UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Robust Interpretable Hourly Runoff Forecasting Based on High-Performance Neural Networks

by Ziyu Sheng

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Shiping Wen

University of Technology Sydney
Faculty of Engineering and Information Technology

June 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Ziyu Sheng* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *Australian Artificial Intelligence Institute*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: June 2025

Abstract

Scientific development and management of river require collaborative efforts among multidisciplinary stakeholders and the integration of cross-disciplinary knowledge, wherein runoff forecasting plays a pivotal role. With the widespread adoption of data-driven deep learning models, particularly neural networks, in the hydrological domain, high-precision runoff forecasting, which was previously difficult to achieve using traditional physically based models, has now become feasible. However, current neural network-based runoff forecasting models still encounter various challenges in real-world applications. To address limitations in efficiency, accuracy, temporal robustness, and interpretability, this thesis progressively proposes a series of diversified and high-performance runoff forecasting frameworks.

First, to address the overemphasis on accuracy in existing runoff forecasting models, we develop a lightweight and robust framework based on Temporal Convolutional Network (TCN), which employs dilated and causal convolutions to expand the receptive field and prevent information leakage. An attention module is integrated to enhance accuracy with low computational cost, and an improved Snapshot ensemble strategy is used during training to boost robustness under extreme perturbations. Moreover, to overcome the limitations of mainstream neural networks, such as limited receptive fields and long-term dependencies modeling, we further propose two high-performance forecasting frameworks. These incorporate an enhanced ResNet with dual pathways and dense shortcuts to optimize information flow and benefit from deeper network structures.

Conventional attention mechanisms that focus on one single dimension are further extended to both temporal and spatial dimensions based on channel-dependence (CD) and channel-independence (CI) strategies. Bidirectional architecture and temporal shortcuts are also integrated to capture richer context and mitigate vanishing gradients in long sequences. Additionally, in response to the performance degradation observed in mainstream models during extended forecasting horizons, we propose a multi-lead-time forecasting framework grounded in the state space model (SSM), characterized by its dual attributes of convolution and recursion while still maintaining linear complexity. The effectiveness of the proposed framework is validated through quantitative evaluation, revealing strong temporal robustness across multiple forecasting horizons within the upcoming 24-hour. Notably, a model-specific local post-hoc explanation technique based on interpretable machine learning (IML) is also used to enhance the interpretability of the model's forecasting process.

In summary, this thesis proposes a set of runoff forecasting models that achieve state-of-the-art overall performance. Through progressive architectural enhancements across models, the structural limitations of existing approaches are effectively overcome, enabling highly accurate multi-lead-time forecasting of fine-grained hourly runoff sequences. These contributions provide robust and reliable solutions for stakeholders in hydrology.

Dedication

To myself . . .

ACKNOWLEDGMENTS

The four years of my Ph.D studies at the Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS), have been among the most unforgettable and rewarding experiences of my life. I would like to express my heartfelt gratitude to all those who have supported, accompanied, and encouraged me throughout this journey.

First and foremost, I would like to express my sincere gratitude to my principal supervisor, Prof. Shiping Wen, for his invaluable guidance and inspiration throughout my Ph.D journey. Academically, Professor Wen's profound expertise and extensive knowledge have continually guided me to scale new heights in research and encouraged me to explore uncharted research frontiers with confidence. Whenever I encountered confusion or setbacks, his patient instruction and unwavering encouragement always gave me the strength to move forward. Beyond academics, Prof. Wen's thoughtful care made me feel at home even while studying abroad, providing warmth akin to that of family. It has been a great honor to be mentored by such a distinguished and respected scholar. Prof. Wen is not only my supervisor but also a role model whom I deeply admire and strive to emulate. The knowledge and academic spirit he has imparted to me will remain lifelong treasures, for which I am eternally grateful.

I would also like to express my heartfelt thanks to my co-supervisor, Prof. Feng Liu, whose professional guidance and encouragement have been a tremendous source of motivation throughout my doctoral studies. From the initial design of my research framework to the final completion of this thesis, his patient advice and timely feedback have been

crucial to my academic growth. I am especially grateful to Dr. Yan Zheng, Dr. Yuhan Huang, Dr. Jiahao Xia, and Prof. Huiwei Wang for their invaluable support during my Ph.D journey. Their insightful perspectives and thoughtful suggestions have significantly broadened my academic horizons and helped me overcome numerous challenges. Their generous assistance helps me grow from a novice into a professional academic researcher. I am deeply grateful to all members of the Australian Artificial Intelligence Institute (AAII) for providing an exceptional research environment, state-of-the-art facilities, and unwavering academic support. Their hard work and dedication have been fundamental to my research journey. My sincere appreciation goes to all the experts and scholars who participated in my candidature assessments and thesis examination. Their valuable suggestions and constructive feedback have significantly strengthened the rigor and completeness of my research, for which I am profoundly grateful.

I am sincerely thankful to all my friends and research group members. In particular, I would like to extend my special thanks to Dr. Zhencheng Fan, Dr. Linhao Zhao, Dr. Boqian Li, Dr. Guangyang Tian, Dr. Wuzhida Bao, Dr. Shanshan Zhao, and Dr. Shengbo Wang for their generous help and encouragement in both my research and life. The four years of working and living alongside them have become a cherished memory in my life.

Finally, I would like to express my deepest gratitude to my parents, family, and my girlfriend. Their unconditional love, understanding, and support have always been the source of my strength and motivation. I am eternally grateful to them for helping me see a broader and more meaningful world.

List of Publications

1. **Z. Sheng**, S. Wen, Z.-K. Feng, K. Shi and T. Huang, "A novel residual gated recurrent unit framework for runoff forecasting," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12736-12748, 2023.

2. **Z. Sheng**, S. Wen, Z.-K. Feng, J. Gong, K. Shi, Z. Guo, Y. Yang and T. Huang, "A survey on data-driven runoff forecasting models based on neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1083-1097, 2023.

3. **Z. Sheng**, Z. An, H. Wang, G. Chen and K. Tian, "Residual LSTM based short-term load forecasting," *Applied Soft Computing*, vol. 144, pp. 110461, 2023.

4. **Z. Sheng**, Y. Cao, Y. Yang, Z.-K. Feng, K. Shi, T. Huang and S. Wen, "Residual temporal convolutional network with dual attention mechanism for multilead-time interpretable runoff forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

5. **Z. Sheng**, H. Wang, G. Chen, B. Zhou and J. Sun, "Convolutional residual network to short-term load forecasting," *Applied Intelligence*, vol. 51, no. 4, pp. 2485-2499, 2021.

6. C. Zhang, **Z. Sheng**, C. Zhang and S. Wen, "Multi-lead-time short-term runoff forecasting based on ensemble attention temporal convolutional network," *Expert Systems with Applications*, vol. 243, pp. 122935, 2024.

7. H. Wang, T. Liu, **Z. Sheng** and H. Li, "Explanatory subgraph attacks against graph neural networks," *Neural Networks*, vol. 172, pp. 106097, 2024.

8. J. Xia, W. Huang, M. Xu, J. Zhang, H. Zhang, **Z. Sheng** and D. Xu, "Unsupervised part discovery via dual representation alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

9. M. G. Al-Thani, **Z. Sheng**, Y. Cao, and Y. Yang, "Traffic Transformer: Transformer-based framework for temporal traffic accident prediction," *Manara-Qatar Research Repository*, 2024.

10. **Z. Sheng**, S. Wen, Z. Feng, K. Shi and T. Huang, "ResBi-Mamba Plus: Deep bidirectional Mamba with spatiotemporal attention for robust interpretable hourly runoff forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2025. [Under Review]

11. **Z. Sheng**, T. Huang, H. Zhu, and S. Wen, "Residual bidirectional temporal convolutional network with spatial-temporal attention for robust interpretable runoff forecasting," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 2025. [Under Review]

12. **Z. Sheng**, Y. Cao, Y. Yang, and S. Wen, "Residual bidirectional gated recurrent unit with spatiotemporal shortcuts and dual-path attention for robust interpretable runoff forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, 2025. [Under Review]

Table of Contents

List of Figures

Figure **Page**

List of Tables

List of Abbreviations

Adam Adaptive Moment Estimation

ALO Ant Lion Optimizer

ANN Artificial Neural Network

ARIMA Autoregressive Integrated Moving Average Model

Bi-LSTM Bidirectional LSTM

Bi-Mamba Bidirectional Mamba

BN Batch Normalization

BPTT Backpropagation Through Time

CBR Columbia Basin Research

CD Channel-Dependence

CI Channel-Independence

CNN Convolutional Neural Network

CRBM Convolutional Restricted Boltzmann Machine

CV Computer Vision

DAM Dual Attention Mechanism

DART Data Access in Real Time

DBN Deep Belief Network

EA-TCN Ensemble Attention Temporal Convolutional Network

EFAS European Flood Awarness System

EMD/EEMD Empirical Mode Decomposition/Ensemble Empirical Mode Decomposition

FC Fully-Connected Layer

FCN Fully-Convolutional Network

GAP Global Average Pooling

GNN Graph Neural Network

GP Gaussian Process

GPR Gaussian Process Regression

GPT Generative Pre-trained Transformer

GRU Gated Recurrent Unit

IML Interpretable Machine Learning

LLM Large-Scale Language Model

LSTM Long Short-Term Memory

LTI Linear Time-Invariant

LUBE Lower and Upper Bound Estimation

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MLP multi-layer perceptron

MSE Mean Squared Error

MTNet Memory Time Series Network

NLP Natural Language Processing

NSE Nash-Sutcliffe Coefficient of Efficiency

PSO Particle Swarm Optimization

QPE Quantitative Precipitation Estimation

R^2 Coefficient of Determination

ReLU Rectified Linear Unit

ResNet Residual Network

RNN Recurrent Neural Network

SENet Squeeze-and-Excitation Network

SSA Singular Spectrum Analysis

SSD State Space Duality

SSM State Space Model

SOTA State-Of-The-Art

SVM Support Vector Machine

SVR Support Vector Regression

STResBiGRU SpatioTemporal Residual Bidirectional Gated Recurrent Unit

SWAT Soil and Water Assessment Tool

TCN Temporal Convolutional Network

TFT Temporal Fusion Transformer

VGG Visual Geometry Group

ZOH Zero-Order Hold

CHAPTER 1

INTRODUCTION

1.1 Background and Motivations

River runoff is an important part of the natural water cycle, supplying essential water for drinking, irrigation, and industrial use [1]. Its potential energy can also be harnessed by hydroelectric facilities to generate substantial electricity [2], supporting economic and social development. River runoff also contributes to the stability of basin ecosystems and is a key factor in environmental protection and ecological management efforts. Consequently, the scientific development and management of river runoff serve as the cornerstone for harmonious coexistence between humanity and nature, as well as for the sustainable socio-economic development [3]. This necessitates collaborative efforts among stakeholders across multiple domains and the integration of interdisciplinary knowledge, with runoff forecasting playing a pivotal role in this context [4]. Leveraging cutting-edge techniques such as expert knowledge, statistical models, and machine learning algorithms, runoff forecasting utilizes historical data to forecast river runoff over specific future timeframes, thereby offering data support and decision-making guidance for the

rational planning and management of water resources [5]. As an important branch of time series forecasting, runoff forecasting involves the integrated assessment of more complex hydrometeorological data compared to forecasting tasks in domains such as finance, transportation, and power systems. Moreover, it imposes stricter requirements on both forecasting timeliness and model accuracy.

In order to adapt to different specific tasks and scenarios, runoff forecasting is further classified into hourly forecasting [6], daily forecasting [7], monthly forecasting and annual forecasting according to different time granularity [8, 9]. Monthly and annual runoff forecasting are mainly used for climate change studies and long-term development planning of river basins, while daily runoff forecasting is used for the daily scheduling and maintenance of hydropower plants. Among all subcategories of runoff forecasting, hourly runoff forecasting focuses on the shortest time scale, characterized by significant randomness and uncertainty due to the influence of numerous external factors [10]. This makes it a non-stationary time series and the most challenging to forecast [11, 12]. Figure 1.1 presents the hourly river discharge curve for the Columbia River in the United States over the entire year of 2019. It can be observed that the runoff exhibits a pronounced seasonal and non-stationary trend. However, hourly runoff forecasting is crucial for providing timely warnings and responses to imminent hydrological events, thus holding significant research value [13–15].

