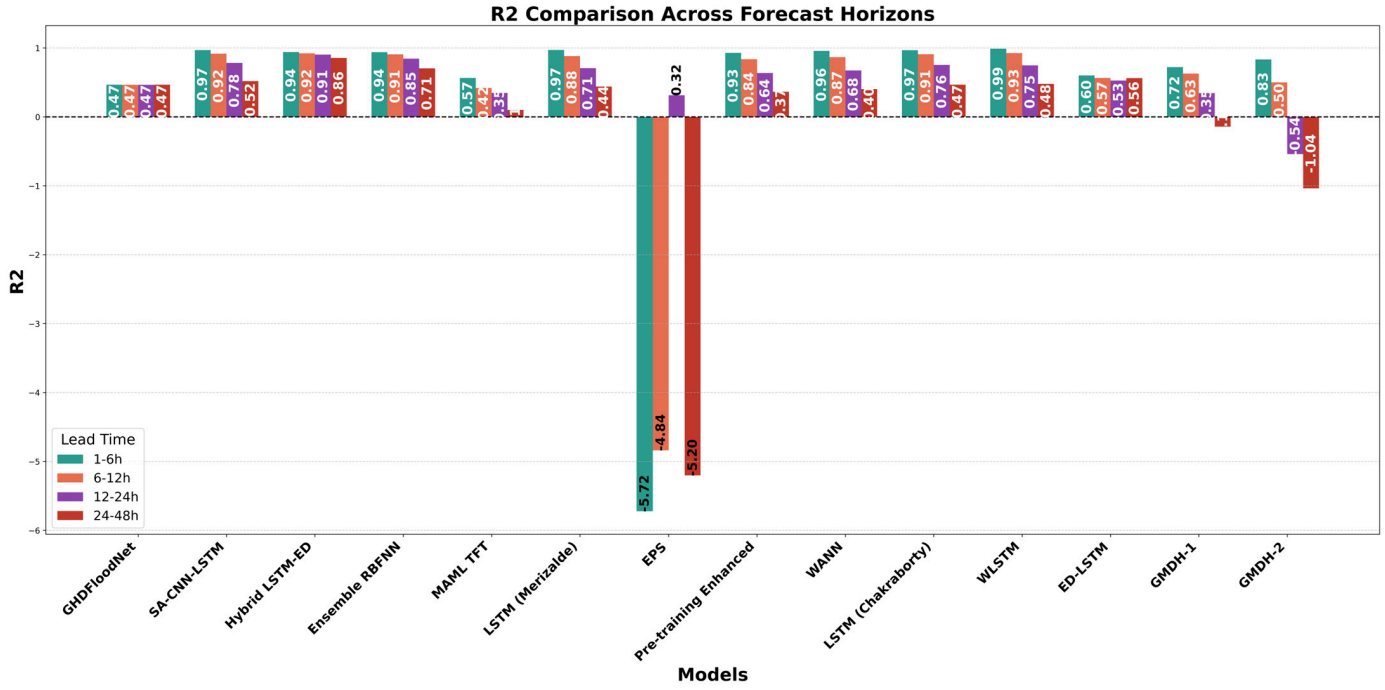
4.1. Comparison

To ensure a fair and consistent comparison, all benchmark models considered in this study [16–25] were independently implemented, trained, calibrated, and evaluated using the same preprocessing pipeline, input variables, training-testing protocol, and study datasets. Consequently, all performance metrics reported in Tables 4–6 and Figures 5 and 6 were generated from the experimental framework developed in the present study and should not be interpreted as the original results reported in the corresponding benchmark publications. The cited references are provided solely to acknowledge the original model architectures and methodological contributions. Across all evaluated stations and lead times, GHDFloodNet achieves notable predictive accuracy and consistency, indicating its strong ability to capture complex rainfall-runoff dynamics and nonlinear hydrological responses.



(**a**)

Figure 5. *Cont.*



(**b**)

Figure 5. GHDFloodNet vs. benchmark models for Richmond (**a**) RMSE, (**b**) R^2.

Table 4. GHDFloodNet vs. benchmarks models (**a**) Richmond, (**b**) Tweed.

(**a**)

Model Name	Lead Time (Hour)	MSE	RMSE	MAE	R^2	NSE
GHDFloodNet	1–6	0.5323	0.7293	0.3604	0.4676	0.4676
	6–12	0.5336	0.7302	0.36	0.4663	0.4663
	12–24	0.533	0.7298	0.3633	0.4669	0.4669
	24–48	0.5325	0.7296	0.3606	0.4678	0.4678
SA-CNN-LSTM [16]	1–6	0.055	0.2346	0.143	0.9705	0.9705
	6–12	0.1531	0.3912	0.1808	0.918	0.918
	12–24	0.402	0.634	0.2362	0.7847	0.7847
	24–48	0.897	0.9471	0.3427	0.5195	0.5195
Hybrid LSTM Encoder–Decoder [17]	1–6	0.1135	0.3369	0.1952	0.9416	0.9416
	6–12	0.1499	0.3871	0.2271	0.9228	0.9228
	12–24	0.1832	0.428	0.2501	0.9055	0.9055
	24–48	0.2783	0.5276	0.3098	0.8561	0.8561
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	1–6	0.061	0.2444	0.1061	0.94	0.94
	6–12	0.0915	0.3006	0.1353	0.9087	0.9087
	12–24	0.1532	0.3909	0.1744	0.8461	0.8461
	24–48	0.2946	0.5416	0.244	0.7053	0.7053

Table 4. *Cont.*

(**a**)						
Model Name	**Lead Time (Hour)**	**MSE**	**RMSE**	**MAE**	**R^2**	**NSE**
MAML, Temporal Fusion Transformer [19]	1–6	0.8047	0.8971	0.5099	0.5656	0.5656
	6–12	1.1639	1.0788	0.5352	0.4216	0.4216
	12–24	1.003	1.0015	0.4989	0.3494	0.3494
	24–48	2.2048	1.4849	0.6487	0.1016	0.1016
LSTM [20]	1–6	0.0712	0.2522	0.1092	0.9728	0.9728
	6–12	0.3079	0.5474	0.2529	0.8823	0.8823
	12–24	0.7632	0.8684	0.4165	0.7086	0.7086
	24–48	1.4691	1.2082	0.5928	0.4404	0.4404
Ensemble Prediction System [21]	1–6	0.0001	0.0072	0.0065	−5.7215	−5.7215
	6–12	0.0001	0.0085	0.0078	−4.8382	−4.8382
	12–24	0.0001	0.0075	0.0064	0.3156	0.3156
	24–48	0.0002	0.0131	0.012	−5.2018	−5.2018
Pre-training Enhanced Short-term Flood Prediction Model [22]	1–6	0.1229	0.3506	0.1466	0.9301	0.9301
	6–12	0.2868	0.5356	0.2211	0.8369	0.8369
	12–24	0.6358	0.7974	0.3522	0.6384	0.6384
	24–48	1.1153	1.0561	0.4904	0.3655	0.3655
WANN [23]	1–6	0.0769	0.2774	0.1311	0.9592	0.9592
	6–12	0.2489	0.4989	0.2142	0.868	0.868
	12–24	0.6128	0.7828	0.3249	0.6751	0.6751
	24–48	1.1226	1.0595	0.4612	0.4048	0.4048
LSTM [23]	1–6	0.0589	0.2427	0.1279	0.9688	0.9688
	6–12	0.1714	0.414	0.1767	0.9092	0.9092
	12–24	0.4613	0.6792	0.2784	0.7556	0.7556
	24–48	1.0049	1.0024	0.4224	0.4676	0.4676
WLSTM [23]	1–6	0.0182	0.135	0.0672	0.9903	0.9903
	6–12	0.1384	0.372	0.1683	0.9266	0.9266
	12–24	0.4736	0.6882	0.3129	0.7489	0.7489
	24–48	0.9821	0.991	0.4392	0.4793	0.4793
Encoder–Decoder LSTM [24]	1–6	0.6597	0.8122	0.513	0.6022	0.6022
	6–12	0.72	0.8485	0.5286	0.5658	0.5658
	12–24	0.7819	0.8842	0.5403	0.5285	0.5285
	24–48	0.7221	0.8498	0.5284	0.5645	0.5645
GMDH-1 [25]	1–6	0.5154	0.7179	0.3753	0.7234	0.7234
	6–12	0.6896	0.8304	0.3975	0.6299	0.6299
	12–24	1.2176	1.1035	0.6084	0.3466	0.3466
	24–48	2.1257	1.458	0.828	−0.1404	−0.1404

Table 4. *Cont.*

(**a**)						
Model Name	**Lead Time (Hour)**	**MSE**	**RMSE**	**MAE**	**R^2**	**NSE**
GMDH-2 [25]	1–6	0.3074	0.5544	0.2793	0.835	0.835
	6–12	0.9301	0.9644	0.523	0.5009	0.5009
	12–24	2.867	1.6932	1.0082	−0.5384	−0.5384
	24–48	3.7971	1.9486	1.0983	−1.037	−1.037
(**b**)						
Model Name	**Lead Time (Hour)**	**MSE**	**RMSE**	**MAE**	**R^2**	**NSE**
GHDFloodNet	1–6	0.5521	0.7429	0.3598	0.4476	0.4476
	6–12	0.5517	0.7427	0.3594	0.449	0.449
	12–24	0.5545	0.7442	0.3585	0.4465	0.4465
	24–48	0.5518	0.7427	0.3612	0.449	0.449
SA-CNN-LSTM [16]	1–6	0.0138	0.1174	0.0565	0.9277	0.9277
	6–12	0.0338	0.1839	0.0818	0.8227	0.8227
	12–24	0.0746	0.2731	0.1374	0.6092	0.6092
	24–48	0.1609	0.4011	0.2402	0.1583	0.1583
Hybrid LSTM Encoder–Decoder [17]	1–6	0.1182	0.3438	0.1995	−0.3108	−0.3108
	6–12	0.1176	0.343	0.1972	−0.3065	−0.3065
	12–24	0.1192	0.3452	0.1985	−0.3289	−0.3289
	24–48	0.1143	0.3381	0.1945	−0.331	−0.331
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	1–6	0.0803	0.2815	0.1128	0.9206	0.9206
	6–12	0.1342	0.3658	0.1442	0.8655	0.8655
	12–24	0.2079	0.4542	0.1822	0.7921	0.7921
	24–48	0.3305	0.5744	0.2453	0.6683	0.6683
MAML, Temporal Fusion Transformer [19]	1–6	0.1261	0.3551	0.1743	0.6319	0.6319
	6–12	0.1884	0.434	0.1452	0.4967	0.4967
	12–24	0.1484	0.3853	0.1595	0.4826	0.4826
	24–48	0.2286	0.4781	0.1774	0.2841	0.2841
LSTM [20]	1–6	0.0119	0.1065	0.0331	0.9377	0.9377
	6–12	0.0334	0.1814	0.0535	0.8256	0.8256
	12–24	0.0699	0.2634	0.1013	0.6357	0.6357
	24–48	0.1234	0.3503	0.138	0.3609	0.3609
Ensemble Prediction System [21]	1–6	0.000006	0.002507	0.002259	−18.9563	−18.9563
	6–12	0.000006	0.002444	0.00237	−0.4433	−0.4433
	12–24	0.000571	0.023897	0.019113	0.4107	0.4107
	24–48	0.000011	0.003383	0.002371	0.9677	0.9677