Runoff forecasting has undergone decades of research and refinement since its inception, gradually evolving into two main branches: process-based runoff forecasting models and data-driven runoff forecasting models. In the early stages of runoff forecasting development, due to limitations in computational power and incomplete theoretical frameworks, process-based forecasting was a more commonly used approach [16]. It primarily involves simulating the complex physical processes of the natural water cycle, such as precipitation, evaporation, and groundwater flow, to forecast runoff. The

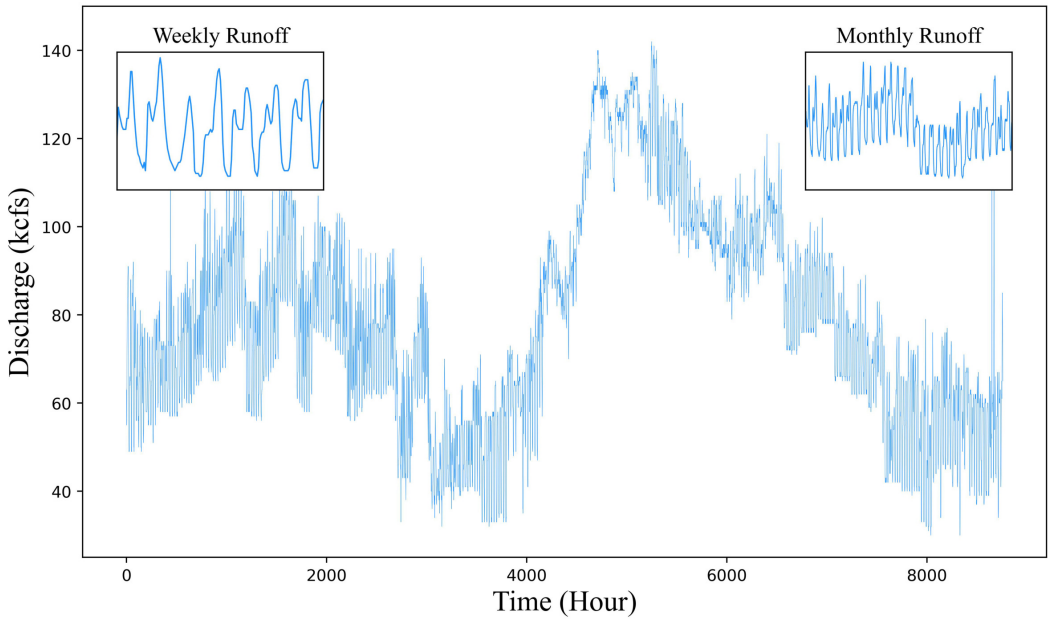


Figure 1.1: Hourly discharge curve of the Columbia River in 2019. The subfigure in the top left corner presents an enlarged view of the weekly runoff curve, while the subfigure in the top right corner presents the enlarged monthly runoff curve.

advantage of these models lies in their good interpretability and the ability to model and forecast multiple different hydrological processes simultaneously, maintaining a certain level of application in the current hydrological field. For instance, the United States Department of Agriculture (USDA) developed the Soil and Water Assessment Tool (SWAT) model [17], while the Danish Hydraulic Institute (DHI) proposed the famous MIKE SHE model [18]. Both models are representative distributed physical hydrological models that can simulate different components of the water cycle through corresponding independent sub-modules and model the entire complex water cycle process through the integration of different modules. However, process-based runoff forecasting models have revealed numerous inherent flaws in practical applications. They are complex, require extensive detailed physical data, and their construction and use heavily depend on the researchers' expertise and experience in the hydrological field, making them challenging to apply broadly and limiting their generalization capability [19].

The widespread adoption of automated monitoring systems in hydrological systems leads to the establishment of monitoring stations on many major rivers around the world. These stations accumulate a large amount of hydrological data. The increase in computational power, together with this wealth of data, shifts the focus in runoff forecasting from process-based models to data-driven models [20]. Unlike process-based models, data-driven models depend on historical data for modeling and learning. They forecast future runoff by understanding the relationship between input features and outputs, without relying on complex physical processes. The main advantage of data-driven models is their independence from specialized knowledge and complex parameters, facilitating easier construction and training with machine learning frameworks and data science toolkits. This ease of use enables their flexible adaptation to new regional data and rapid deployment. As the most widely used and representative data-driven model, artificial neural network (ANN) has gradually gained favor in runoff forecasting [21, 22]. Compared to traditional linear models, ANN possesses a powerful nonlinear fitting capability, enabling it to flexibly learn the intricate interactions among various external factors in hydrological processes, making it more suitable for handling non-stationary runoff sequences. Over the years, ANN has evolved into two main streams: the convolutional neural network (CNN) [23] and the recurrent neural network (RNN) [24]. CNN is known for its strong multi-level abstract feature learning ability and can reduce the number of parameters through sparse connections and weight sharing, which has led to its widespread application in the field of computer vision (CV) [25, 26]. The RNN model, exemplified by long short-term memory network (LSTM) [27, 28], has capabilities beyond traditional neural networks in processing sequence inputs. It can selectively retain and forget input information through gating units, giving the network a flexible "memory" for capturing temporal dependencies in sequence inputs, making it a milestone model in natural language processing (NLP) [29, 30]. With the rapid penetration of

neural networks across various fields, CNN and LSTM have also been extensively used in runoff forecasting.

Although neural networks have demonstrated outstanding performance in runoff forecasting, their practical application has also revealed several problems that significantly constrain the overall model performance. These problems mainly manifest in three aspects: accuracy, temporal robustness, and interpretability. 1) While runoff forecasting models based on neural networks have far surpassed early traditional forecasting models in terms of accuracy, the hydrological field's demands for forecasting precision continue to rise. Existing mainstream models struggle to meet the industry's needs for accuracy-oriented tasks, fundamentally due to a bottleneck in the models' feature learning capabilities. Although LSTM has a superior feature extraction capability, its gating units bring a huge number of parameters, making it difficult to be stacked multiple layers to learn more higher-level abstract features [31]. On the other hand, CNN with fewer parameters still faces limitations in temporal forecasting ability due to its smaller receptive field and issues with information leakage. 2) Short-term runoff forecasting models generally suffer from poor temporal robustness, where the model's accuracy is high for short lead times but dramatically declines as the lead time increases (runoff forecasting must be performed before a certain period of time, and this kind of advance is called lead time [32]). This is primarily because forecasting runoff over longer periods often requires inputting information from more time steps, and the increased amount of information makes it difficult for the model to fully capture long-term dependencies in long sequence inputs, restricting the model's multi-lead-time forecasting capability. 3) Current research on runoff forecasting models is mainly performance-oriented, often overlooking model interpretability. This has resulted in a lack of trust from stakeholders in high-risk areas towards these high-accuracy but low-transparency models. To effectively address the three common challenges in runoff forecasting outlined above, this

thesis proposes a comprehensive research framework aimed at providing high-accuracy, robust, and interpretable models based on cutting-edge neural network techniques for stakeholders in the hydrological field.

1.2 Research Questions and Objectives

1.2.1 Research Questions

The application of neural networks in runoff forecasting has been increasingly deepened, demonstrating significantly superior performance compared to traditional models. However, this advancement has also revealed a number of pressing issues that remain unresolved. Specifically, this study focuses on the following research questions:

RESEARCH QUESTION 1 (RQ1): *How to achieve an optimal balance between accuracy, efficiency, and robustness in next-generation runoff forecasting models?*

Most of the research work on runoff forecasting is accuracy-oriented, but the blind pursuit of accuracy can lead to a reduction in robustness and efficiency. Zhang et al. clearly point out that multiple well-trained neural networks, including the famous VGG, AlexNet and ResNet, often sacrifice robustness against adversarial attacks in the pursuit of higher accuracy, and thus misclassify samples with added noise [33] . In addition, although neural networks such as LSTM have better performance in processing time series forecasting, their gating units can significantly increase the number of parameters. And the cell of each LSTM is heavily dependent on the output of the previous time step, so LSTM also cannot be processed in parallel. Short-term runoff forecasting requires a high level of rapid response capability, which requires timely warning and decision support for unexpected precipitation and hydrological events within a short period of time, so the efficiency of the model will directly affect whether it can be used in practice.

RESEARCH QUESTION 2 (RQ2): *How to further overcome the performance bottlenecks of existing runoff forecasting models?*

Most current runoff forecasting models are built upon neural networks based on recurrent architectures, such as RNN, LSTM, and gated recurrent unit (GRU). Compared with traditional models, these networks exhibit a certain degree of memory capability, allowing them to selectively retain and forget information in the temporal dimension. However, neural networks based on recurrent architectures still suffer from several inherent limitations. Specifically, as the loss is backpropagated along the time dimension, the presence of long input sequences with numerous time steps may lead to vanishing/-exploding gradient problems, which hinder effective weight updates during training. Although LSTM and GRU introduce gating units to regulate information flow, the effectiveness of these gates diminishes as the sequence length increases, reducing the model's ability to capture long-term dependencies. The inherent parametric complexity of gating units also limits the scalability of network depth, making it difficult to stack multiple layers for extracting higher-level abstract features essential for accurate forecasting. These architectural bottlenecks significantly constrain the performance of existing models. Therefore, designing novel architectures to overcome these limitations is essential for the future advancement of runoff forecasting.

RESEARCH QUESTION 3 (RQ3): *How to develop temporally robust architectures for runoff forecasting models so that they can consistently maintain stable performance across multiple forecasting horizons of varying lengths?*

With the continuous advancement of time series forecasting techniques and the widespread application of neural network-based data-driven models in runoff forecasting, mainstream models have demonstrated significant improvements over traditional methods for specific forecasting horizons. However, current hourly runoff forecasting is undergoing a paradigm shift from single-step to multi-step forecasting, and stakeholders

are placing increasing demands on the models' multi-lead-time forecasting performance. Although most existing models can achieve accurate predictions over short horizons (e.g., 1-4 hours ahead), their accuracy deteriorates dramatically for longer horizons exceeding 12 hours, with accuracy declines often surpassing 50%. This phenomenon highlights a common limitation in existing models: the lack of temporal robustness. To enable robust multi-lead-time runoff forecasting, substantial architectural enhancements are required to empower models with the capacity to capture the rich contextual information embedded in exponentially growing input sequences. Addressing this challenge constitutes a critical focus of this thesis.

RESEARCH QUESTION 4 (RQ4): *How to ensure that the forecasting process of the proposed runoff forecasting model remains interpretable to a certain extent for stakeholders?*

Neural networks are often regarded as black box models, mainly because they transform the input non-linearly through the activation function. Although they can fit the data better than traditional linear models, the state inside the network is difficult to understand well. In addition, neural networks learn adaptively by adjusting weights through optimisation algorithms such as gradient descent, and we cannot understand the meaning of each weight, and therefore we cannot further explain the forecasting process of the model. In some specific high-risk areas, people not only need the model to make accurate forecast, but also need the model to be able to explain how the forecast was made. Therefore, the next-generation runoff forecasting architecture should not only demonstrate superior performance but also ensure that its forecasting process is reasonably interpretable. Models with higher interpretability are more likely to gain the trust of stakeholders and have greater potential for real-world deployment.

1.2.2 Research Objectives

To answer these research questions, we set up the corresponding Research Objectives (RO) as follows:

RESEARCH OBJECTIVE 1 (RO1): *To develop an efficient runoff forecasting model based on lightweight neural network architecture and ensemble learning strategy.* (Aims to answer RQ1)

To address the limitations of existing runoff forecasting models that overly prioritize accuracy at the expense of efficiency and robustness, we propose an innovative lightweight and robust forecasting framework. This framework employs the ensemble strategy in place of conventional model training approaches to enhance model robustness. Multiple homogeneous or heterogeneous weak learners are integrated either sequentially or in parallel, depending on specific ensemble strategies. This integration reduces the risks of overfitting and model bias, enhances stability and data diversity, and ultimately improves robustness. In terms of efficiency, we replace the inefficient recurrent architecture with lightweight neural networks that support parallel computation. By reducing the number of model parameters and leveraging parallelism, we significantly improve both training and inference efficiency, enabling short-term runoff forecasting models to respond quickly to hydrological changes. The proposed framework maintains accuracy while avoiding compromises in efficiency and robustness, demonstrating outstanding comprehensive performance.