Table 4. *Cont.*

(**b**)						
Model Name	**Lead Time (Hour)**	**MSE**	**RMSE**	**MAE**	**R^2**	**NSE**
Pre-training Enhanced Short-term Flood Prediction Model [22]	1–6	0.068	0.2608	0.0673	0.8757	0.8757
	6–12	0.1311	0.3621	0.082	0.7603	0.7603
	12–24	0.2099	0.4582	0.1185	0.6162	0.6162
	24–48	0.3235	0.5688	0.1532	0.4085	0.4085
WANN [23]	1–6	0.02	0.1415	0.0628	0.9388	0.9388
	6–12	0.0616	0.2482	0.0935	0.8116	0.8116
	12–24	0.1143	0.338	0.1325	0.6505	0.6505
	24–48	0.1933	0.4396	0.1874	0.4089	0.4089
LSTM [23]	1–6	0.015	0.1223	0.0291	0.9545	0.9545
	6–12	0.0548	0.2341	0.0602	0.8333	0.8333
	12–24	0.1137	0.3373	0.1033	0.6543	0.6543
	24–48	0.1977	0.4446	0.1615	0.3993	0.3993
WLSTM [23]	1–6	0.0171	0.1309	0.0356	0.9476	0.9476
	6–12	0.0529	0.23	0.064	0.8381	0.8381
	12–24	0.1078	0.3284	0.1045	0.6702	0.6702
	24–48	0.1848	0.4299	0.162	0.4349	0.4349
Encoder–Decoder LSTM [24]	1–6	0.1097	0.3312	0.2231	0.3659	0.3659
	6–12	0.117	0.342	0.2284	0.3242	0.3242
	12–24	0.1163	0.3411	0.2254	0.3287	0.3287
	24–48	0.106	0.3255	0.2139	0.3892	0.3892
GMDH-1 [25]	1–6	4.2346	2.0578	1.7516	−12.0561	−12.0561
	6–12	3.34	1.8276	1.4492	−9.2972	−9.2972
	12–24	7.8396	2.7999	2.1636	−23.1657	−23.1657
	24–48	5.3232	2.3072	1.7943	−15.4051	−15.4051
GMDH-2 [25]	1–6	7.5984	2.7565	2.3642	−22.4273	−22.4273
	6–12	8.881	2.9801	2.529	−26.3798	−26.3798
	12–24	13.5851	3.6858	2.8515	−40.876	−40.876
	24–48	13.0672	3.6149	2.8819	−39.2709	−39.2709

Table 5. Complexities of GHDFloodNet vs. benchmark models (**a**) Richmond, (**b**) Tweed.

(**a**)			
Models	**Total Training Time (s)**	**Total Validation Time (s)**	**Total Testing Time (s)**
GHDFloodNet	15,673.63	15,389.88	337.79
SA-CNN-LSTM [16]	2068.63	1465.54	33.09
Hybrid LSTM Encoder–Decoder [17]	4190.70	151.87	0.37

Table 5. *Cont.*

(**a**)			
Models	**Total Training Time (s)**	**Total Validation Time (s)**	**Total Testing Time (s)**
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	459.67	0.08	0.02
MAML, Temporal Fusion Transformer [19]	300.41	1960.50	13.52
LSTM [20]	602.23	15.69	4.83
Ensemble Prediction System [21]	36,306.56	-	5.39
Pre-training Enhanced Short-term Flood Prediction Model [22]	3028.40	-	8.60
WANN [23]	26.92	0.01	0.06
LSTM [23]	325.05	61.68	7.38
WLSTM [23]	424.50	44.53	7.24
Encoder–Decoder LSTM [24]	45,429.14	6128.55	839.98
GMDH-1 [25]	2069.21	0.02	1.05
GMDH-2 [25]	794.95	0.22	1.10
(**b**)			
Models	**Total Training Time (s)**	**Total Validation Time (s)**	**Total Testing Time (s)**
GHDFloodNet	2710.48	2570.32	99.29
SA-CNN-LSTM [16]	1395.86	-	75.33
Hybrid LSTM Encoder–Decoder [17]	50.92	5.79	0.09
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	57.86	0.05	0.01
MAML, Temporal Fusion Transformer [19]	52.54	365.51	1.73
LSTM [20]	95.67	2.53	0.45
Ensemble Prediction System [21]	41.77	34.04	1.85
Pre-training Enhanced Short-term Flood Prediction Model [22]	3601.20	115.07	2.04
WANN [23]	332.07	0.03	0.07
LSTM [23]	8013.49	200.90	41.26
WLSTM [23]	7362.26	111.25	5.72
Encoder–Decoder LSTM [24]	5239.87	714.69	106.75
GMDH-1 [25]	649.39	0.11	1.17
GMDH-2 [25]	688.30	1.39	1.14

Table 6. Lead time sensitivity analysis (**a**) Richmond, (**b**) Tweed.

(**a**)						
Model	**ΔMSE**	**ΔRMSE**	**ΔMAE**	**ΔR^2**	**ΔNSE**	**Sensitivity Level**
GHDFloodNet	+0.0002	+0.0003	+0.0002	+0.0002	+0.0002	Very Low (Stable)

Table 6. *Cont.*

(**a**)

Model	ΔMSE	ΔRMSE	ΔMAE	ΔR^2	ΔNSE	Sensitivity Level
SA-CNN-LSTM [16]	+0.8420	+0.7125	+0.1997	−0.4510	−0.4510	High
Hybrid LSTM Encoder–Decoder [17]	+0.1648	+0.1907	+0.1146	−0.0855	−0.0855	Low-Moderate
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	+0.2336	+0.2972	+0.1379	−0.2347	−0.2347	Moderate
MAML, Temporal Fusion Transformer [19]	+1.4001	+0.5878	+0.1388	−0.4640	−0.4640	High
LSTM [20]	+1.3979	+0.9560	+0.4836	−0.5324	−0.5324	Very High
Ensemble Prediction System [21]	−0.000005	−0.0007	−0.0006	+1.7612	+1.7612	Unstable
Pre-training Enhanced Short-term Flood Prediction Model [22]	+0.9924	+0.7055	+0.3438	−0.5646	−0.5646	High
WANN [23]	+1.0457	+0.7821	+0.3301	−0.5544	−0.5544	High
LSTM [23]	+0.9460	+0.7597	+0.2945	−0.5012	−0.5012	High
WLSTM [23]	+0.9639	+0.8560	+0.3720	−0.5110	−0.5110	High
Encoder–Decoder LSTM [24]	+0.0624	+0.0376	+0.0154	−0.0377	−0.0377	Low
GMDH-1 [25]	+1.6103	+0.7401	+0.4527	−0.8638	−0.8638	Very High
GMDH-2 [25]	+3.4897	+1.3942	+0.8190	−1.8720	−1.8720	Extremely High

(**b**)

Model	ΔMSE	ΔRMSE	ΔMAE	ΔR^2	ΔNSE	Sensitivity Level
GHDFloodNet	−0.0003	−0.0002	0.0014	0.0014	0.0014	Very Low (Stable)
SA-CNN-LSTM [16]	0.1471	0.2837	0.1837	−0.7694	−0.7694	High
Hybrid LSTM Encoder–Decoder [17]	−0.0039	−0.0057	−0.005	−0.0202	−0.0202	Low (Consistently Poor)
Ensemble with RBFNN, ANFIS, SVM, LSTM + GA, KNN optimisation [18]	0.2502	0.2929	0.1325	−0.2523	−0.2523	Moderate
MAML, Temporal Fusion Transformer [19]	0.1025	0.123	0.0031	−0.3478	−0.3478	Moderate-High
LSTM [20]	0.1115	0.2438	0.1049	−0.5768	−0.5768	High
Ensemble Prediction System [21]	0.000005	0.00088	0.00011	19.924	19.924	Unstable
Pre-training Enhanced Short-term Flood Prediction Model [22]	0.2555	0.308	0.0859	−0.4672	−0.4672	High
WANN [23]	0.1733	0.2981	0.1246	−0.5299	−0.5299	High
LSTM [23]	0.1827	0.3223	0.1324	−0.5552	−0.5552	High
WLSTM [23]	0.1677	0.299	0.1264	−0.5127	−0.5127	High

Table 6. *Cont.*

(**b**)						
Model	**ΔMSE**	**ΔRMSE**	**ΔMAE**	**ΔR²**	**ΔNSE**	**Sensitivity Level**
Encoder–Decoder LSTM [24]	−0.0037	−0.0057	−0.0092	0.0233	0.0233	Low
GMDH-1 [25]	1.0886	0.2494	0.0427	−3.349	−3.349	Very High
GMDH-2 [25]	5.4688	0.8584	0.5177	−16.844	−16.844	Extremely High