RESEARCH OBJECTIVE 2 (RO2): *To integrate advanced machine learning methods and construct novel deep high-performance neural network architectures that overcome the performance bottlenecks of existing runoff forecasting models.* (Aims to answer RQ2)

As outlined in RQ2, the widely used RNN and its variants exhibit inherent limita-

tions that can significantly impair model accuracy. To address this problem, this study proposes several high-performance forecasting frameworks that collectively overcome current performance bottlenecks and improve predictive accuracy from multiple perspectives. We deeply integrate lightweight and plug-and-play attention modules into the models. These modules adaptively recalibrate the importance of features along both temporal and spatial dimensions, thereby enhancing the model's ability to learn from long input sequences. In addition, we migrate and modify several novel backbone architectures that have demonstrated outstanding performance in CV and NLP tasks, such as ResNet which effectively mitigates performance degradation in deep networks. The proposed frameworks significantly exceed mainstream models in both depth and width, enabling the capture of high-level abstract features and comprehensive global modeling of multivariate inputs.

RESEARCH OBJECTIVE 3 (RO3): *To develop multi-lead-time runoff forecasting models with high temporal robustness based on bidirectional architecture and dual-path multidimensional attention mechanism.* (Aims to answer RQ3)

Existing runoff forecasting models generally maintain robust performance only over short forecasting horizons. This limitation is mainly due to the fact that long-term forecasting depends on multivariate input sequences that are several times longer than the lead time. These extended sequences contain richer contextual information, yet current mainstream models are unable to effectively capture the significantly increased long-term dependencies because of inherent limitations such as vanishing gradient, limited receptive field, and quadratic computational complexity. To enhance temporal robustness, we propose a bidirectional architecture embedded with the multidimensional spatiotemporal attention mechanism. Considering the widespread use of neural networks based on recurrent architectures in runoff forecasting, we further introduce temporal residual connections that enable intact information transfer across cells, thereby mitigating the

degradation of modeling capacity over long sequences. In addition, we improve the novel Mamba architecture, which features linear complexity, and apply it for the first time to hourly runoff forecasting. The proposed methodology effectively addresses the limitations of current models in handling long input sequences, demonstrating significant improvements in multi-lead-time prediction performance.

RESEARCH OBJECTIVE 4 (RO4): *To enhance the transparency of the forecasting process by deeply integrating interpretable machine learning methods into runoff forecasting models.* (Aims to answer RQ4)

The black-box nature of neural network models makes it difficult to mathematically characterize the nonlinear mapping from input to output, and it remains challenging to provide global interpretability for models with complex architectures. To address the lack of interpretability that is often overlooked in mainstream runoff forecasting models, we incorporate the theories and methodologies of interpretable machine learning (IML) into the proposed forecasting framework. At the module level, we adopt model-specific local post-hoc explanation techniques to intuitively visualize the model's forecasting process. Enhancing interpretability not only strengthens stakeholders' trust and acceptance of the model, but also enables the timely identification and correction of flawed patterns and biases that may be learned during training process.

1.3 Research Contributions

This study provides a comprehensive review and analysis of the multiple challenges faced by current neural network-based runoff forecasting models. In response to these challenges, we systematically propose a series of optimization strategies and architectural enhancements aimed at addressing their limitations. The main contributions of this research can be summarized as follows:

A Novel Runoff Forecasting Model with Integrated Accuracy, Efficiency, and Robustness

- We propose an enhanced temporal convolutional network (TCN) for short-term runoff forecasting, achieving higher accuracy and efficiency than recurrent models.
- A lightweight temporal attention module is embedded to highlight key time steps with minimal overhead.
- The Snapshot ensemble boosts accuracy and robustness without extra parameters or training cost.

A high-accuracy runoff forecasting model with dense shortcut connections and multidimensional spatiotemporal attention mechanism

- We combine TCN with an improved ResNet architecture featuring dual residual pathways and dense shortcut connections, enhancing accuracy through optimized information flow and higher-level abstract feature learning.
- A plug-and-play dual-path attention mechanism is proposed to model the importance of time steps and features in parallel, enabling explicit recalibration of critical information in the feature maps.
- The output layer of the model is further embedded with a global attention module, which works in conjunction with the dual-path attention module to enhance the model's ability to perceive global information.

An interpretable runoff forecasting model with a bidirectional architecture and temporal residual connections

- This study proposes our redesigned BiGRU to process input sequences bidirectionally and fuses the information efficiently through a spatiotemporal attention module.
- In addition to dense spatial shorcuts, unique temporal shortcuts are established between GRU cells to enhance efficient information flow in the temporal dimension.
- IML-based heatmap visualization is used to provide intuitive and reasonable explanations for the forecasting process of local model modules.

Robust multi-lead-time runoff forecasting using emerging Mamba architecture

- The Mamba architecture, which combines both convolutional and recurrent properties while maintaining linear complexity, is introduced for the first time in hourly runoff forecasting.
- This study breaks the depth bottleneck of existing Mamba models and constructs a deep bidirectional Mamba to improve long-sequence modeling capability.
- The proposed model achieves superior multi-lead-time forecasting performance and maintains temporal robustness over extended horizons up to 48 hours.

1.4 Research Significance

The theoretical and practical significance of this thesis is summarized as follows:

Theoretical Significance: This thesis conducts a systematic review of widely used neural network-based runoff forecasting models and summarizes their key limitations, particularly the lack of attention to model efficiency, temporal robustness, and interpretability in existing research. These challenges define the motivation for this work

and guide the construction of a progressive theoretical framework, in which a series of models are proposed to fill corresponding gaps in the field. The first contribution is the introduction of a novel runoff forecasting model that integrates lightweight neural networks with ensemble learning and attention mechanisms, aiming to achieve a balance among accuracy, efficiency, and robustness, and providing new perspectives for performance-driven hydrological modeling. Furthermore, this study proposes an innovative spatiotemporal attention mechanism designed for multivariate inputs, extending traditional single-dimension attention toward a more comprehensive modeling of complex dependencies in temporal and spatial dimensions. To address the sharp decline in model performance over long forecasting horizons, a new architecture is developed by combining the dual-path densely connected ResNet with the bidirectional structure. This model achieves superior performance across various lead times and offers a leading theoretical solution for stakeholders with high demands for temporally robust runoff forecasting.

Practical Significance: The research outputs of this thesis demonstrate significant practical value and broad potential for future development. The proposed high-performance multi-lead-time runoff forecasting frameworks establish new benchmarks for fine-grained hourly runoff forecasting, effectively addressing critical challenges observed during the deployment of existing models. Each model introduced in this work has been rigorously evaluated through ablation and comparative experiments on real-world datasets, validating both the effectiveness of individual components and the superiority of the overall architecture. Moreover, the methodology is highly modular, with low coupling and model-agnostic components that can be efficiently transferred and integrated into most existing mainstream runoff forecasting models. The intuitive visualization of the prediction process further improves local interpretability, thereby enhancing stakeholder trust and facilitating real-world adoption. The outcomes of this research can be rapidly

generalized and deployed in real-world applications that require timely and accurate fine-grained runoff forecasting, including flood early warning systems and hydropower dispatch.

1.5 Thesis Structure

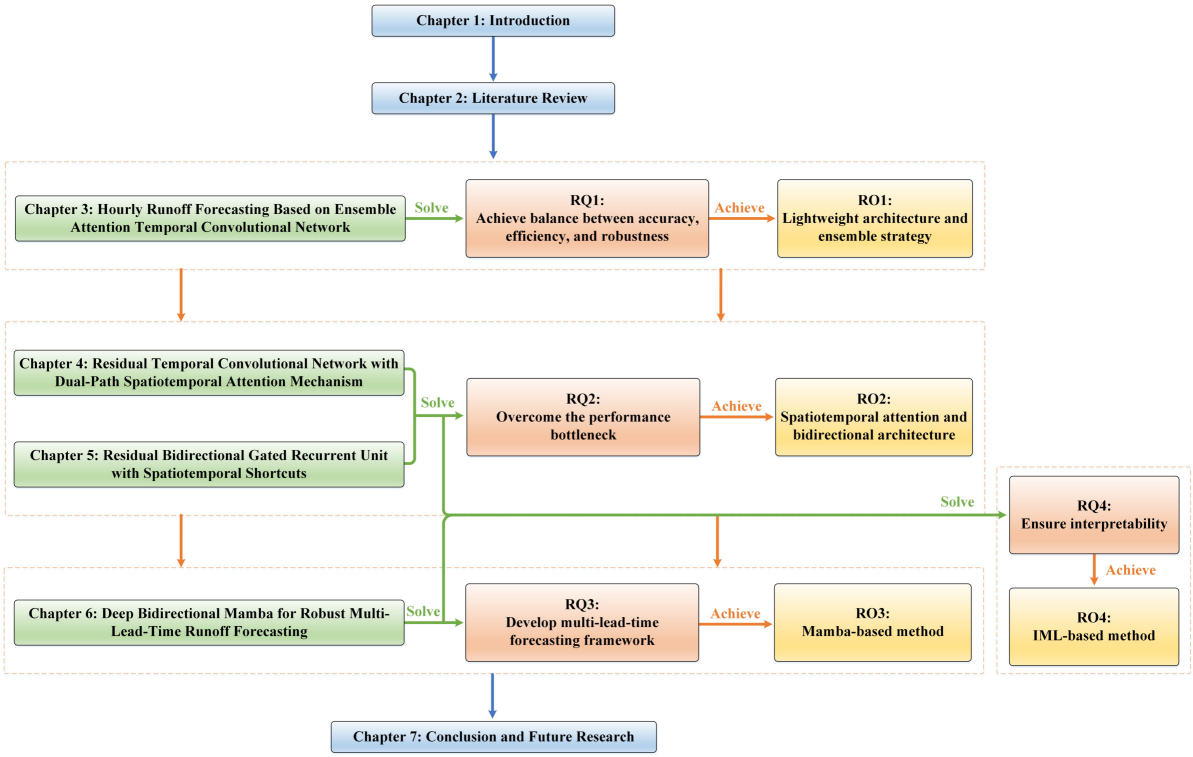


Figure 1.2: Thesis Structure.

The structure of the thesis is shown in Figure 1.2 and the chapters are organized as follows:

- **CHAPTER 2**: This chapter provides a comprehensive review of the literature relevant to this study. It begins by introducing the concept of hourly runoff forecasting and tracing its development over time. The focus then shifts to the review of existing models applied to runoff forecasting, categorized by different types of

neural network architectures. Each model's strengths and limitations are analyzed in detail, which not only highlights the current research status of runoff forecasting models in hydrology but also reveals various practical challenges encountered by mainstream models. These findings establish a solid foundation for subsequent optimization and improvement in this thesis.

- **CHAPTER 3**: This chapter presents a novel framework for short-term runoff forecasting, termed the Ensemble Attention Temporal Convolutional Network (EA-TCN). Its primary contribution is the seamless combination of TCN, lightweight attention mechanism, and ensemble learning strategy, which together boost accuracy, efficiency, and robustness. TCN serves as the foundation of the framework, offering an architecture with shared parameters and parallel computation that greatly improves computational speed. The use of causal and dilated convolutions enables effective modeling of long-term dependencies in time series data. The incorporated attention mechanism enhances the ability to capture intricate temporal patterns with minimal computational overhead. Furthermore, by employing the Snapshot ensemble technique, the framework simulates the training of multiple models within one single learning process, thereby achieving further gains in forecasting accuracy and robustness.

- **CHAPTER 4**: This chapter demonstrates an improved hybrid model, ResTCN-DAM, based on the EA-TCN framework. The model integrates the strengths of TCN, ResNet, and dual attention mechanism (DAM). It leverages TCN's efficient parallel processing of temporal data and combines it with an enhanced ResNet featuring dual pathways and dense shortcuts, allowing lightweight TCN layers to be deeply stacked for high-level feature extraction. The DAM module effectively captures interdependencies across both temporal and feature (spatial) dimensions in multivariate inputs, highlighting critical time steps and features while suppress-

ing irrelevant ones with minimal computational cost. Through the deep integration of these modules, the model significantly enhances forecasting performance from multiple perspectives and successfully overcomes the performance bottlenecks (RO2). In addition, model interpretability is improved by employing IML-based heatmap visualizations to reveal the internal forecasting process (RO4).

- **CHAPTER 5**: This chapter targets the most widely used recurrent-architecture-based neural networks in runoff forecasting and proposes a novel interpretable deep hourly forecasting framework: the SpatioTemporal Residual Bidirectional Gated Recurrent Unit (STResBiGRU). This framework fundamentally relies on a significantly enhanced BiGRU network, which captures rich hidden dependencies in long sequences by traversing the input in both forward and backward directions. A key innovation lies in the introduction of unique temporal shortcut connections between GRU cells, enabling more efficient information propagation across time steps. Benefiting from the lightweight structure of GRU, the model can also be deeply integrated with the ResNet with dense spatial shortcuts, which significantly increases network depth. The incorporation of spatiotemporal residual connections further improves the mode of information transmission, mitigates the vanishing gradient problem caused by deep networks and long sequences, and enhances feature reuse. In addition, during the fusion of bidirectional information flows in the BiGRU, a DPA module is embedded after each branch to explicitly model the interdependencies among features across both temporal and spatial dimensions in the forward and backward GRU outputs. Compared to conventional LSTM/GRU, the deeply enhanced STResBiGRU demonstrates substantially improved forecasting performance (RO2), and the forecasting processes of both the forward and backward BiGRU branches are further visualized in detail using IML techniques (RO4).

- **CHAPTER 6**: This chapter proposes an hourly runoff forecasting framework, ResBi-Mamba Plus, based on the latest state space model (SSM), to align with the growing trend in hydrological forecasting from single-step to multi-lead-time forecasting. The model is built upon the redesigned bidirectional Mamba (Bi-Mamba), where each unidirectional branch integrates dual attributes of convolution and recursion while still maintaining linear complexity, significantly improved long-term forecasting performance compared to Transformer-based models with quadratic complexity. Through deep integration with ResNet and spatiotemporal attention mechanism, ResBi-Mamba Plus extends the effective forecasting horizon to 48 hours and consistently outperforms mainstream models across various lead times (RO3). In terms of interpretability, the framework employs model-agnostic IML visualization techniques, which demonstrate that the learned attention weights align well with fundamental temporal characteristics of time series data (RO4).

- **CHAPTER 7**: This chapter summarizes the research contributions of this study, analyzes the limitations encountered during the research process, and further discusses potential directions for future work.

CHAPTER 2

LITERATURE REVIEW

As a pivotal subfield within the domain of time series forecasting, runoff forecasting plays a crucial role in water resource management and scheduling. This chapter presents a comprehensive review of existing literature on runoff forecasting to evaluate the current state of research in the field and to identify key limitations in mainstream approaches. Section 2.1 introduces the concept and categories of runoff forecasting. Section 2.2 reviews runoff forecasting models based on traditional statistical methods. Section 2.3 presents a brief overview of representative modern machine learning-based models. As the core of this chapter, Section 2.4 provides an in-depth analysis of mainstream neural network-based runoff forecasting models. Section 2.5 discusses related studies in deep learning that are relevant to this research, including research on ensemble methods and model interpretability.

2.1 Preliminaries

2.1.1 Fundamental Concepts

The rapid development of the economy and society is inseparable from the rational utilization of water resources [34]. However, river runoff can be considered as a complex nonlinear system, which is affected by both natural climate change [35] and human social activities [36], showing obvious randomness and uncertainty [37]. Therefore, reliable runoff forecasting plays an indispensable role in the protection and rational allocation of water resources [38]. Runoff forecasting is an important branch in the field of time series forecasting [39]. Time series forecasting inputs the observed historical data in a specific time period in the past as features into the model, and accurately forecasts future data by capturing and modeling the interdependence between sequence inputs in the time dimension. Compared with regression analysis, time series forecasting is much more complicated [40]. The core difference between the two lies in the assumptions about the data. The regression analysis assumes that data are independent of each other, and changing the order of input data will not affect the outputs. Time series forecasting assumes that there is a correlation between data, and changing the order of sequence inputs will result in different outputs.

2.1.2 Category

As an important subfield of time series forecasting, runoff forecasting has undergone decades of development and refinement, resulting in a relatively mature and well-established system. Based on the underlying forecasting approach, runoff prediction models can be broadly categorized into process-based models (also known as physically driven models) [41, 42] and data-driven models. Process-based models simulate complex hydrological and physical processes within a watershed to perform forecasting. Their

main advantages lie in strong interpretability and relatively low computational cost. However, these models heavily rely on domain-specific knowledge, making them difficult to develop and less generalizable. In contrast, data-driven models learn abstract relationships between inputs and outputs directly from large-scale datasets, allowing them to predict future runoff without requiring prior knowledge of complex physical processes. As a result, they can be generalized more rapidly and flexibly to a wide range of practical scenarios. Data-driven models are gradually replacing process-based models as the mainstream approach in runoff forecasting, and they can be further divided into traditional statistical methods, modern machine learning methods and neural network-based methods. The following sections will introduce and analyze representative studies in runoff forecasting following the framework illustrated in Figure 2.1.

2.2 Traditional Statistical Methods

Traditional runoff forecasting methods are mostly based on statistical modeling [43], which relies on assumptions about data-generating mechanisms such as stationarity [44]. In the age of limited computing power, time series decomposition [45] with relatively simple procedure is the more commonly used runoff forecasting method. Time series decomposition divides the time series into three items by multiplicative decomposition and additive decomposition according to different time series categories: trend item, seasonal item and residual item, and then uses moving average method or exponential average method to analyze the data. These runoff forecasting methods have been gradually replaced by better-performing algorithms.

Autoregressive model represented by autoregressive integrated moving average model (ARIMA) [46, 47] is a traditional forecasting method that is still widely used. ARIMA model consists of autoregressive model (AR) [48], integration (I) and moving average model (MA) [49]. AR is an autoregressive model that only depends on the

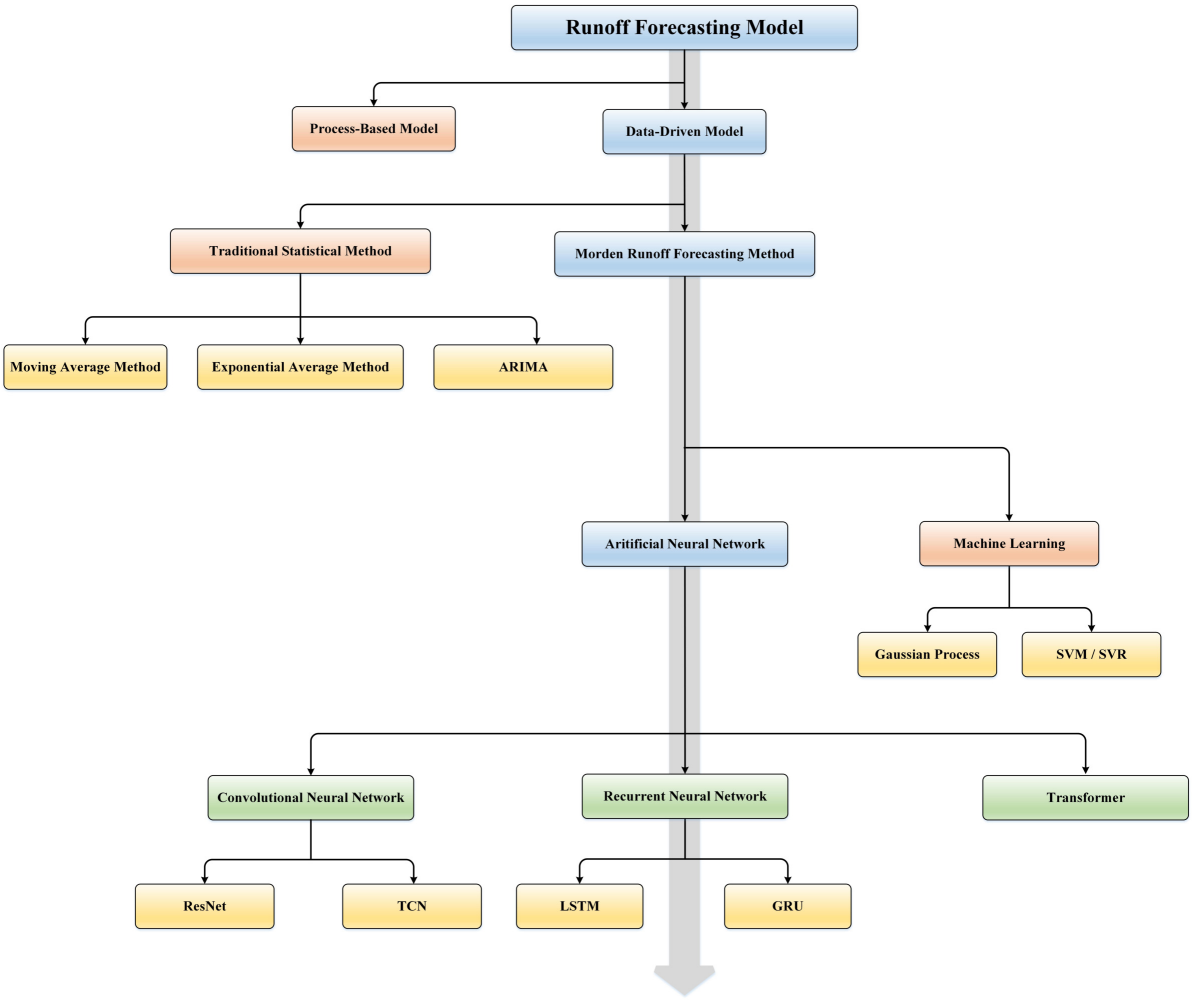


Figure 2.1: The categories of runoff forecasting models. Different colors represent the level of abstraction of the model categories. Blue and red are the highest level, green is second, and yellow represents the specific model

historical information of the sequence inputs and does not depend on other explanatory variables. The forecast of the AR model for the current time point is the regression of p historical values, and p is the order of AR. Since ARIMA requires the time series to be stationary, it is necessary to integrate the unstable time series such as runoff to obtain a stationary sequence, and I represents the d-order integration process. MA is a q-order moving average model. Unlike AR model, MA model's forecasting for the current time point depends on q historical forecasting error values. ARIMA integrates these three components to transform non-stationary time series into stationary ones,

and subsequently performs forecasting by regressing the dependent variable on past observations and error terms.

Although ARIMA is a traditional time series forecasting method, it still plays a critical role in runoff forecasting [50]. For example, Zhang et al. propose a hybrid medium-long term runoff forecasting model using singular spectrum analysis (SSA) and ARIMA [51]. SSA-ARIMA decomposes the medium-long term runoff series into subsequences with periodic characteristics through SSA, and then uses ARIMA to further forecast and correct the subsequences to improve accuracy. Wang et al. also propose a similar forecasting model [52], using the time domain decomposition method: empirical mode decomposition/ensemble empirical mode decomposition (EMD/EEMD) to decompose the original runoff series into subsequences with different time scale characteristics, and then use ARIMA to forecast each subsequence. Multiple experiments demonstrate that the performance of these hybrid models combined with ARIMA is obviously better than that of one single model. ARIMA, the representative of traditional time series forecasting models, is still a reliable and efficient runoff forecasting method.

2.3 Modern Machine Learning Methods

With the revolution of computing power, modern forecasting methods represented by machine learning have become the mainstream data-driven runoff forecasting models. Support vector machine (SVM) [53] is a classic model based on structural risk minimization, which has excellent performance in regression problems and classification problems, and support vector regression (SVR) is a form of SVM applied in regression problems [54]. SVM maps the original data space to the high-dimensional space, thereby transforming the nonlinear problem into a linear problem in the high-dimensional space, and by selecting appropriate kernel function, the inner product of the high-dimensional space is transformed into the inner product of the low-dimensional space, which largely

avoids the curse of dimensionality. SVM can maintain high accuracy when the feature dimension is large and the sample size is small, and has outstanding generalization capability.

Zhao et al. propose a chaotic least squares support vector machine hybrid model using EMD (EMD-CLSSVM) for long-term runoff forecasting [55]. EMD-CLSSVM decomposes the original runoff series into multiple stationary subsequences by EMD, and uses different methods for forecasting according to whether the series has chaotic properties: least squares support vector machine is used to predict subsequences with chaotic properties, polynomial method is used to deal with subsequences without chaotic characteristics, and finally the forecast results of multiple subsequences are reconstructed to obtain the final forecast of the original runoff. Experiments prove that the proposed EMD-CLSSVM model can well overcome the shortcomings of traditional runoff forecasting models that cannot accurately forecast the chaotic runoff sequence, and effectively improve the performance of the model.