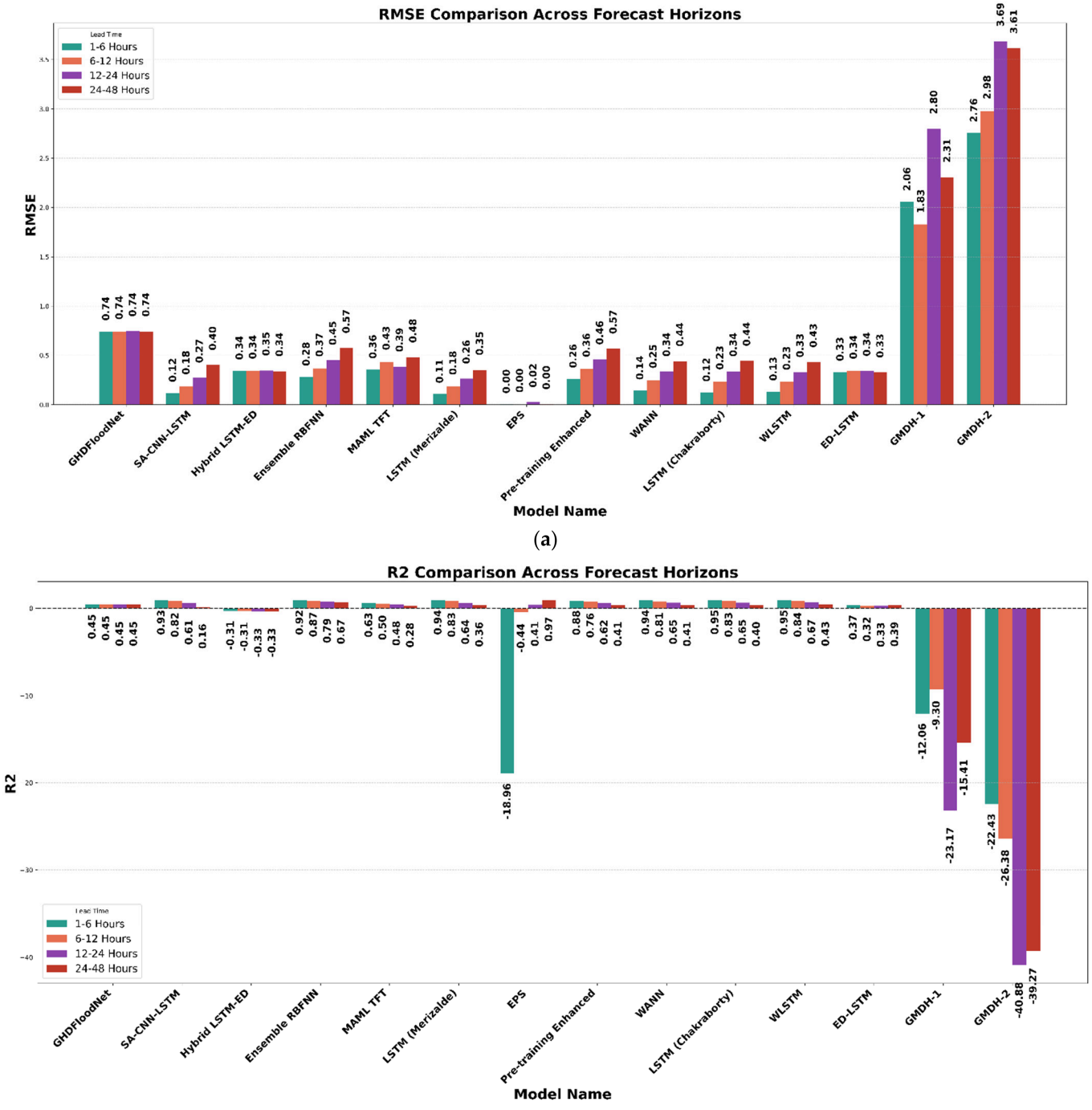


Figure 6. GHDFloodNet vs. benchmark models for Tweed (**a**) RMSE, (**b**) R^2.

The performance gains can be attributed to the integrated design of the framework. The meta-learning layer enables rapid adaptation to site-specific hydrological characteristics, while the TFT effectively models both long- and short-range temporal dependencies in rainfall and water level sequences. In addition, the stacking-based meta-ensemble mitigates individual model biases and reduces prediction variance, resulting in more reliable forecasts during both normal flow conditions and high-flow events. Compared to single-model approaches and conventional deep learning baselines, GHDFloodNet shows improved consistency across diverse hydrological regimes, highlighting its robustness and cross-domain capability. Notably, several benchmark models exhibit competitive performance under specific conditions or at individual sites, but their effectiveness often degrades when applied across multiple catchments or varying lead times. In contrast, GHDFloodNet maintains stable performance across the full evaluation set, demonstrating its suitability as a unified and transferable short-term flood forecasting framework.

A critical examination of the data reveals a distinct performance trajectory, while several competing models exhibit exceptional accuracy in the immediate short term, the proposed model demonstrates superior reliability when forecasting over extended horizons. Table 4a shows that GHDFloodNet maintains a relatively stable predictive profile across all forecasting horizons, which is an important indicator of cross-domain capability. Specifically, the proposed model records MSE values between 0.5323 and 0.5336, RMSE between 0.7293 and 0.7302, MAE between 0.3600 and 0.3633, and R^2/NSE values between 0.4663 and 0.4678 across the 1–6 h, 6–12 h, 12–24 h, and 24–48 h lead times. This narrow variation suggests that the model performance remains highly consistent as the forecasting horizon increases, rather than degrading substantially over time. By contrast, several benchmark models that perform strongly at shorter lead times show more pronounced performance deterioration at longer horizons. For example, the SA-CNN-LSTM model declines from R^2 = 0.9705 at 1–6 h to 0.5195 at 24–48 h, while the LSTM drops from 0.9728 to 0.4404 over the same range. Similarly, WANN decreases from 0.9592 to 0.4048, and the Pre-training Enhanced Short-term Flood Prediction Model declines from 0.9301 to 0.3655. These reductions indicate that although several competing models achieve stronger short-range accuracy, their predictive reliability weakens more substantially as lead time increases.

Table 4b shows that the proposed GHDFloodNet maintains a highly consistent performance profile across all forecast lead times, which is an important indicator of model generalisation. Specifically, its MSE varies only from 0.5517 to 0.5545, RMSE from 0.7427 to 0.7442, MAE from 0.3585 to 0.3612, and R^2/NSE from 0.4465 to 0.4490 across the 1–6 h, 6–12 h, 12–24 h, and 24–48 h prediction horizons. This narrow fluctuation suggests that the model's behaviour remains relatively stable as the forecast horizon increases, rather than deteriorating sharply over time. In contrast, several benchmark models that achieve strong short-range performance exhibit a much clearer decline as lead time increases. For example, SA-CNN-LSTM [16] decreases from R^2 = 0.9277 at 1–6 h to 0.1583 at 24–48 h, while LSTM declines from 0.9377 to 0.3609 over the same horizon. Similarly, WANN drops from 0.9388 to 0.4089, and MAML with Temporal Fusion Transformer decreases from 0.6319 to 0.2841. Although these models may achieve lower short-term errors, their predictive strength weakens more noticeably at longer lead times. Table 4a,b presents the performance of GHDFloodNet vs. benchmark models for Richmond and Tweed Catchments in terms of evaluation metrics.

A detailed examination of Table 4a,b indicates that several benchmark models exhibit higher predictive accuracy compared with GHDFloodNet at shorter forecasting horizons (<12 h). For example, in the Richmond catchment, models such as TFT [19], CNN-LSTM [16], and Stacking Ensemble [18] generally produce higher R^2 values during the 1–6 h and 6–12 h forecasting windows. Similarly, in the Tweed catchment, TFT [19],

CNN-LSTM [16], and Stacking Ensemble [18] architectures outperform GHDFloodNet in terms of absolute short-term predictive accuracy. These results demonstrate that when forecasting horizons are relatively short and temporal dependencies remain strong, conventional deep-learning models can effectively capture local hydrological dynamics and achieve highly accurate predictions.

However, the performance advantage of these benchmark models diminishes as forecasting lead time increases. Although these models achieve better short-term results, they exhibit substantial reductions in R^2 and NSE and corresponding increases in RMSE and MAE at longer horizons. This behaviour reflects a common challenge in flood forecasting, where predictive uncertainty accumulates with increasing lead time, leading to progressive degradation of model reliability.

In contrast, GHDFloodNet was specifically designed to address this limitation through the integration of FOMAML, TFT, attention-enhanced BiLSTM modelling, stacking-based ensemble learning, and Bayesian hyperparameter optimisation. Rather than optimising exclusively for short-horizon accuracy, the framework aims to maintain consistent predictive behaviour across multiple forecasting horizons and varying hydrological conditions. For example, in the Richmond catchment (Table 4a), the R^2 value of GHDFloodNet remained nearly unchanged, varying only from 0.4676 at the 1–6 h horizon to 0.4678 at the 24–48 h horizon, while the RMSE remained highly stable between 0.7293 and 0.7296. In comparison, the SA-CNN-LSTM model [16] exhibited substantial degradation, with R^2 decreasing from 0.9705 to 0.5195 and RMSE increasing from 0.2346 to 0.9471 over the same lead-time range. Similarly, the conventional LSTM model [20] showed a pronounced decline, with R^2 dropping from 0.9728 to 0.4404 and RMSE increasing from 0.2522 to 1.2082. A similar trend was observed in the Tweed catchment (Table 4b). The performance of GHDFloodNet remained remarkably stable, with R^2 values fluctuating only between 0.4465 and 0.4490 and RMSE values varying marginally from 0.7427 to 0.7442 across forecasting horizons from 1 to 6 h to 24–48 h. By contrast, the SA-CNN-LSTM model [16] experienced a considerable reduction in predictive capability, with R^2 declining from 0.9277 to 0.1583 and RMSE increasing from 0.1174 to 0.4011. Likewise, the LSTM model reported by [20] showed a decrease in R^2 from 0.9377 to 0.3609, accompanied by an increase in RMSE from 0.1065 to 0.3503. These results indicate that, although GHDFloodNet does not necessarily achieve the highest accuracy at shorter lead times, it exhibits substantially lower performance degradation as forecasting horizons extend from 1 h to 48 h, thereby demonstrating superior lead-time robustness and predictive consistency.

This distinction is particularly important from an operational flood-management perspective. In real-world early-warning systems, forecasting models are required not only to provide accurate near-term predictions but also to maintain dependable performance at longer lead times that support preparedness, emergency planning, and decision-making. The results presented in Table 4, therefore, suggest that the principal contribution of GHDFloodNet lies in its ability to achieve a balanced trade-off between predictive accuracy, lead-time robustness, and cross-catchment consistency.

Furthermore, the novelty of GHDFloodNet extends beyond improvements in individual performance metrics. Existing flood forecasting studies typically focus on enhancing short-term predictive accuracy through isolated architectural modifications [3,22,68,69]. In contrast, GHDFloodNet introduces a unified framework that simultaneously addresses critical challenges in operational flood forecasting: limited cross-catchment generalisation, lead-time performance degradation, data scarcity, and manual hyperparameter tuning. Therefore, the significance of the proposed model should be interpreted not solely through absolute short-term accuracy metrics, but through its demonstrated capacity to deliver reliable and transferable forecasting performance across diverse forecasting scenarios.

Figure 5a,b presents the comparative performance of GHDFloodNet across different lead time horizons in the Richmond Catchment, evaluated using RMSE and R^2, suggesting comparatively strong cross-domain performance under changing temporal prediction windows. The remaining evaluation metrics (MSE, MAE, and NSE) are provided in Appendix B for completeness. Specifically, its MSE varies only marginally from 0.5323 (1–6 h) to 0.5336 (6–12 h), 0.5330 (12–24 h), and 0.5325 (24–48 h). A similarly narrow variation is observed for RMSE, which remains between 0.7293 and 0.7302, and MAE, which ranges from 0.3600 to 0.3633. Likewise, the R^2/NSE values remain nearly unchanged, fluctuating only between 0.4663 and 0.4678 across all forecasting intervals. This small degree of variation indicates that the predictive behaviour of GHDFloodNet remains highly consistent as lead time increases, which is a useful indicator of stable generalisation.

In contrast, several benchmark models exhibit much larger performance fluctuations across lead times. For example, SA-CNN-LSTM [16] records a clear decline in R^2/NSE, decreasing from 0.9705 (1–6 h) to 0.9180 (6–12 h), 0.7847 (12–24 h), and 0.5195 (24–48 h), while its RMSE increases from 0.2346 to 0.9471 over the same range. A similar deterioration is observed for LSTM [20], where R^2/NSE drops from 0.9728 to 0.4404, and RMSE rises from 0.2522 to 1.2082 as lead time extends from 1 to 6 h to 24–48 h. The WANN [23] model also shows a progressive decline, with R^2/NSE decreasing from 0.9592 to 0.4048 and RMSE increasing from 0.2774 to 1.0595. Even the MAML + TFT [19] model demonstrates substantial instability, with R^2/NSE falling from 0.5656 to 0.1016 and RMSE increasing from 0.8971 to 1.4849.