Moreover, gaussian process (GP) is also a machine learning method that can be used for both regression and classification problems [56, 57]. The essence of GP is a combination of a series of random variables obeying normal distribution in an index set, which can solve complex time-varying nonparametric functions, so it is suitable for complex time series forecasting. Sun et al. use the form of GP in regression: gaussian process regression (GPR) to forecast monthly runoff [58]. Runoff forecasting methods based on machine learning have their own advantages and limitations in different aspects, so it is of great significance to select an appropriate algorithm according to specific needs and scenarios.

2.4 Neural Network-Based Runoff Forecasting Models

Neural network is a highly abstract artificial intelligence model of the way the human brain processes external information. It is the most commonly used and novel technology in time series forecasting. Neuron is the basic unit of information processing in human brain. There are about 100 billion neurons in the brain, which form a biological neural network. As early as 1904, biologists had discovered the physical structure of neurons in the human brain. A standard biological neuron consists of multiple dendrites for receiving incoming information and an axis for transmitting information to other neurons. sudden composition. When the signal received by the dendrite exceeds the threshold, the neuron will be activated and send a signal to the axon to deactivate the corresponding neuron.

The mechanism of the artificial neural network is just with reference to the biological neural network, McCulloch and Pitts proposed the first neuron model based on this mechanism in 1943. A standard neuron in the network has n inputs, and each input has a corresponding weight. Neuron linearly combines the inputs according to the weights and performs nonlinear transformation on them through activation function to obtain the neuron's output. The proposal of neurons laid the foundation of neural networks, but the original neuron model had many limitations, such as the weight of neurons was preset, so the model did not have the ability to learn. In 1958, computational scientist Rosenblatt proposed a two-layer perceptron model that could be used for simple image recognition. The perceptron model is the first artificial neural network with learning ability. However, the two layers of the perceptron are the input layer and the output layer, not the hidden layer structure in modern neural networks, so the perceptron is just a single-layer neural network that can only solve simple linear classification tasks. In 1986, Rumelhar and Hinton proposed a milestone in neural network: backpropagation

algorithm [59]. In neural network, forward propagation means that the training data is passed into the neural network, and the forecast of the network is obtained after the training of hidden layers. Backpropagation calculates the error between forecast and label value through cost function, and updates the weights of each layer through optimization algorithms such as gradient descent along the direction from output to input, this allows neural network to continuously improve its accuracy through iterative training.

The proposal of the two-layer neural network marks the continuous growth and maturity of modern neural network. People can train networks through huge amount of input data, and continuously improve the number of layers and performance of the neural networks through various optimization algorithms, so that modern neural network develops in the direction of deep learning and shows excellent performance in various fields. Among them, in the field of runoff forecasting, three neural network models are considered as the mainstream methods: CNN, RNN, and transformer, which is currently emerging in CV and NLP. This section explores the application of these high-performance neural networks in runoff forecasting.

2.4.1 Runoff Forecasting Models Based on Convolutional Neural Network

CNN is a modified neural network that can perform convolution operations. A standard CNN contains multiple layers: convolutional layers, pooling layers and fully connected layers. Two core properties of convolutional neural networks are sparse connections and weight sharing. CNN performs convolution by sliding convolution kernels in each feature map, which makes neurons in each layer only need to be sparsely connected with part neurons in the previous layer and share the parameters of one convolution kernel. CNN requires significantly fewer parameters than the fully connected neural network, which

enables the stacking of more layers to achieve enhanced feature extraction capability. [60].

Li et al. propose a rainfall-runoff forecasting model [61] based on CNN and deep belief network (DBN) [62]: convolutional deep belief networks (CDBN). Benefits from its sparse connection and weight sharing properties, CNN can stack multiple layers to extract more abstract features. However, the continuous increase of network depth brings the following problems: too many parameters lead to greatly increased training time, gradient descent algorithm is easy to fall into local optimum, and vanishing/exploding gradient problem makes deep network difficult to train. DBN is the most effective method to solve the problem of training deep neural network. It assigns appropriate initial weights to the entire network through layer-by-layer unsupervised pre-training, which largely reduces the training difficulty, so that the network only needs to be fine-tuned to obtain the optimal weight value. The proposed CDBN is a probabilistic generative model, inspired by [63], consisting of multiple layers of convolutional restricted Boltzmann machines (CRBM), including a visible layer for receiving input and hidden layers for extracting features. In the experimental part, CDBN uses historical rainfall and runoff data of different lengths to forecast the runoff in the lead time of 1 day, 3 days and 5 days. The highest R^2 of CDBNs is 0.94, which well proves its validity.

Yan et al. propose a flow interval forecasting model [64] based on CNN, residual network (ResNet) and lower and upper bound estimation (LUBE) [65]. The proposed stResNet-LUBE consists of 4 modules: convolutional layers perform preliminary feature extraction on the input historical rainfall, historical runoff and future rainfall data, and then visual geometry group (VGG)-based [66] ResNet is used for further forecasting. The forecast result will be passed to the additional feature fusion module, and the final output is obtained through the LUBE fully connected layer. Results of the interval forecast for the next 6 hours at a confidence of 90% on the spatiotemporal dataset of Tunxi show

that the stResNet-LUBE using ResNet structure has obvious performance improvement compared with stCNN-LUBE. R^2 increases from 0.948 to 0.966, proving that ResNet can indeed improve the performance of CNN.

Vanilla CNN needs to linearly stack multiple layers to learn long-term dependencies, while TCN can achieve the same effect with fewer layers by dilated convolution. The proposal of TCN enables CNN to have both accuracy and efficiency in time series forecasting, and TCN has already demonstrated its excellent performance in runoff forecasting. Lin et al. propose a runoff forecasting model that combines TCN and encoder-decoder (ED) [67] architecture for Jianxi basin, China [68]. The proposed TCN-ED not only possesses the excellent time series forecasting capability of TCN, but also significantly improves the stability of the model through the encoder-decoder architecture, making TCN-ED more robust to fluctuations in the rainfall process.

2.4.2 Runoff Forecasting Models Based on Recurrent Neural Network

RNN is a model that can process sequence inputs. Traditional neural networks do not have the ability to process sequential inputs, that is, previous inputs do not affect subsequent inputs. However, when dealing with time series such as runoff, we want the model to capture the hidden relationships between the sequence inputs in the time dimension. When RNN processes time series, the state of the hidden layer of each moment is jointly determined by the current input and the state of the hidden layer of the previous moment, which makes the RNN have memory and can store the previous information and learn the hidden relationships in the time dimension. RNN also suffers from several limitations, such as the long-term dependencies and vanishing/exploding gradient, which may lead to the failure of the model to be well trained. As a result, the most widely used recurrent architecture-based neural network in current practice is

LSTM. Different from vanilla RNN, LSTM has its unique gating mechanism [69], which can selectively memorize useful information and discard useless information through forget gate, input gate and output gate.

Due to the above advantages, LSTM has become the most widely used runoff forecasting model. Yuan et al. propose a hybrid LSTM and ant lion optimizer model (LSTM-ALO) to perform monthly runoff forecasting for Astor River Basin [8]. LSTM-ALO adopts the ALO optimization method to optimize two important parameters of LSTM: the number of hidden layers HN and the learning rate α. Multiple sets of control experiments have proved that ALO can improve the performance of LSTM under different model inputs, and LSTM-ALO has better results than the hybrid of LSTM and particle swarm optimization method (LSTM-PSO).

LSTM can also be combined with CNN to improve accuracy. Li et al. perform rainfall-runoff forecasting using CNN-LSTM combined model [70]. The proposed model leverages the strengths of both CNN and LSTM: CNN extracts features from two-dimensional rainfall radar maps, while LSTM handles the outputs of CNN along with upstream runoff data. Experiments demonstrate that the proposed CNN-LSTM model can accurately forecast the downstream runoff of the Elbe River Basin in Sachsen, Germany during wet and dry periods. The combined model based on CNN and LSTM has also been proven to work well for river basins with historical rainfall radar maps and recorded runoff data.

Liu et al. propose another hybrid model: Conv-TALSTM that combines CNN, LSTM and temporal attention mechanism [71] to forecast the runoff of the Han Jiang River Basin [72]. The inputs of the proposed model use historical daily meteorological and hydrological data recorded by meteorological and hydrological stations in the Han Jiang River Basin, including daily precipitation, temperature, wind speed, relative humidity, sunshine duration, and daily flow. Due to the massive input features, one-dimensional CNN is used to extract abstract features from different variables in the preprocessed

inputs, and the outputs of the convolutional layer are passed to the subsequent LSTM layer to learn the hidden relationship in the time dimension. The temporal attention mechanism models the importance of LSTM outputs at different moments and represents them as weights. The outputs of the LSTM layer are weighted and summed with the weights normalized by softmax, and finally, the forecast result of the model is obtained through a fully connected layer. Multiple experiments demonstrate that Conv-TALSTM performs runoff forecasting better than traditional neural networks and distributed hydrological model Wetspa [73].

Several variants have been developed based on LSTM. Among them, GRU simplifies the structure and improves computational efficiency by combining certain gating units, making it popular in runoff forecasting. Bi et al. propose a hybrid model CAGANet based on CNN, GRU, attention mechanism and AR model to forecast daily runoff of the Qingxi River Basin [74]. Daily runoff has strong uncertainty and instability, so the proposed method first uses a linear interpolation method [75] to enhance the stability of the input hydrological data, and then inputs the enhanced data into CAGANet. The enhanced data is divided into long-term data and short-term data. The long-term data is input into the convolution layer for feature extraction, and the extracted features are passed to the attention mechanism layer to assign weights, then the weighted data is passed to the GRU layer to perform time series forecasting for the nonlinear part. The short-term data is passed into the AR model to forecast the linear part, and then the nonlinear part and the linear part are integrated to obtain the final forecast result of CAGANet. Several sets of comparative experiments have strongly demonstrated that the proposed CAGANet has higher accuracy than vanilla LSTM and AM-LSTM with attention mechanism, and the NSE can reach 0.854 without data enhancement. After using the data enhancement method to improve the data stability, the NSE of the model is even as high as 0.993, which proves that the CAGANet using the data enhancement method has excellent daily

runoff forecasting performance.

Zhou et al. add a bidirectional structure on the basis of vanilla LSTM [76], and can make full use of the complementary information of the past and the future for inference on NLP-related tasks by processing sequences both forward and backward. The proposed deep bi-directional LSTM (DB-LSTM) first utilizes a standard LSTM to process the input sequence forward, followed by a second LSTM layer that processes the output of the first layer in the opposite direction. These two LSTM layers form a basic processing unit, and these paired LSTM layers are stacked to build a deep LSTM model. By increasing the model depth and introducing the bidirectional structure, the proposed DB-LSTM has obvious advantages over traditional models.

2.4.3 Runoff Forecasting Models Based on Transformer

Attention mechanism is a kind of mechanism commonly used in deep learning models, which can make the model amplify high value features of the input. Sequence-to-sequence (S2S) models using attention mechanism have shown significant performance improvements in various fields. At present, many LSTM-based sequence-to-sequence models have demonstrated outstanding accuracy in runoff forecasting [77, 78]. However, LSTM cannot directly connect any two positions in the time series, but the connection between any two positions can make the model more comprehensive to learn the complex hidden relationship between time series. Therefore, Vaswani et al. propose the Transformer model based on full attention structure, which surpassed LSTM in the NLP field [79–81]. Transformer is a high-performance cutting-edge neural network model. Like most sequence-to-sequence models, Transformer consists of an encoder and a decoder, and both the encoder and the decoder contain 6 blocks. Each encoder block and decoder block contains Multi-Head-Attention [82] composed of Self-Attention [83], where the encoder block contains one Multi-Head Attention, and the decoder block contains two Multi-Head

Attention, one of which is Masked Multi-Head Attention, used to prevent future information leakage. First, the Transformer model obtains the representation vector of the time series input. The representation vector is obtained by adding the Embedding of the input features and the Embedding of the input position. The representation vector matrix is passed to the 6 stacked blocks in the encoder, and the encoded information matrix of all time series inputs is obtained. The encoded information matrix is then passed to the decoder, which in turn predicts the next sequential input based on the currently predicted inputs.

Transformer has also been initially applied in runoff forecasting. Yin et al. propose the first Transformer model RR-Former for rainfall-runoff modeling [84]. The model can directly strengthen or weaken the connection of two arbitrary positions of time series inputs through the Transformer's full attention module. Experiments show that the accuracy of the proposed RR-Former is significantly better than that of the LSTM-based model on two tasks of single rainfall-runoff simulation and regional rainfall-runoff simulation, which proves the feasibility and effectiveness of Transformer in the runoff forecasting task. Although the Transformer architecture has not been applied on a large scale in runoff forecasting, its excellent long-distance modeling capability and multimodal fusion ability make it have great potential and research value, and it is an important development direction of future runoff forecasting models.

Transformer is an emerging neural network branch with vigorous vitality. Although its application in the field of runoff forecasting has just started, Transformer has become the most popular research object in the field of CV and NLP, and various variants with different characteristics have appeared. Lin et al. study the current mainstream Transformer variant structures [85], and propose that these variants are mainly to improve the vanilla Transformer from three directions: efficiency, generalization capability and application scenarios. In terms of improving the efficiency of the model, the current

direction of improvement is mainly to use lightweight attention mechanism and divide-and-conquer methods to modify the self-attention module, so as to solve the problem that Transformer is inefficient when processing long sequences. To improve the generalization capability of Transformer, structural bias, regularization and large-scale pre-training are introduced. In addition, Transformer can also be customized for specific downstream tasks, so that it can perform well on specific tasks. Transformer is still under rapid development and we believe it is one of the most promising runoff forecasting methods at present.

When applying Transformer to runoff forecasting, in addition to exploring the lightweight aspect of the model, making full use of Transformer's multi-modal fusion capability is also one of the effective ways to improve model performance. The data we obtain for runoff forecasting in a region is often diverse, so the fusion of different modal data for prediction can significantly improve the expressiveness of the model. For example, we can fuse the information of the two modes of rainfall radar map and historical runoff data in the target area through Transformer. First, networks like CNN can be used to extract features from two-dimensional images, and then time series forecasting model can be used to process one-dimensional historical runoff data, and then the output feature maps of the two models can be fused through the self-attention module of Transformer. Through this approach, models that incorporate data from multiple modalities can make more accurate predictions.

2.5 Other Related Studies

2.5.1 Ensemble Methods

Through previous surveys and research, we found that most of the current runoff forecasting models are accuracy-oriented. However, Su et al. point out that well-trained

deep neural networks still have the potential to misclassify adversarial examples [33]. Experimental results based on evaluations of 18 ImageNet models indicate that these high-performance models often have obvious trade-offs between accuracy and robustness, and blindly pursuing accuracy may bring serious loss of robustness. Ensemble method [86] is considered as an important way to improve robustness and accuracy. Individual models may incorrectly classify or predict some samples in complex scenarios. Therefore, integrating multiple models through a specific algorithm to obtain a model with stronger accuracy and robustness can reduce the probability of the model making wrong forecasts to a certain extent.

If there is no strong dependency between the individual learners used for ensemble, and a series of individual learners can be generated in parallel, then this type of ensemble method is called Bagging series ensemble method [87]. For the outputs of multiple homogeneous learners trained in parallel, the Bagging ensemble method adopts different aggregation strategies depending on the task type. For classification tasks, the final prediction is obtained through majority voting, whereas for regression tasks,Äîincluding time series forecasting-the outputs are typically averaged. By leveraging multi-model decision-making, Bagging significantly reduces the impact of individual model errors on overall performance, thereby enhancing robustness. It is simple to implement, model-agnostic, and exhibits strong generalization capabilities. However, its major drawback lies in the substantially increased computational cost resulting from the training of multiple models.

When individual learners exhibit strong dependencies and must be generated in sequence, the ensemble approach is referred to as the Boosting series ensemble method. Freund et al. demonstrate a powerful ensemble approach to boosting: Adaboost [88]. AdaBoost assigns weights to training samples and weak learners. After each iteration of training, it increases the weight of misclassified samples and reduces the weight of

correctly classified samples, which makes subsequent weak learners pay more attention to the misclassified samples in previous training process. In addition, the weak learners are also given weights according to their performance, and finally the weak learners will be integrated according to the weights to obtain a strong learner with better comprehensive performance. Huang et al. further combine ResNet with AdaBoost, and propose BoostResNet [89], which treats each residual block as a weak learner, then the output of BoostResNet is the progressive integration of all residual blocks. Ensemble method also has great application potential in runoff forecasting, which can help people train more robust and accurate runoff forecasting models.

2.5.2 Interpretability

Interpretable machine learning is also a hot research direction in artificial intelligence, and its goal is to balance the performance and interpretability of machine learning. When developing machine learning models, people tend to pay more attention to the accuracy, robustness and generalization of the model, but these indicators cannot completely determine whether a model can be widely recognized and applied in the field. If people don't fully understand why machine learning models make decisions, or if the models are not as transparent as expected, those models may not be adopted. According to Molnar [90], interpretability refers to how well people can comprehend the reasoning behind a decision or reliably anticipate a model‚Äôs output. The interpretability of a model allows its behavior to be better understood and accepted by people.Interpretable machine learning mainly explains three aspects: the motivation of the model's forecast, the reason why the model makes the forecast, and how people can trust the model's forecast. In order to explain the above problems, many interpretability methods have been proposed [91], which can be further divided into intrinstic or post-hoc method, model-specific or model-agnostic method, and local or global method according to the

different characteristics of the methods. In the field of time series forecasting, there is more focus on whether interpretability methods are model-specific or model-agnostic.

Chang et al. propose a multivariate time series forecasting model based on memory network [92]: memory time series network (MTNet) [93], which considers the interpretability of time series forecasting models. MTNet uses an attention mechanism to model the importance of the previous sequence to the current forecast in the time dimension. Therefore, in the experimental part, MTNet adopts a local model-agnostic interpretability method to visualize the weight of the attention module. The historical sequence segments that are similar to the recent data will be given a higher weight by the attention module, which increases the interpretability of the attention module. Lim et al. propose the Temporal Fusion Transformer (TFT) [94] based on the Transformer structure for interpretable multi-horizon time series forecasting. TFT has a good trade-off between performance and interpretability, combining multi-horizon forecasting with interpretable time dynamics. From the beginning of the design, TFT has considered how to reasonably interpret the forecast results of the model. Self-attention module in the traditional Transformer mainly considers the importance of time, while ignoring the importance of features at a specific time point. Therefore, the variable selection network (VSN) is used in TFT for variable selection, and the attention mechanism is also used to identify the time steps that have the greatest impact on model performance. In the experimental part, the authors demonstrate three model-specific interpretability use cases, starting with analyzing variable importance of inputs, and quantifying the variable selection weights of VSN to explain the general relationships learned by VSN. The second is to visualize the weights learned by the attention module to increase people's trust in the attention module. The final step involves detecting regimes or events that cause notable shifts in temporal patterns. In the S&P 500 volatility scenario, the model,Äôs attention weight exhibited clear changes during the 2008 financial crisis,

demonstrating its ability to recognize important regimes.

Cheng et al. proposed a novel multi-modality graph neural network (MAGNN) [95]. Interpretability is achieved through model-specific feature weight visualization. The model uses multi-modality sources as input to the model and retains historical information from the sources (e.g., historical precipitation, soil quality, climate, etc.) The inner-modality attention model captures graph-structured relationships between inputs and target outputs. Meanwhile, the inter-modality attention mechanism assigns adaptive weights to different sources based on their contributions to prediction. Model interpretability can be realized by visualizing attention weights, such as through heatmaps, for both intra-modality graph attention and inter-modal source attention. The resulting weight differences reveal feature importance and help identify the key factors driving the forecasting outcomes.

CHAPTER 3

HOURLY RUNOFF FORECASTING BASED ON ENSEMBLE ATTENTION TEMPORAL CONVOLUTIONAL NETWORK

RQ1 highlights the limitations of current runoff forecasting models, which often prioritize accuracy at the expense of robustness and efficiency. Addressing RO1, this chapter proposes the Ensemble Attention Temporal Convolutional Network (EA-TCN), an advanced framework tailored for hourly runoff forecasting. The innovation stems from combining Temporal Convolutional Network (TCN), lightweight attention module, and ensemble learning strategy into a cohesive architecture. This design jointly improves precision, efficiency, and robustness. The TCN backbone, with its shared parameters and parallelizable structure, accelerates computation while effectively modeling long-term dependencies through causal and dilated convolutions. The attention mechanism enhances temporal feature extraction, allowing EA-TCN to capture complex patterns and achieve strong temporal robustness across varying forecasting horizons, surpassing traditional models. Moreover, the Snapshot ensemble technique enables the framework to mimic the benefits of training multiple models within one single learning process, further strengthening performance. Extensive ablation and comparative studies on the US Columbia River

dataset confirm the contribution of each component and demonstrate the superiority of their integration. Overall, our findings establish EA-TCN as an oustanding solution for short-term runoff forecasting.

Section 3.1 introduces the background and discusses the current challenges as well as the motivation behind the proposed methodology. Section 3.2 defines the runoff forecasting task. Section 3.3 presents the proposed methodology in detail, including the structure and functionality of each module. Section 3.4 provides a comprehensive evaluation of the model through ablation and comparative experiments. Finally, Section 3.5 summarizes the chapter and outlines the limitations identified in this study.

3.1 Introduction

River runoff is affected by various factors such as meteorological activities, underlying surfaces, and human development [96, 97], and exhibits strong nonlinearity, randomness and uncertainty. However, changes in river runoff have a significant impact on the ecological environment and human activities in the river basin, which brings huge opportunities and challenges to the current use of water resources [98]. Therefore, accurate runoff forecasting has gradually become an indispensable part of modern hydraulic engineering system. Runoff forecasting can provide data and theoretical support for flood control, water resources planning, hydropower station scheduling, and river basin environmental governance, so that water conservancy department can better deal with the opportunities and challenges brought about by complex and changeable river runoff [99].

Fine-grained hourly runoff forecasting typically involves runoff over horizons ranging from a few hours to several days [100]. At this finer temporal resolution, river runoff is highly sensitive to external factors such as temperature fluctuations, sudden rainfall

events, and human activities, resulting in significant temporal variability and frequent high-frequency fluctuations. These characteristics give rise to strong nonlinearity and non-stationarity in the data, making hourly runoff forecasting inherently a task of modeling a high-frequency, complex, disturbance-prone, and rapidly responsive dynamic system. Moreover, compared to daily, monthly, or annual runoff forecasting, hourly forecasting relies on much longer sequences containing high-resolution dynamic features. This leads to a substantial increase in both data volume and density, placing greater demands on the model's capability to capture temporal dependencies [101].

Hourly runoff forecasting is virtually indispensable in hydrological scenarios that emphasize real-time monitoring, short-term warning, and rapid response. In addition, runoff forecasting can bring huge economic benefits. Taking the European Flood Awarness System (EFAS) [102] as an example, by calculating the potential avoidable flood losses, it can be concluded that for every 1 euro invested in early flood warning, the return is about 400 euros. It can be seen from the example of EFAS that this cross-border continental-scale flood warning system based on runoff forecasting has very considerable potential monetary benefits, which intuitively shows that runoff forecasting has great research significance and broad application prospects. For the reasons outlined above, this study focuses on hourly runoff forecasting, which is considered the most challenging among runoff forecasting tasks. The review of mainstream runoff forecasting models reveals that most existing approaches are predominantly accuracy-oriented and therefore tend to rely heavily on complex time series forecasting. Neural networks based on recurrent architecture, such as LSTM and GRU, have delivered impressive performance in runoff forecasting [103], yet they exhibit certain inherent limitations. For example, LSTM relies on numerous gating units to regulate input information, resulting in an excessive number of parameters. Furthermore, its sequential design requires each cell's computation to depend on the previous output, leading to low computational efficiency.