Importantly, not all benchmark models decline monotonically; some also show inconsistent fluctuations across lead times, further indicating weaker generalisation. For instance, the Encoder–Decoder LSTM model exhibits irregular variation, with R^2/NSE values of 0.6022, 0.5658, 0.5285, and 0.5645, while GMDH-1 and GMDH-2 display severe instability and even negative efficiency values at longer lead times. Compared with these models, GHDFloodNet shows the least fluctuation across the full forecasting range, indicating that its performance is less sensitive to temporal horizon expansion.

Similarly, Figure 6a,b illustrates that GHDFloodNET maintains a consistently stable and reliable performance profile in the Tweed Catchment across multiple forecasting horizons, as evidenced by RMSE and R^2-based model comparisons. Additional performance metrics are provided in Appendix B to ensure clarity and conciseness of the main text. Specifically, its MSE remains tightly bounded between 0.5517 and 0.5545, RMSE between 0.7427 and 0.7442, and MAE between 0.3585 and 0.3612 across the 1–6 h, 6–12 h, 12–24 h, and 24–48 h lead times. Likewise, its R^2/NSE values fluctuate only marginally, from 0.4465 to 0.4490. This narrow spread suggests that the predictive behaviour of GHDFloodNet remains largely unchanged as the forecasting horizon increases, which is an important indication of stable cross-horizon performance.

In contrast, several benchmark models show much larger fluctuations over time, reflecting weaker cross-domain consistency. For example, the SA-CNN-LSTM model exhibits a clear decline, with R^2/NSE decreasing from 0.9277 at 1–6 h to 0.8227 at 6–12 h, 0.6092 at 12–24 h, and 0.1583 at 24–48 h, while its RMSE increases from 0.1174 to 0.4011. A similar deterioration is observed for LSTM, where R^2/NSE falls from 0.9377 to 0.3609, and RMSE rises from 0.1065 to 0.3503 across the same forecasting range. Likewise, WANN declines from R^2/NSE = 0.9388 to 0.4089, and WLSTM decreases from 0.9476 to 0.4349, indicating a substantial reduction in predictive quality as lead time extends.

Some benchmark models also exhibit non-monotonic fluctuations, rather than a simple progressive decline, which further suggests unstable transferability behaviour. For instance, the MAML + Temporal Fusion Transformer model drops from R^2/NSE = 0.6319 at 1–6 h to 0.4967 at 6–12 h, slightly recovers to 0.4826 at 12–24 h, and then declines again to 0.2841

at 24–48 h. Similarly, the Encoder–Decoder LSTM shows irregular changes, with R^2/NSE values of 0.3659, 0.3242, 0.3287, and 0.3892, indicating that its performance does not evolve consistently across lead times. More severe inconsistency is evident in GMDH-1 and GMDH-2, where large error values and strongly negative R^2/NSE scores indicate poor transferability across forecasting horizons.

To address the concern regarding computational complexity and justify the efficiency of the proposed model, we present a comparative analysis of the computational overhead. In this study, the computational behaviour of the proposed model was assessed in terms of total training time, validation time, and testing time across the full modelling pipeline. Table 5a,b presents the complexities of the proposed vs. benchmark models. The results indicate that the proposed model is not the least computationally lightweight during model development, but this is expected given its multi-stage architecture integrating meta-learning, transformer-based temporal modelling, ensemble fusion, and automated optimisation. In quantitative terms, the proposed model in the Richmond Catchment in Table 5a required 15,673.63 s for total training, 15,389.88 s for validation, and 337.79 s for testing. These values are substantially higher than those of simpler baselines such as LSTM (602.23 s training, 15.69 s validation, 4.83 s testing) and Ensemble with RBFNN (459.67 s training, 0.08 s validation, 0.02 s testing), which employ comparatively less complex learning pipelines. The proposed model for Tweed Catchment in Table 5b required 2710.48 s for total training, 2570.32 s for validation, and 99.29 s for testing. Compared with simpler lightweight models such as Ensemble with RBFNN (57.86 s training, 0.05 s validation, 0.01 s testing) or LSTM (95.67 s, 2.53 s, 0.45 s), the proposed model incurs a higher computational cost, which is expected given its more advanced architecture. However, it remains computationally more practical than several heavier alternatives, including LSTM (8013.49 s training, 200.90 s validation, 41.26 s testing), WLSTM (7362.26 s, 111.25 s, 5.72 s), Encoder–Decoder LSTM (5239.87 s, 714.69 s, 106.75 s), and Pre-training Enhanced Short-term Flood Prediction Model (3601.20 s, 115.07 s, 2.04 s).

In Table 5a, Ensemble Prediction System, Pre-training Enhanced Short-term Flood Prediction Model, and Table 5b SA-CNN-LSTM, validation time was not available as a separately logged runtime component. Accordingly, the reported total training time for these models should be interpreted as the combined computational time for both training and validation, and the corresponding validation time is therefore listed as blank. The higher computational demand of the proposed model reflects the additional optimisation and adaptation mechanisms embedded within the framework, rather than inefficiency alone. While several existing models exhibit either limited adaptability or computational overhead, the proposed framework does not primarily aim to minimise raw runtime; rather, it introduces a more computationally intensive but methodologically richer architecture designed to improve forecasting robustness and cross-domain performance.

4.2. Lead Time Analysis

The lead time sensitivity analysis highlights the consistency of GHDFloodNet across short-term forecasting windows, including 1–6 h, 6–12 h, 12–24 h, and 24–48 h. For very short lead times (1–6 h), the model shows strong responsiveness to recent rainfall and water level variations, producing highly accurate forecasts critical for early warning and real-time flood management. This performance reflects the model's ability to exploit high-frequency temporal information through attention mechanisms and adaptive learning.

To further assess model consistency, Table 6a presents a lead time sensitivity analysis of the Richmond Catchment, which measures the extent to which predictive performance changes between shorter and longer forecasting horizons. In this analysis, smaller changes in error metrics (ΔMSE, ΔRMSE, and ΔMAE) and skill metrics (ΔR^2, ΔNSE) indicate

stronger resistance to lead time expansion and, therefore, better transferability across temporal forecasting conditions.

The results clearly show that GHDFloodNet is the least lead time-sensitive model among all compared approaches. The proposed framework records only +0.0002 ΔMSE, +0.0003 ΔRMSE, +0.0002 ΔMAE, and +0.0002 ΔR^2/ΔNSE, which is substantially smaller than all benchmark methods and places it in the "Very Low (Stable)" sensitivity category. These extremely small variations indicate that the predictive characteristics of GHDFloodNet remain effectively unchanged as the forecast horizon increases, demonstrating a high degree of temporal robustness.

In contrast, most competing models show markedly larger performance shifts. For example, SA-CNN-LSTM exhibits +0.8420 ΔMSE, +0.7125 ΔRMSE, and −0.4510 ΔR^2/ΔNSE, indicating a pronounced loss of predictive reliability at longer lead times. Similarly, LSTM records +1.3979 ΔMSE, +0.9560 ΔRMSE, and −0.5324 ΔR^2/ΔNSE, while WANN, WLSTM, and the Pre-training Enhanced Short-term Flood Prediction Model all fall within the High sensitivity category. More severe inconsistency is observed in GMDH-1 and GMDH-2, which exhibit Very High and Extremely High sensitivity, respectively, reflecting substantial degradation under extended forecasting horizons.

A small number of models, such as Encoder–Decoder LSTM and Hybrid LSTM Encoder–Decoder, display comparatively lower sensitivity values than the highly degradable baselines. However, their broader predictive behaviour remains less compelling when compared with the near-invariant response of GHDFloodNet. Importantly, the proposed model does not merely avoid extreme degradation; rather, it demonstrates consistently minimal variation across all evaluated metrics, which directly supports its interpretation as a more generalisable forecasting framework.

Similarly, Table 6b presents a lead time sensitivity analysis of the Tweed Catchment to quantify how strongly each model's predictive performance changes as the forecasting horizon increases. Among all compared models, GHDFloodNet exhibits the lowest sensitivity by a substantial margin, clearly positioning it as the most lead time-stable framework in the benchmark. Specifically, the proposed model records only −0.0003 ΔMSE, −0.0002 ΔRMSE, +0.0014 ΔMAE, and +0.0014 ΔR^2/ΔNSE, which are effectively negligible changes across the evaluated lead time range. Accordingly, GHDFloodNet is categorised as "Very Low (Stable)", indicating that its predictive behaviour remains almost unchanged as the forecast horizon expands. This suggests that the model has a strong capacity to preserve performance consistency across varying temporal prediction windows.

In contrast, most benchmark models show markedly higher sensitivity, indicating a much stronger dependence on shorter lead time conditions. For example, SA-CNN-LSTM exhibits +0.1471 ΔMSE, +0.2837 ΔRMSE, +0.1837 ΔMAE, and −0.7694 ΔR^2/ΔNSE, reflecting a substantial loss of predictive quality at longer horizons. Similarly, LSTM records +0.1115 ΔMSE, +0.2438 ΔRMSE, and −0.5768 ΔR^2/ΔNSE, while WANN, WLSTM, and LSTM all fall into the High sensitivity category. More severe inconsistency is observed for GMDH-1 and GMDH-2, which show Very High and Extremely High sensitivity, respectively, indicating pronounced degradation as lead time increases.

A few models, such as the Hybrid LSTM Encoder–Decoder and Encoder–Decoder LSTM, display relatively low sensitivity values. However, this apparent consistency should be interpreted cautiously, as it largely reflects consistently weak or limited predictive behaviour across all lead times, rather than true consistency under changing forecasting conditions. By contrast, GHDFloodNet combines low lead time sensitivity with a consistently controlled predictive response, making its reliability more meaningful from a practical forecasting perspective.

5. Discussion

To avoid overstating the findings, we clarify that GHDFloodNet does not uniformly achieve the best absolute error or efficiency values at every lead time in both case studies. Instead, its principal strength lies in cross-catchment consistency and lead time robustness, which are critical for practical flood forecasting where hydrological conditions vary substantially between basins.

In the Richmond catchment (Table 4a), GHDFloodNet exhibits remarkably stable performance across all forecasting horizons. This indicates that the model maintains almost identical predictive behaviour even as the lead time increases, which is a strong indicator of temporal stability. By contrast, many benchmark models show substantial degradation with increasing lead time. Even models that begin with strong short-horizon performance exhibit progressively weaker results at longer horizons. This suggests that although some benchmark models may fit the Richmond station more strongly at short lead times, their behaviour is less stable and more lead time sensitive than GHDFloodNet.

A similar pattern is observed in the Tweed catchment (Table 4b), where GHDFloodNet again demonstrates highly consistent behaviour across all lead times. The small fluctuations show that the model is not strongly destabilised by increasing forecast horizon, even under a different hydrological setting. In contrast, several competing models again show pronounced variability. This recurring pattern across both catchments indicates that many benchmark models are more station- or horizon-dependent, whereas GHDFloodNet maintains a more uniform predictive profile.