The lack of efficiency can significantly affect both training and inference speed during real-world deployment, and may even determine whether a model can be implemented on edge hydrological monitoring devices or embedded systems. In addition, contemporary runoff forecasting datasets are typically collected automatically by sensors or monitoring stations deployed across river basins. As a result, these datasets often contain measurement errors or missing values due to limitations in sensing and recording processes. At the hourly scale, small perturbations or anomalies can have a disproportionately large impact on model performance compared to coarse-grained forecasting. In high-risk scenarios such as flood early warning [104], such sensitivity may lead to severe misjudgments. This imposes particularly stringent requirements on the robustness of hourly runoff forecasting models. However, robustness remains a largely overlooked metric in the design and evaluation of existing models. Most mainstream runoff forecasting models fail to explicitly consider robustness during construction and testing, which can substantially compromise their overall performance in practical applications [105].

To address these critical challenges, we present EA-TCN, an innovative and efficient model designed for runoff forecasting. The model builds upon the capabilities of TCN, while incorporating a lightweight attention module and the Snapshot ensemble method. The main contributions of this study include the following:

- An improved lightweight TCN architecture based on convolutional operations is applied to runoff forecasting. Compared to conventional recurrent architecture-based neural networks, TCN's smaller parameter size, greater architectural flexibility, and inherent parallel processing capability make EA-TCN a strong competitor, surpassing existing models in both accuracy and computational efficiency.

- A key strength of EA-TCN is its integration of our proposed plug-and-play attention module specifically crafted for temporal data. This module efficiently identifies

the importance of each time step in the input sequence while incurring very low computational cost.

- To simultaneously improve accuracy and robustness, the Snapshot ensemble method is incorporated into EA-TCN. This approach enables the effect of training multiple models in one single process, without introducing additional parameters or extending training time.

3.2 Definition

Long short-term multivariate hourly runoff forecasting. Given historical multivariate hourly runoff series with a fixed-length look-back window of L samples $X = \{x_1, x_2, \ldots, x_L\}$, where data point $x_t \in \mathbb{R}^M$ at t-th time step consists of M variables, the objective of the model is to predict the runoff sequence $\mathscr{F} = (x_{L+1}, x_{L+2}, \ldots, x_{L+T})$ in the next T hours. The forecasting process is conducted through the sliding window, whereby after each forecasting of the future runoff sequence $\mathscr{F}$ is completed, the look-back window slides forward according to a predefined step size T for subsequent forecasts. To ensure that the predicted values closely approximate the ground truth, it is important for the model to be trained on sufficiently long historical sequences, particularly when capturing temporal dependencies and periodic runoff behaviors. Therefore, the look-back window length of historical data L should be greater than or equal to the forecasting horizon T.

3.3 Methodology

The proposed EA-TCN integrates three complementary sub-modules, each designed to address key challenges in time series forecasting. The TCN backbone enables parallel processing of input sequences, improving efficiency and overcoming the limitations of recurrent models with long sequences. An adaptive attention mechanism, inspired by

the squeeze-and-excitation network (SENet), adjusts the weight of each time step to highlight critical temporal patterns and dependencies, thus boosting predictive accuracy. The Snapshot ensemble method further enhances generalization and robustness by capturing diverse model states at different training stages, without the need for training multiple separate models. These components work in concert to deliver a solution that combines accuracy, efficiency, and resilience, distinguishing EA-TCN from neural networks based on recurrent architecture. This section provides a detailed introduction to the architecture and advantages of each module.

3.3.1 Temporal Convolutional Network

For a long time, RNN, LSTM and GRU are considered as mainstream sequence forecasting methods, the recurrent architecture of RNN enables the network to retain and utilize the previous inputs, giving the network the capability of memorizing information. LSTM and GRU incorporate gating mechanisms into the foundational RNN structure, thereby endowing the network with the capacity to capture long-term dependencies and solve the problem of vanishing/exploding gradient. Vanilla CNN is rarely used in the field of sequence forecasting, because the filter size limits the receptive field of CNN, making it difficult to model long input sequences. However, RNN can only read and process one single input at a time, which means that the network must complete the processing of the input at the present time before it can continue to process subsequent inputs, so RNN cannot perform massive parallel processing like CNN. In addition, RNN is a compute-intensive model. All intermediate information needs to be retained in each training, so it will occupy a large amount of memory resources. With the continuous development of CNN, more and more CNN structures are tentatively used in the field of sequence forecasting, such as the Wavenet for speech generation modeling proposed by Google [106]. The advantage of CNN in dealing with sequence forecasting problems is

that each calculation does not need to wait for the completion of the previous information processing, so each calculation is independent, and massive parallel processing can be performed to improve efficiency.

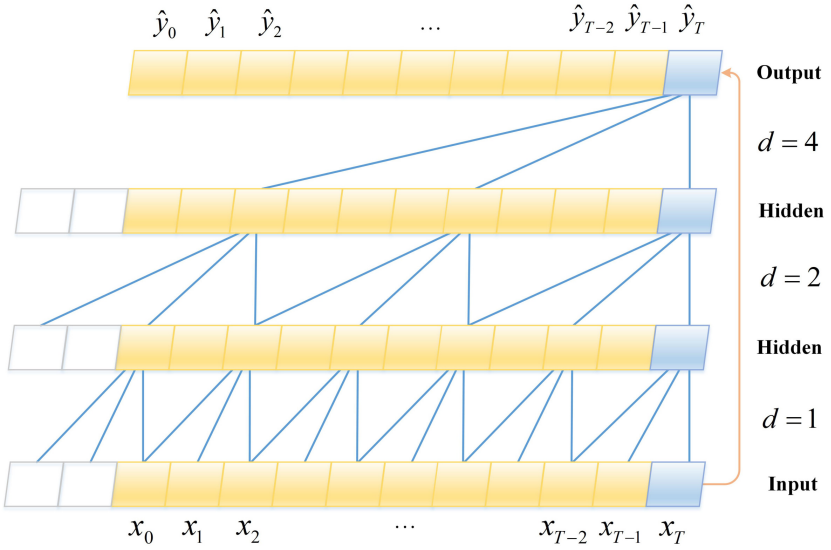


Figure 3.1: The structure of TCN. The TCN showed in the figure employs dilation factors of $d = 1, 2, 4$, allowing the model's receptive field to expand exponentially. The orange line on the right represents the 1×1 convolution shortcut connection within TCN that links the input directly to the output. The utilization of residual connection and A convolution significantly enhances the TCN's capability to capture long-term dependencies.

Can CNN replace RNN and become a universal sequence forecasting model? Bai et al. propose an improved CNN: TCN [107], and compares TCN with recurrent architectures: RNN, LSTM and GRU on 11 different types of sequence forecasting tasks. Experimental results show that TCN has higher efficiency and accuracy on 9 out of 11 tasks, which proves that TCN can surpass the mainstream methods based on recurrent architecture in sequence forecasting. On this basis, TCN also possesses the unique lightweight architecture and parallel processing capability of CNN. Its computational cost and complexity are significantly lower than those of models based on recurrent architecture. This affords us the potential to construct a deeper model for learning high-level abstract features, which underpins our choice of TCN over widely used time series forecasting

models like LSTM. The structure of TCN is shown in Figure 3.1. In order to better apply the convolution structure to the processing of sequence problems, TCN uses causal convolution. The traditional convolutional neural network has a drawback when performing sequence forecasting, that is, the output at time t may depend on the elements at time $t+1$ and later in the previous layer during convolution, which indicates that the model uses future data to forecast the present input. Causal convolution imposes strict time constraints on ordinary convolution operations. The output at time t will only be convolved with elements at present and earlier time in the previous layer. This one-way structure ensures that future information will not be leaked. TCN also uses 1D fully-convolutional network (FCN) and zero padding, so that the length of each convolution layer is consistent with the input layer.

Another characteristic of TCN is dilated convolution [108]. Both ordinary FCN and causal convolution have a problem, that is, the size of the filter will limit the modeling length of the time series inputs. If the traditional CNN wants to increase the receptive field to learn long-term dependencies, it needs to linearly stack a large amount of layers, which will greatly increase the depth of the network. TCN introduces dilated convolution to solve this problem. Dilated convolution allows the convolutional layer to sample the sequence input at intervals, and control the sampling rate through the dilation factor. As shown in Figure 3.1, the dilation factor of the first layer is $d = 1$, which indicates that the model at this time is performing a regular convolution, and every input is sampled. In the second layer, $d = 2$, which means that every two inputs will be sampled once as input. The dilation factor increases to $d = 4$ in the last layer, at this time, every four inputs will be sampled once as input. The dilated convolution operation F on element s of the sequence input can be formulated as:

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \tag{3.1}$$

where $\mathbf{x}$ represents the 1-D sequence input, f and k represent the filter and filter size respectively, d is the dilation factor. The dilation factor d, which increases exponentially with the number of layers, enables the model to obtain a large receptive field with a small number of layers, thus improving the model's ability to learn long-term dependencies. Moreover, as shown Figure 3.1, a residual connection from input to output is also established in each TCN module. It is worth noting that the shortcut connection type widely used in ResNet is the identity mapping shortcut connection. In TCN, since the input and output may have different dimensions, a 1×1 convolution residual connection is used to ensure that the input and output have the same dimension when performing element-wise addition.

As a kind of modified CNN that can perform sequence forecasting, TCN has the following advantages: TCN is capable of conducting extensive parallel processing, when the input sequence is long, the training efficiency will be improved; TCN has a flexible receptive field by stacking multiple convolutional layers and controlling the dilation factor and filter size, which can be adjusted according to different tasks; The back propagation path of TCN is different from the direction of the sequence, which avoids the vanishing/exploding gradient in recurrent architectures; TCN has the feature of CNN weight sharing, that is, each layer shares one convolution kernel, so it takes up lower memory during training process. It is thus well-suited for short-term runoff forecasting tasks demanding both precision and efficiency.

3.3.2 Lightweight Attention Mechanism based on Squeeze-and-Excitation Network

Short-term runoff forecasting requires flexible forecast across various future lead times, making multi-lead-time capability essential for the proposed model. Yet, existing approaches often show degraded performance as lead time increases, largely because

accommodating longer horizons demands multiplying the number of input time steps. To address this, it is crucial for the model to focus on informative inputs while filtering out irrelevant ones. We therefore tightly integrate an attention mechanism with the TCN, enabling EA-TCN to adaptively extract critical temporal features.

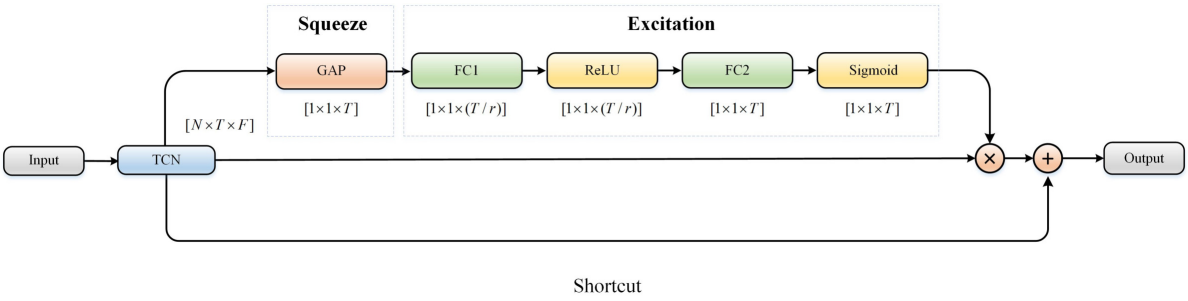


Figure 3.2: The modified SENet module is embedded in each TCN layer of EA-TCN, enabling the model to focus on informative time steps while down-weighting less relevant ones.

The attention mechanism [109] in neural networks resembles human focus by concentrating on key elements rather than treating all inputs equally. It adaptively assigns weights to different parts of the input, highlighting important information. In EA-TCN, we incorporate SENet [110], a lightweight attention module from the CV domain [111], to explicitly capture temporal dependencies. Figure 3.2 demonstrates the SENet structure, which comprises two key operations: squeeze and excitation. The squeeze step uses a global average pooling (GAP) layer to reduce the TCN's three-dimensional output $[N \times T \times F]$ into a $[1 \times 1 \times T]$ vector by aggregating along the feature dimension, where N is batch size, T is the number of time steps, and F is the number of features. This compresses all features at each time step into a single real number that represents the global distribution of the response of the time step in the feature dimension. The squeeze operation is expressed as:

$$\tilde{x_1} = \frac{1}{f} \sum_{i=1}^{f} x_1^i \tag{3.2}$$

where f is the number of features. The squeeze operation expands the receptive field while reducing parameters and computational cost, which underpins SENet's lightweight design. The subsequent excitation step uses two fully connected layers: the first compresses the dimensionality by a factor determined by hyperparameter r to lower parameter count, and the second restores it, generating weights via a Sigmoid activation. These weights rescale the TCN's output, completing the time step recalibration. The excitation process is defined as:

$$\hat{x}_1 = \sigma(W_2\delta(W_1\tilde{x_1})) \tag{3.3}$$

δ denotes the ReLU activation following the first fully connected layer, σ is the Sigmoid applied after the second, and W represents the respective layer weights.