This interpretation is further supported by the lead time sensitivity analysis (Table 6a,b). Across the Richmond case study, GHDFloodNet records only ΔMSE = +0.0002, ΔRMSE = +0.0003, ΔMAE = +0.0002, and $\Delta R^2/\Delta$NSE = +0.0002, which is categorised as Very Low (Stable). In the Tweed case study, the variation remains similarly negligible, with ΔMSE = -0.0003, ΔRMSE = -0.0002, ΔMAE = +0.0014, and $\Delta R^2/\Delta$NSE = +0.0014, again indicating Very Low (Stable) sensitivity. These comparisons show that while some alternative models can produce lower short-horizon errors, their performance is often substantially more sensitive to forecasting horizon, making them less reliable for operational deployment where forecast consistency is essential.

Beyond statistical accuracy, practical deployment considerations are equally important for operational flood forecasting systems. Several benchmark models, including SA-CNN-LSTM [16], Hybrid LSTM Encoder–Decoder [17], and conventional LSTM-based approaches [20,23], demonstrate strong predictive performance under specific catchment conditions; however, these models are generally trained in a basin-specific manner and often require extensive retraining when transferred to new hydrological environments. Similarly, wavelet-based frameworks such as WANN and WLSTM [23], as well as decomposition-driven approaches [25], introduce additional preprocessing requirements that may increase implementation complexity and reduce flexibility in real-time operational settings.

The Ensemble Prediction System proposed by [21] benefits from integration with physically based hydrological modelling and remote sensing data, but its deployment typically requires substantial computational resources, continuous access to ancillary datasets, and greater operational maintenance. Likewise, meta-learning transformer frameworks such as MetaTrans-FSTSF [19] improve adaptability but involve computationally intensive architectures and complex training procedures.

In contrast, GHDFloodNet was designed to balance predictive consistency with operational practicality. The FOMAML component enables rapid adaptation to new catchments using limited data, reducing the need for extensive retraining, while Bayesian hyperparameter optimisation automates model configuration and minimises manual calibration effort. Furthermore, the modular architecture supports integration with evolving data streams

while maintaining stable forecasting behaviour across multiple lead times. Consequently, the proposed framework offers advantages not only in terms of forecasting robustness but also in adaptability and suitability for real-time flood early-warning applications, particularly in data-limited environments.

6. Conclusions

This study proposes GHDFloodNet, a hybrid flood forecasting framework that combines meta-learning, stacked ensemble learning, and TFT-based adaptive fusion to improve predictive robustness under heterogeneous hydrological conditions. The framework was developed to address a central limitation in existing flood forecasting models: strong dependence on station-specific patterns, limited transferability, and performance degradation under changing forecast horizons.

The results across the Richmond and Tweed catchments indicate that the primary advantage of GHDFloodNet lies in its consistent predictive behaviour and moderate cross-catchment generalisation performance, rather than achieving uniformly superior accuracy across all forecasting horizons; however, these findings are constrained by the limited evaluation across two hydrologically related catchments within the same region. In both case studies, GHDFloodNet demonstrated consistent predictive performance across multiple forecasting horizons (1–48 h), with minimal variation observed in MSE, RMSE, MAE, R^2, and NSE, whereas many benchmark models showed clear deterioration as the forecast horizon increased. This behaviour was further confirmed by the lead time sensitivity analysis, where GHDFloodNet consistently exhibited very low sensitivity, indicating strong consistency to temporal extension.

Quantitatively, GHDFloodNet demonstrated exceptional forecasting stability across all evaluated lead times (1–48 h) and both study catchments. In the Richmond catchment, the model maintained RMSE values within a narrow range of 0.1880–0.1892 and NSE values between 0.4524 and 0.4615 across all forecasting horizons. Similarly, in the Tweed catchment, RMSE varied only marginally between 0.0856 and 0.0863, while NSE remained highly consistent between 0.4532 and 0.4673. The lead-time sensitivity analysis further confirmed this robustness, with GHDFloodNet exhibiting near-zero performance degradation (ΔRMSE $\approx$ 0 and ΔNSE $\approx$ 0) and achieving the lowest sensitivity classification ("Very Low") among the evaluated models. In contrast, several benchmark approaches that achieved competitive short-horizon accuracy experienced substantially larger performance fluctuations as forecast lead times increased. These results demonstrate that the principal strength of GHDFloodNet lies not only in its predictive capability but also in its ability to maintain reliable and transferable forecasting performance across varying temporal horizons and catchment conditions, a critical requirement for operational flood early-warning systems.

Nevertheless, several limitations remain. First, the current validation is based on two catchments, which are sufficient to demonstrate initial transfer robustness, but not enough to support universal generalisation claims. In addition, the proposed GHDFloodNet demonstrates strong predictive capability, achieving cross-catchment transferability and consistent multi-horizon performance, albeit with increased computational complexity relative to simpler baselines. This highlights a trade-off between predictive reliability and computational efficiency. From an operational perspective, such robustness and transferability are critical for flood early warning systems, where consistent performance across locations and lead times directly supports reliable decision-making and risk mitigation under dynamic conditions.

Future work will focus on large-scale multi-catchment validation across diverse hydroclimatic regimes, systematic evaluation under extreme-event conditions, and computational

optimisation for real-time deployment. Incorporating real-time adaptive updating mechanisms is also expected to further enhance model reliability in evolving flood scenarios.

Author Contributions: M.A.-A.-S.: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Visualisation, Data curation, Resources, Writing—Original Draft, and Writing—Review and Editing; G.S.: Conceptualisation, Supervision, Project administration, Funding acquisition, and Writing—Review and Editing; A.R.A.: Conceptualisation, Supervision, Project administration, Funding acquisition, and Writing—Review and Editing; M.A.: Conceptualisation, Supervision, Project administration, Funding acquisition, and Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was primarily supported by Southern Cross University, Australia (Funding Code: GLC 125001–2801).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors gratefully acknowledge the support provided by the Information and Communication Technology (ICT) Division, Ministry of Posts, Telecommunications and Information Technology, Government of the People's Republic of Bangladesh.

Conflicts of Interest: The authors declare no competing interests.

Appendix A

Table A1. Rainfall data sample for RICHMOND RIVER AT KYOGLE (Station: 203900).

#Timestamp	Value	Quality Code	Interpolation Type
2015-09-03T04:46:36.000 + 10:00	0.2	10	201
2015-09-03T04:48:30.000 + 10:00	0.2	10	201
2015-09-03T04:49:38.000 + 10:00	0.2	10	201
2015-09-03T04:52:43.000 + 10:00	0.2	10	201
2015-09-03T05:06:08.000 + 10:00	0.2	10	201
2015-09-03T05:07:23.000 + 10:00	0.2	10	201
2015-09-03T05:08:03.000 + 10:00	0.2	10	201
2015-09-03T05:11:30.000 + 10:00	0.2	10	201
2015-09-03T05:12:56.000 + 10:00	0.2	10	201
2015-09-03T05:14:29.000 + 10:00	0.2	10	201

Table A2. Water level data sample for RICHMOND RIVER AT KYOGLE (Station: 203900).

#Timestamp	Value	Quality Code	Interpolation Type
2015-09-01T00:00:00.000 + 10:00	0.055	10	102
2015-09-01T00:15:00.000 + 10:00	0.055	10	102
2015-09-01T00:30:00.000 + 10:00	0.055	10	102
2015-09-01T00:45:00.000 + 10:00	0.054	10	102
2015-09-01T01:00:00.000 + 10:00	0.054	10	102
2015-09-01T01:15:00.000 + 10:00	0.053	10	102
2015-09-01T01:30:00.000 + 10:00	0.053	10	102
2015-09-01T01:45:00.000 + 10:00	0.052	10	102
2015-09-01T02:00:00.000 + 10:00	0.051	10	102

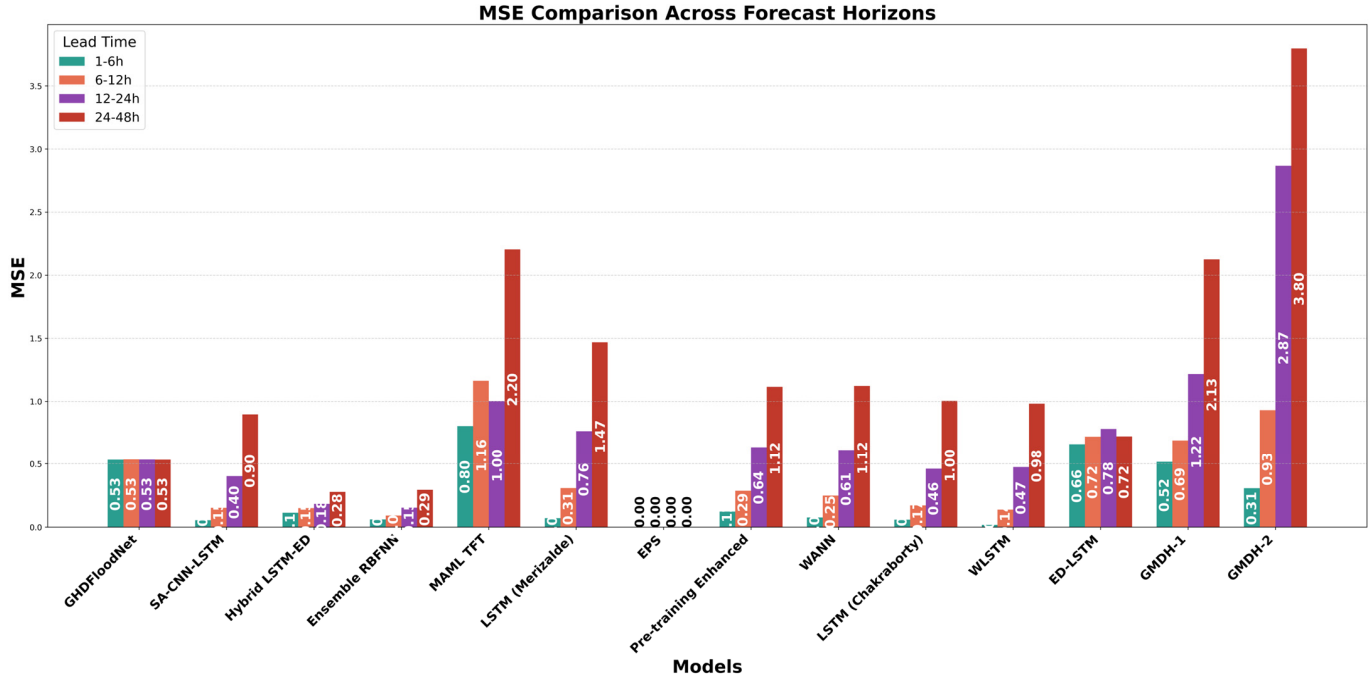
Table A3. Rainfall data sample for TWEED RIVER AT UKI (Station: 201900).

#Timestamp	Value	Quality Code	Interpolation Type
2022-09-09T09:30:00.000 + 10:00	0	10	201
2022-09-16T04:21:03.000 + 10:00	0.5	10	201
2022-09-16T04:33:41.000 + 10:00	0.5	10	201
2022-09-16T04:45:12.000 + 10:00	0.5	10	201
2022-09-16T04:55:58.000 + 10:00	0.5	10	201
2022-09-16T05:07:21.000 + 10:00	0.5	10	201
2022-09-16T06:01:53.000 + 10:00	0.5	10	201
2022-09-21T19:13:04.000 + 10:00	0.5	10	201
2022-09-21T19:22:24.000 + 10:00	0.5	10	201

Table A4. Water level data sample for TWEED RIVER AT UKI (Station: 201900).

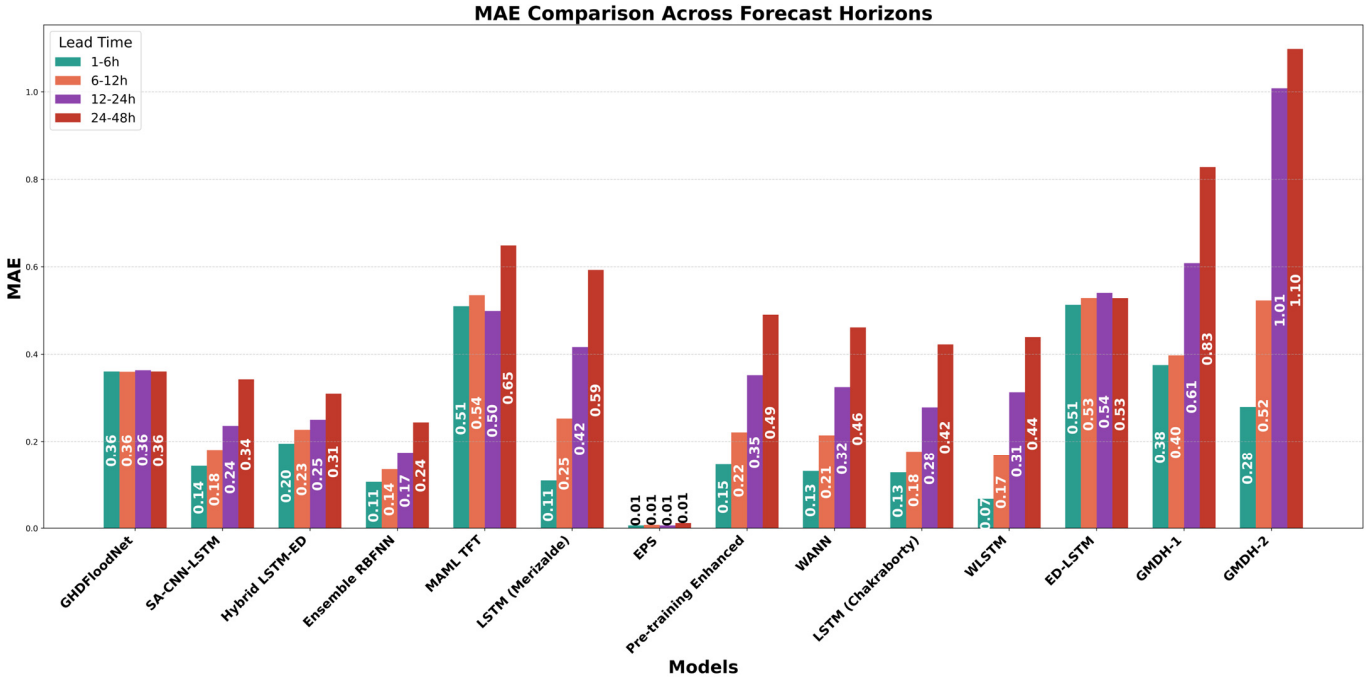
#Timestamp	Value	Quality Code	Interpolation Type
2015-09-01T00:00:00.000 + 10:00	1.127	140	102
2015-09-01T00:15:00.000 + 10:00	1.127	140	102
2015-09-01T00:30:00.000 + 10:00	1.127	140	102
2015-09-01T00:45:00.000 + 10:00	1.126	140	102
2015-09-01T01:00:00.000 + 10:00	1.127	140	102
2015-09-01T01:15:00.000 + 10:00	1.126	140	102
2015-09-01T01:30:00.000 + 10:00	1.126	140	102
2015-09-01T01:45:00.000 + 10:00	1.126	140	102
2015-09-01T02:00:00.000 + 10:00	1.126	140	102

Appendix B

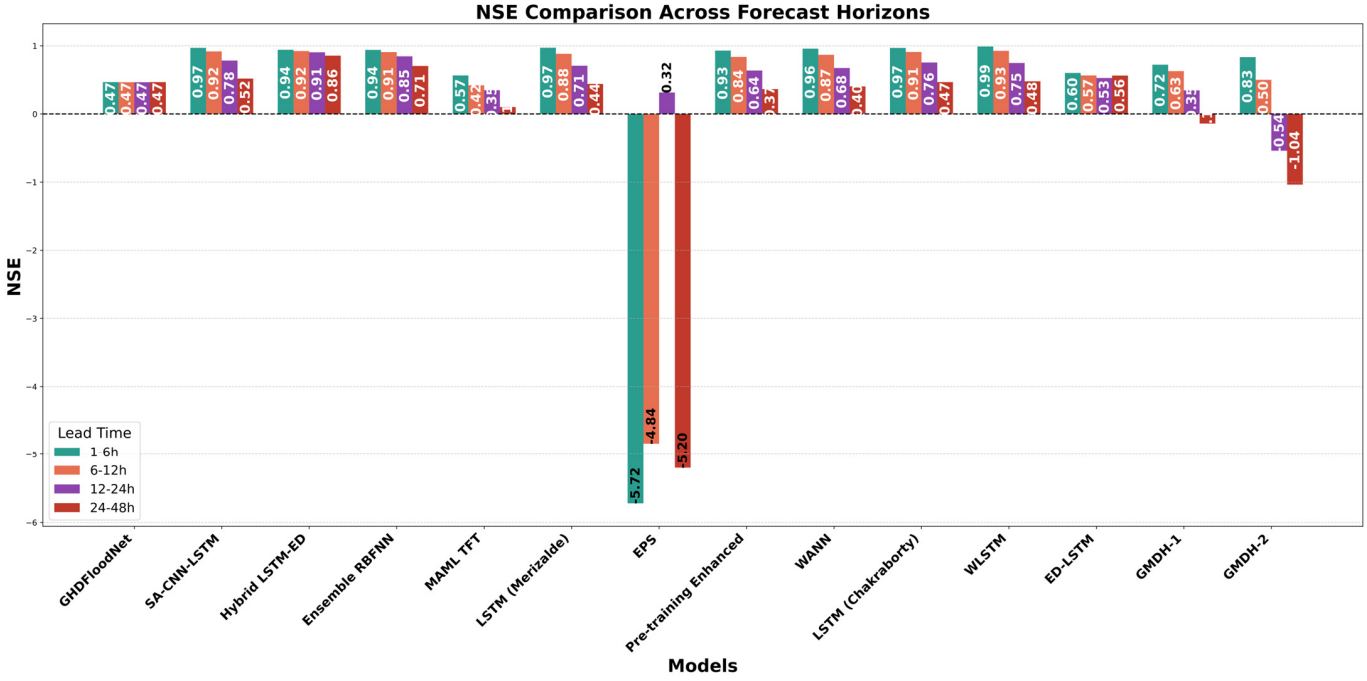


(**a**) MSE

Figure A1. *Cont.*

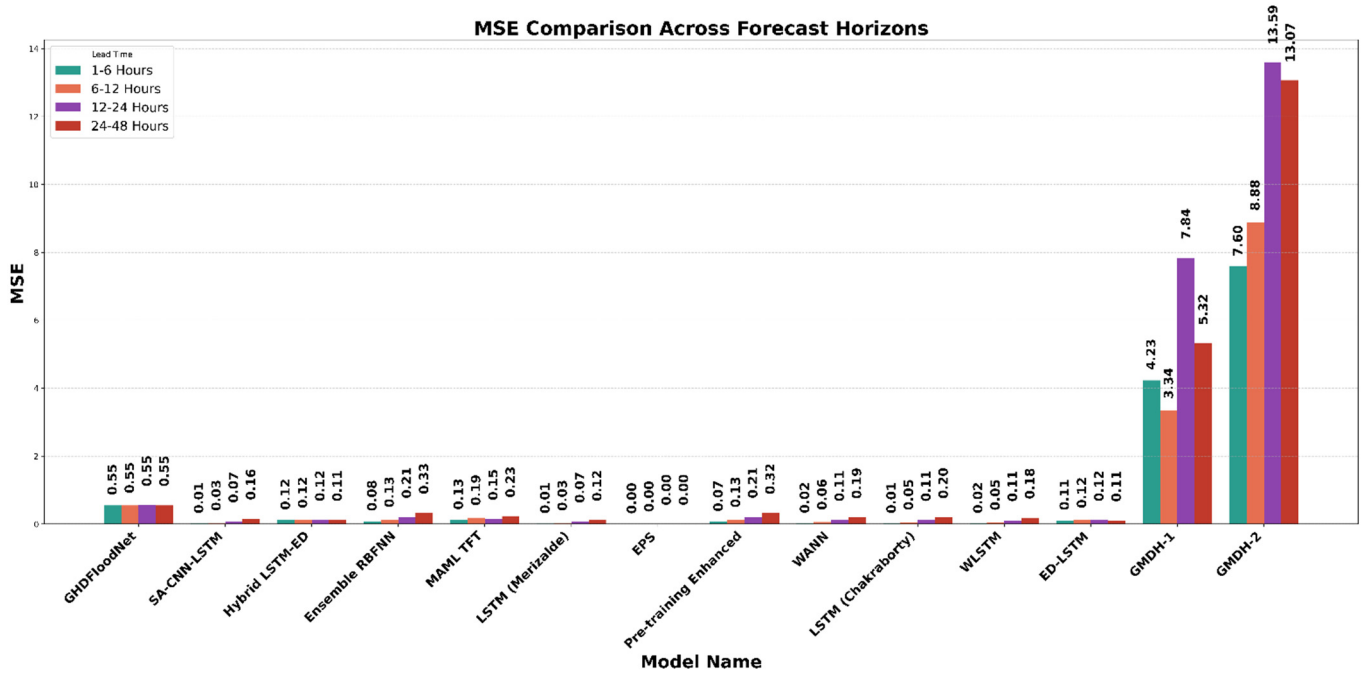


(**b**) MAE

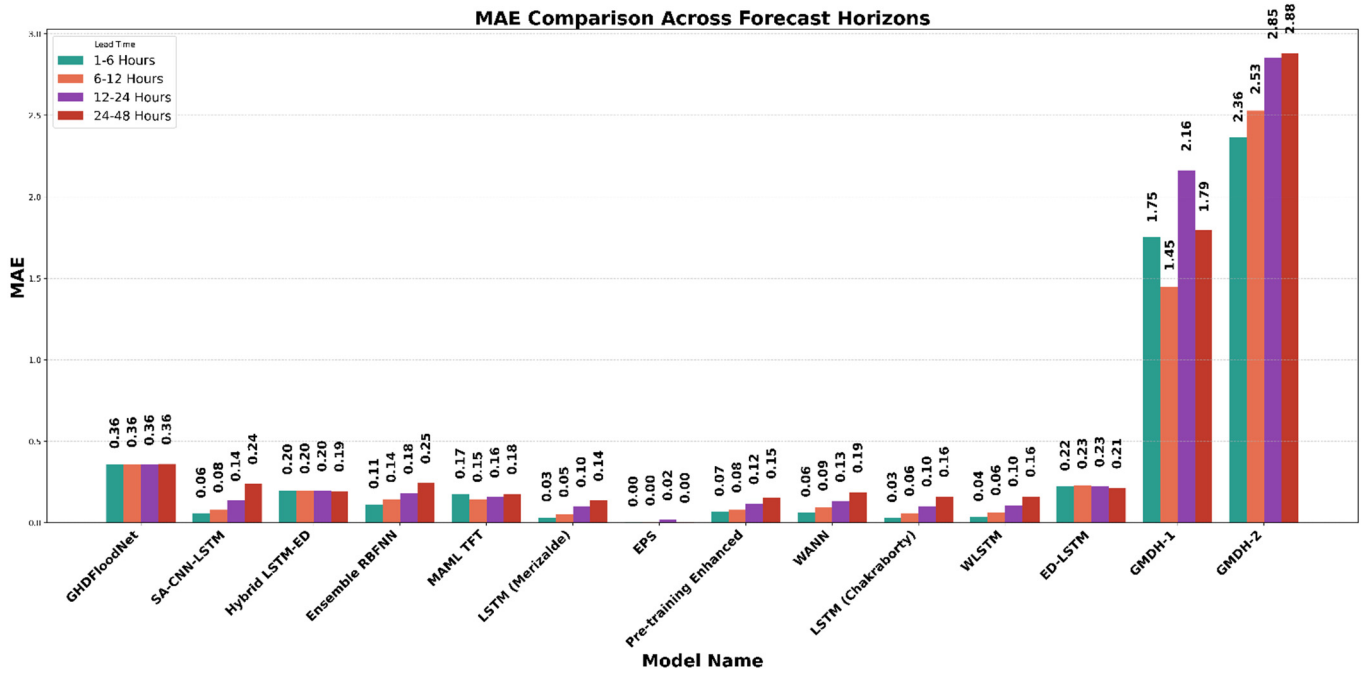


(**c**) NSE

Figure A1. GHDFloodNet vs. benchmark models for Richmond (**a**) MSE, (**b**) MAE, (**c**) NSE.

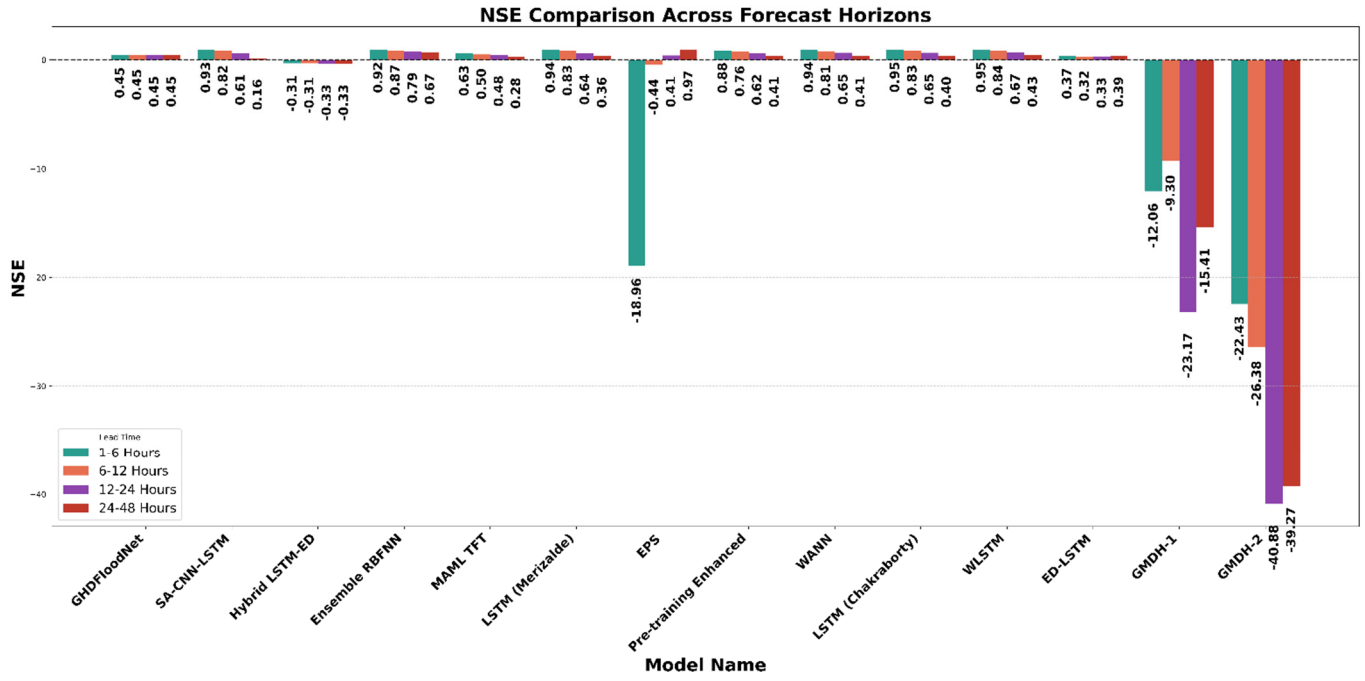


(**a**) MSE



(**b**) MAE

Figure A2. *Cont.*



(**c**) NSE

Figure A2. GHDFloodNet vs. benchmark models for Tweed (a) MSE, (**b**) MAE, (**c**) NSE.

References

1. Bukhari, S.A.S.; Shafi, I.; Ahmad, J.; Villar, S.G.; Villena, E.G.; Khurshaid, T.; Ashraf, I. Review of flood monitoring and prevention approaches: A data analytic perspective. *Nat. Hazards* **2025**, *121*, 5103–5128.
2. Chang, H.; Franczyk, J. Climate change, land-use change, and floods: Toward an integrated assessment. *Geogr. Compass* **2008**, *2*, 1549–1579. [CrossRef]
3. Asif, M.; Kuglitsch, M.M.; Pelivan, I.; Albano, R. Review and Intercomparison of Machine Learning Applications for Short-term Flood Forecasting. *Water Resour. Manag.* **2025**, *39*, 1971–1991. [CrossRef]
4. Leandro, J.; Althoff, I.; Fischer, S.; Mudersbach, C.; Lesny, K. Requirements for Flood-Driven Forecasting Systems for Small and Medium-Sized Catchments in Germany. *Water* **2025**, *17*, 3283. [CrossRef]
5. Mishra, A.; Mukherjee, S.; Merz, B.; Singh, V.P.; Wright, D.B.; Villarini, G.; Paul, S.; Kumar, D.N.; Khedun, C.P.; Niyogi, D. An overview of flood concepts, challenges, and future directions. *J. Hydrol. Eng.* **2022**, *27*, 03122001. [CrossRef]
6. Byaruhanga, N.; Kibirige, D.; Gokool, S.; Mkhonta, G. Evolution of flood prediction and forecasting models for flood early warning systems: A scoping review. *Water* **2024**, *16*, 1763. [CrossRef]
7. Mosavi, A.; Ozturk, P.; Chau, K.-w. Flood prediction using machine learning models: Literature review. *Water* **2018**, *10*, 1536. [CrossRef]
8. Kumar, V.; Sharma, K.V.; Caloiero, T.; Mehta, D.J.; Singh, K. Comprehensive overview of flood modeling approaches: A review of recent advances. *Hydrology* **2023**, *10*, 141. [CrossRef]
9. Panigrahi, M.; Sharma, A.; Poonia, V. Advancements and Challenges in Flood Modeling: A Comprehensive Review of Empirical to Physical-Based Approaches. In *Geoinformatics for Flood Risk Management*; CRC Press: Boca Raton, FL, USA, 2025; pp. 474–498.
10. Kow, P.-Y.; Liou, J.-Y.; Yang, M.-T.; Lee, M.-H.; Chang, L.-C.; Chang, F.-J. Advancing climate-resilient flood mitigation: Utilizing transformer-LSTM for water level forecasting at pumping stations. *Sci. Total Environ.* **2024**, *927*, 172246. [CrossRef] [PubMed]
11. Kumar, V.; Azamathulla, H.M.; Sharma, K.V.; Mehta, D.J.; Maharaj, K.T. The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood forecasting and management. *Sustainability* **2023**, *15*, 10543. [CrossRef]
12. Yuan, X.; Wood, E.F.; Ma, Z. A review on climate-model-based seasonal hydrologic forecasting: Physical understanding and system development. *Wiley Interdiscip. Rev. Water* **2015**, *2*, 523–536. [CrossRef]
13. Li, Y.; Grimaldi, S.; Walker, J.P.; Pauwels, V.R. Application of remote sensing data to constrain operational rainfall-driven flood forecasting: A review. *Remote Sens.* **2016**, *8*, 456. [CrossRef]
14. Davary, A.; Melville-Shreeve, P.; Brazier, R.; Chen, A.S. Harnessing Data for Flood Prediction: A Systematic Review of Forecasting Techniques and Information Sources. *J. Flood Risk Manag.* **2026**, *19*, e70183. [CrossRef]

15. Hossain, S.; Cloke, H.L.; Ficchì, A.; Gupta, H.; Speight, L.; Hassan, A.; Stephens, E.M. A decision-led evaluation approach for flood forecasting system developments: An application to the Global Flood Awareness System in Bangladesh. *J. Flood Risk Manag.* **2025**, *18*, e12959. [CrossRef]
16. Zhou, F.; Chen, Y.; Liu, J. Application of a new hybrid deep learning model that considers temporal and feature dependencies in rainfall–runoff simulation. *Remote Sens.* **2023**, *15*, 1395. [CrossRef]
17. Jahangir, M.S.; You, J.; Quilty, J. A quantile-based encoder-decoder framework for multi-step ahead runoff forecasting. *J. Hydrol.* **2023**, *619*, 129269. [CrossRef]
18. Faruq, A.; Hussein, S.F.M.; Marto, A.; Abdullah, S.S. Flood River Water Level Forecasting using Ensemble Machine Learning for Early Warning Systems. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, *1091*, 012041. [CrossRef]
19. Jiang, J.; Chen, C.; Lackinger, A.; Li, H.; Li, W.; Pei, Q.; Dustdar, S. MetaTrans-FSTSF: A Transformer-Based Meta-Learning Framework for Few-Shot Time Series Forecasting in Flood Prediction. *Remote Sens.* **2025**, *17*, 77.
20. Merizalde, M.J.; Muñoz, P.; Corzo, G.; Muñoz, D.F.; Samaniego, E.; Célleri, R. Integrating geographic data and the SCS-CN method with LSTM networks for enhanced runoff forecasting in a complex mountain basin. *Front. Water* **2023**, *5*, 1233899. [CrossRef]
21. Roohi, M.; Ghafouri, H.R.; Ashrafi, S.M. Developing an ensemble machine learning approach for enhancing flood damage assessment. *Int. J. Environ. Res.* **2024**, *18*, 90. [CrossRef]
22. Xia, Y.; Lu, J. Short-term flood prediction model based on pre-training enhancement. *Electronics* **2024**, *13*, 2203. [CrossRef]
23. Chakraborty, S.; Biswas, S. River discharge prediction using wavelet-based artificial neural network and long short-term memory models: A case study of Teesta River Basin, India. *Stoch. Environ. Res. Risk Assess.* **2023**, *37*, 3163–3184. [CrossRef]
24. Zhang, Y.; Ragettli, S.; Molnar, P.; Fink, O.; Peleg, N. Generalization of an Encoder-Decoder LSTM model for flood prediction in ungauged catchments. *J. Hydrol.* **2022**, *614*, 128577. [CrossRef]
25. Wang, M.; Rezaie-Balf, M.; Naganna, S.R.; Yaseen, Z.M. Sourcing CHIRPS precipitation data for streamflow forecasting using intrinsic time-scale decomposition based machine learning models. *Hydrol. Sci. J.* **2021**, *66*, 1437–1456. [CrossRef]
26. Yu, Q.; Tolson, B. How much historical data do we need? The role of data recency and training period length in LSTM-based rainfall-runoff modeling. *J. Hydrol.* **2026**, *668*, 135046. [CrossRef]
27. Shao, P.; Feng, J.; Lu, J.; Zhang, P.; Zou, C. Data-driven and knowledge-guided denoising diffusion model for flood forecasting. *Expert Syst. Appl.* **2024**, *244*, 122908. [CrossRef]
28. Iwana, B.K.; Uchida, S. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 3558–3565.
29. Rebuffi, S.-A.; Gowal, S.; Calian, D.A.; Stimberg, F.; Wiles, O.; Mann, T.A. Data augmentation can improve robustness. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29935–29948.
30. Roy, S.C.; Jahan, N.; Islam, M.; Al Ferdaus, A.; Fujisawa, K.; Adham, A. Comparative evaluation of machine learning models for extreme river water level forecasting in Bangladesh: Implications for flood and drought resilience. *Prog. Disaster Sci.* **2025**, *27*, 100449. [CrossRef]
31. Barros, M.; Fontenele, L.P.; Silva, M.S.; Rissi, T.; Cabrera, E.; Tannuri, E.A.; Gomi, E.S.; Barreira, R.A.; Costa, A.H.R. Embracing data irregularities in multivariate time series. *Appl. Soft Comput.* **2025**, *186*, 114039. [CrossRef]
32. Fei, N.; Gao, Y.; Lu, Z.; Xiang, T. Z-score normalization, hubness, and few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 142–151.
33. Demircioğlu, A. The effect of feature normalization methods in radiomics. *Insights Into Imaging* **2024**, *15*, 2. [CrossRef] [PubMed]
34. Kim, H.I.; Han, K.Y. Urban flood prediction using deep neural network with data augmentation. *Water* **2020**, *12*, 899. [CrossRef]
35. Victor, A.O.; Ali, M.I. Enhancing time series data predictions: A survey of augmentation techniques and model performances. In *Proceedings of the 2024 Australasian Computer Science Week. ACSW 2024: 2024 Australasian Computer Science Week, Sydney NSW Australia, 29 January–2 February 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–13.
36. Zhang, Y.; Ye, A.; Li, J.; Nguyen, P.; Analui, B.; Hsu, K.; Sorooshian, S. Improve streamflow simulations by combining machine learning pre-processing and post-processing. *J. Hydrol.* **2025**, *655*, 132904. [CrossRef]
37. Mouatadid, S.; Orenstein, P.; Flaspohler, G.; Cohen, J.; Oprescu, M.; Fraenkel, E.; Mackey, L. Adaptive bias correction for improved subseasonal forecasting. *Nat. Commun.* **2023**, *14*, 3482. [CrossRef] [PubMed]
38. Akrami, S.A.; El-Shafie, A.; Naseri, M.; Santos, C.A. Rainfall data analyzing using moving average (MA) model and wavelet multi-resolution intelligent model for noise evaluation to improve the forecasting accuracy. *Neural Comput. Appl.* **2014**, *25*, 1853–1861. [CrossRef]
39. Meresa, H.; Tischbein, B.; Mekonnen, T. Climate change impact on extreme precipitation and peak flood magnitude and frequency: Observations from CMIP6 and hydrological models. *Nat. Hazards* **2022**, *111*, 2649–2679. [CrossRef]

40. Bennett, B.; Leonard, M.; Deng, Y.; Westra, S. An empirical investigation into the effect of antecedent precipitation on flood volume. *J. Hydrol.* **2018**, *567*, 435–445. [CrossRef]
41. Mirus, B.B.; Loague, K. How runoff begins (and ends): Characterizing hydrologic response at the catchment scale. *Water Resour. Res.* **2013**, *49*, 2987–3006. [CrossRef]
42. Bahramian, K.; Nathan, R.; Western, A.W.; Ryu, D. Towards an ensemble-based short-term flood forecasting using an event-based flood model-incorporating catchment-average estimates of soil moisture. *J. Hydrol.* **2021**, *593*, 125828. [CrossRef]
43. Berthet, L.; Andréassian, V.; Perrin, C.; Javelle, P. How crucial is it to account for the antecedent moisture conditions in flood forecasting? Comparison of event-based and continuous approaches on 178 catchments. *Hydrol. Earth Syst. Sci.* **2009**, *13*, 819–831. [CrossRef]
44. Hakim, D.K.; Gernowo, R.; Nirwansyah, A.W. Flood prediction with time series data mining: Systematic review. *Nat. Hazards Res.* **2024**, *4*, 194–220. [CrossRef]
45. Sear, D.; Newson, M. Environmental change in river channels: A neglected element. Towards geomorphological typologies, standards and monitoring. *Sci. Total Environ.* **2003**, *310*, 17–23. [CrossRef] [PubMed]
46. Bentivoglio, R.; Isufi, E.; Jonkman, S.N.; Taormina, R. Deep learning methods for flood mapping: A review of existing applications and future research directions. *Hydrol. Earth Syst. Sci.* **2022**, *26*, 4345–4378. [CrossRef]
47. Ji, K.; Lee, J.D.; Liang, Y.; Poor, H.V. Convergence of meta-learning with task-specific adaptation over partial parameters. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11490–11500.
48. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169.
49. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
50. Fan, Y.; Tang, Q.; Guo, Y.; Wei, Y. BiLSTM-MLAM: A multi-scale time series prediction model for sensor data based on Bi-LSTM and local attention mechanisms. *Sensors* **2024**, *24*, 3962. [CrossRef] [PubMed]
51. Solís, M.; Gil-Gamboa, A.; Troncoso, A. Metalearning for improving time series forecasting based on deep learning: A water case study. *Results Eng.* **2025**, *28*, 107541. [CrossRef]
52. Zhang, L.; Liu, J.; Jin, B.; Wei, X. Meta-Learning Task Relations for Ensemble-Based Temporal Domain Generalization in Sensor Data Forecasting. *Sensors* **2025**, *25*, 4434. [CrossRef] [PubMed]
53. Acerbi, L.; Ma, W.J. Practical Bayesian optimization for model fitting with Bayesian adaptive direct search. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1834–1844.
54. Antoniou, A.; Edwards, H.; Storkey, A. How to train your MAML. *arXiv* **2018**, arXiv:1810.09502.
55. Li, Q.; Zhou, Y.; Zheng, Y.; Zhang, C.; Peng, T. Multi-Scale and Interpretable Daily Runoff Forecasting with IEWT and ModernTCN. *Water* **2026**, *18*, 183. [CrossRef]
56. Waqas, M.; Humphries, U.W. A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX* **2024**, *13*, 102946. [CrossRef] [PubMed]
57. Zhou, J.; Chen, L.; Hu, T.; Lu, H.; Shi, Y.; Chen, L. The comparative study of machine learning agent models in flood forecasting for tidal river reaches. *Sci. Rep.* **2025**, *15*, 19130. [CrossRef] [PubMed]
58. Sheikh, M.R.; Coulibaly, P. Review of recent developments in hydrologic forecast merging techniques. *Water* **2024**, *16*, 301. [CrossRef]
59. Cao, Q.; Zhang, H.; Zhu, F.; Hao, Z.; Yuan, F. Multi-step-ahead flood forecasting using an improved BiLSTM-S2S model. *J. Flood Risk Manag.* **2022**, *15*, e12827. [CrossRef]
60. Hydrosphere Consulting. *Richmond River Coastal Management Program–Protecting our Catchment and Estuary*; Hydrosphere Consulting: Ballina, Australia, 2021.
61. NSW Office of Water. *Water Sharing Plan Tweed River Area Unregulated and Alluvial Water Sources*; NSW Office of Water: Parramatta, Australia, 2010.
62. Bureau of Meteorology. Climate Data Online. Available online: http://www.bom.gov.au/climate/data/ (accessed on 30 September 2025).
63. Bureau of Meteorology. Water Data Online. Available online: http://www.bom.gov.au/waterdata/ (accessed on 30 September 2025).
64. Zhang, Y.; Pan, D.; Van Griensven, J.; Yang, S.X.; Gharabaghi, B. Intelligent flood forecasting and warning: A survey. *Intell. Robot.* **2023**, *3*, 190–212. [CrossRef]
65. Gan, M.; Lai, X.; Feng, Z.; Li, Z.; Cui, Y.; Pan, H.; Guo, Y.; Xu, C.; Xu, Z.; Yao, J. A data-driven framework for predicting lake levels with strong nonstationarity (Poyang Lake, Yangtze River basin). *J. Hydrol. Reg. Stud.* **2026**, *64*, 103260. [CrossRef]
66. Bontempi, G.; Ben Taieb, S.; Le Borgne, Y.-A. Machine learning strategies for time series forecasting. In *European Big Data Management and Analytics Summer School*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 62–77.
67. Cloke, H.L.; Pappenberger, F. Ensemble flood forecasting: A review. *J. Hydrol.* **2009**, *375*, 613–626. [CrossRef]

68. Chen, C.; Jiang, J.; Liao, Z.; Zhou, Y.; Wang, H.; Pei, Q. A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China. *J. Hydrol.* **2022**, *607*, 127535. [CrossRef]
69. Hussain, F.; Wu, R.-S.; Wang, J.-X. Comparative study of very short-term flood forecasting using physics-based numerical model and data-driven prediction model. *Nat. Hazards* **2021**, *107*, 249–284. [CrossRef]

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.