Compared to attention mechanisms like CBAM [112] and ECA-Net [113], SENet offers a simpler design that ensures higher computational efficiency and lower complexity, making it well-suited for long time series processing. Its ability to directly modulate responses across time steps provides a global view, ideal for time series where crucial information is distributed throughout the sequence. These strengths led to its adoption as the core attention module in EA-TCN. When combined with TCN, SENet effectively captures temporal dependencies at reduced computational cost, adaptively refining the relevance of each time step. As a result, EA-TCN focuses on essential temporal features while down-weighting less informative ones, significantly improving temporal robustness and enabling precise multi-lead-time runoff forecasting.

3.3.3 Snapshot Ensemble Method

In machine learning, one individual weak learner has various limitations, such as low accuracy and robustness. Moreover, a weak learner is more likely to have certain preferences in the forecasting process, which makes it lack of generalization capability. Ensemble method can integrate the forecasts of multiple weak learners through a specific

algorithm, and obtain a strong learner whose comprehensive performance is stronger than that of one individual weak learner. Through ensemble method, even if a weak learner makes a wrong forecast, other weak learners can correct the wrong results, thereby improving the accuracy and maintaining a certain robustness when there exists noise in the data.

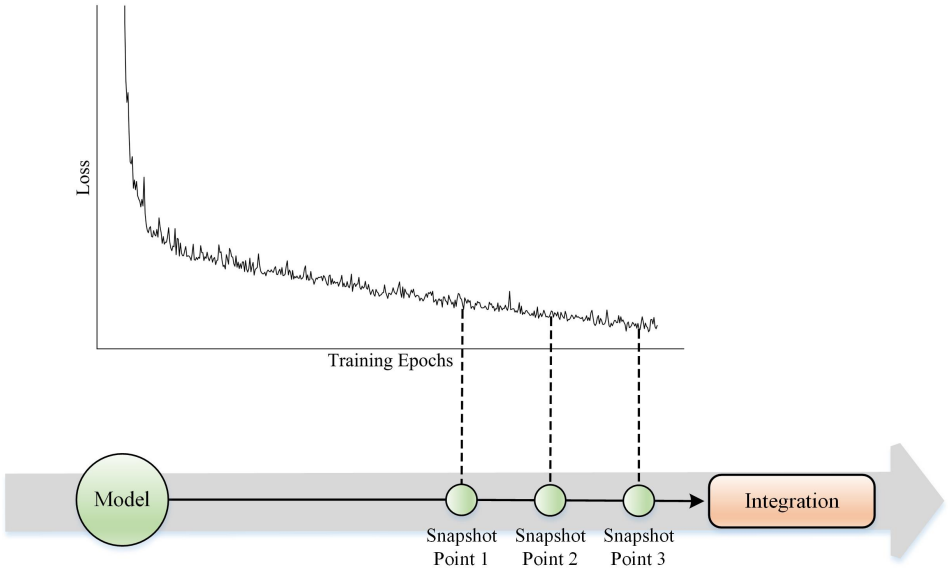


Figure 3.3: The process of Snapshot ensemble method. When the model's loss decreases to a lower level during iterative training, the model parameters at each snapshot point are saved. Upon completion of the training, these parameters are reconstructed into a complete weak learner for forecasting. This implies that Snapshot ensemble method is capable of obtaining the impact of training multiple homogeneous models within a singular training procedure.

Snapshot ensemble method is a novel ensemble method proposed by Huang et al. [114]. Unlike typical Bagging ensemble methods, the Snapshot ensemble method only needs to be trained once, and multiple models that converge to different local optimal values can be obtained and integrated. In order to achieve the effect of obtaining multiple models through only one training process, the Snapshot ensemble method uses the cosine annealing learning rate schedule: in the training process, as the epoch increases, the learning rate drops rapidly, making the model quickly step into the local optimum,

and take a snapshot of the model at this time to save the parameters. After saving the model, the learning rate returns to a larger value, which makes the model escape the current local optimal point and find a new optimal point. Repeating this process continuously, we can get multiple models in the local optimum. The typical learning rate schedule gradually reduces the learning rate during the training process, making the model converge to a flat local optimal point. Using this method to train a single model may obtain a higher accuracy value. However, the cosine annealing learning rate schedule makes the models saved in different local optimums have a higher diversity, and the models obtained after integration may have better generalization capability and accuracy.

In this chapter, the optimizer of the prposed EA-TCN is adaptive moment estimation (Adam) [115]. Adam can adjust the learning rate adaptively, so we don't need to set the learning rate schedule. Therefore, we adopt a simplified strategy similar to NoCycle Snapshot: as shown in Figure 3.3, when the loss is reduced to a relatively low level during the training progress, we take several snapshots of the same model. The models saved in these snapshots all have relatively low loss, but maintain a certain diversity. Integrating these models can improve the generalization performance of the model. Moreover, due to the reduction of generalization error, the Snapshot ensemble method also plays a role of regularization. Thanks to the aforementioned advantages, the proposed model achieves a significant improvement in robustness against disturbances such as noise and missing values in the data, with a computational cost during training that is almost negligible. In addition, it also contributes to a certain improvement in forecasting accuracy for normal samples. In the experimental part, we test the impact of the ensemble method on the model performance and robustness.

3.3.4 Model Design and Implementation Details

The proposed EA-TCN conducts short-term runoff forecasting at lead times of 2, 4, 8, 12, and 24 hours, using input sequences set to four times the corresponding lead time to effectively capture preceding time-series patterns. The rectified linear unit (ReLU) activation function [116] is utilized in the attention module and the dimension reduction part of the model output to implement non-linear transformations of the input data. Relative to traditional activation functions such as Sigmoid, ReLU boasts superior computational efficiency and the ability to alleviate the stagnation in training caused by vanishing gradients. The function is defined as follows:

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{3.4}$$

In the context of the loss function, we have opted for the mean squared error (MSE). This function is characterized by its straightforward gradient computation and a broad foundation of applications. Moreover, its sensitivity to large errors is beneficial in enhancing the accuracy of the model. The formula for MSE is:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N} \left(\mathrm{y}_i^{obs} - \mathrm{y}_i^{pred} \right)^2}{N} \tag{3.5}$$

where N represents the sample size, while y_i^{obs} and y_i^{pred} denote the observed (actual) and predicted values, respectively. To comprehensively assess the performance of models, three commonly used evaluation metrics in the field of hydrology are employed: mean absolute error (MAE), mean absolute percentage error (MAPE), and the Nash-Sutcliffe Coefficient of Efficiency (NSE). MAE exhibits strong robustness against outliers and is characterized by high computational efficiency, making it suitable for regression problems. A MAE value closer to 0 signifies a lower error in the model. MAPE is also a commonly used metric in time series forecasting. It measures the relative percentage error of

predictions with respect to the actual values and is particularly sensitive to outliers. In this thesis, MAPE is presented in percentage format for consistency. NSE is a widely used metric in hydrology for evaluating predictive accuracy [117]. Similar to the Coefficient of Determination (R^2), it is primarily designed to assess whether a hydrological model performs better than a simple benchmark that uses the mean of observed values as the prediction. NSE places greater penalization on larger flow discrepancies, making it particularly effective for evaluating the impact of extreme hydrological events on model accuracy. The NSE value ranges from $(-\infty, 1]$, where negative values indicate that the model performs worse than the mean predictor, while values close to 1 suggest excellent model performance. The formulas for the three evaluation metrics are as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\left(\mathrm{y}_i^{obs} - \mathrm{y}_i^{pred}\right)\right| \tag{3.6}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\mathrm{y}_i^{obs} - \mathrm{y}_i^{pred}}{\mathrm{y}_i^{obs}}\right| \tag{3.7}$$

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{N}\left(\mathrm{y}_i^{obs} - \mathrm{y}_i^{pred}\right)^2}{\sum_{i=1}^{N}\left(\mathrm{y}_i^{obs} - \overline{\mathrm{y}^{obs}}\right)^2} \tag{3.8}$$

$\overline{\mathrm{y}^{obs}}$ and $\overline{\mathrm{y}^{pred}}$ respectively denote the mean values of the observed (actual) and predicted data. The guidelines proposed by Ritter et al. establish benchmarks for NSE values: value over 0.90 denotes a "very good" model performance. Value between 0.80 and 0.90 is considered "good", while it from 0.65 to 0.90 is viewed as "acceptable". NSE Value below 0.65 indicates "unsatisfactory" model performance [118].

The model is trained for 100 iterations, with snapshots taken at epochs 90, 95, and 100 using the Snapshot ensemble approach. For the TCN dilation factor, we adopt the

commonly used exponential scheme based on powers of 2, specifically $[1,2,4]$, which efficiently expands the receptive field with increasing depth. This setup enables the model to capture long-term dependencies with fewer layers, balancing performance and computational cost. Adam is selected as the optimizer, combining Momentum and RMSProp principles by computing exponential moving averages of gradients and variances to adapt learning rates dynamically and accelerate convergence. The implementation is based on Keras 2.1.3 and TensorFlow 1.13.1.

3.4 Experimental Results and Analysis

In the experimental section, a series of ablation studies are conducted to validate the effectiveness of each module within the EA-TCN framework. In addition, comparative experiments with mainstream neural network models are performed to demonstrate that the proposed methodology consistently outperforms existing approaches in both accuracy and efficiency.

3.4.1 Dataset

This chapter focuses on the Columbia River in the United States as its case study, with its watershed map illustrated in Figure 3.4. The Columbia River, a principal river in the western part of North America, originates from the Rocky Mountains in British Columbia, Canada, traversing through both Canada and the United States before ultimately flowing into the Pacific Ocean at Astoria, Oregon, USA. Characterized by its abundant water flow, the Columbia River has an average annual discharge of 7,860 m^3/s. During the spring snowmelt period, the discharge can surge to a maximum of 17,000 m^3/s, contributing an annual total of approximately 2.34×10^{11} m^3 to the Pacific Ocean. The Columbia River Basin encompasses a diverse range of topographical and geomorphological features, stretching from plateaus to plains, and exhibits a variety of

climatic and precipitation patterns, ranging from maritime to semi-arid climates. This variation results in significant seasonality and trends in river runoff, making it a typical non-stationary time series. Consequently, it serves as an ideal subject for training and testing various time series forecasting models, including neural networks. Moreover, the Columbia River Basin encompasses major cities such as Vancouver and Portland, housing over 400 medium and large dams, including the notable Grand Coulee Dam. These dams annually provide a substantial amount of hydroelectric power, constituting a significant source of hydropower in the United States. Additionally, the Columbia River and its tributaries support a rich aquatic ecosystem. Its famed salmon migration annually attracts a vast array of wildlife and bird species. Therefore, research on the Columbia River holds considerable economic, social, and ecological significance.

The hourly runoff dataset is sourced from the Columbia River DART (Data Access in Real Time) platform [119] [1], which provides hourly water quality data. This data is collected by the U.S. Army Corps of Engineers, Northwestern Division, and is curated, published, and maintained by the Columbia Basin Research (CBR) at the University of Washington. CBR offers researchers globally access to high-quality historical and real-time environmental data through this open-source portal. This initiative facilitates active exploration into the management and operation of large rivers and their impacts on regional hydropower, fisheries, and the ecological environment. The hourly water quality data provided by DART encompasses measurements from multiple independent water quality monitors located at various points within the watershed. For our study, the data set was obtained from the monitor designated as Canada/US Boundary (CIBW), whose specific geographic location is also indicated in Figure 3.4. This dataset records hourly outflow discharge, temperature, barometric pressure, and dissolved gas data from the monitoring station since 1997. Among these data, we selected discharge, temperature,

[1]Available at http://www.cbr.washington.edu/dart/inventory.

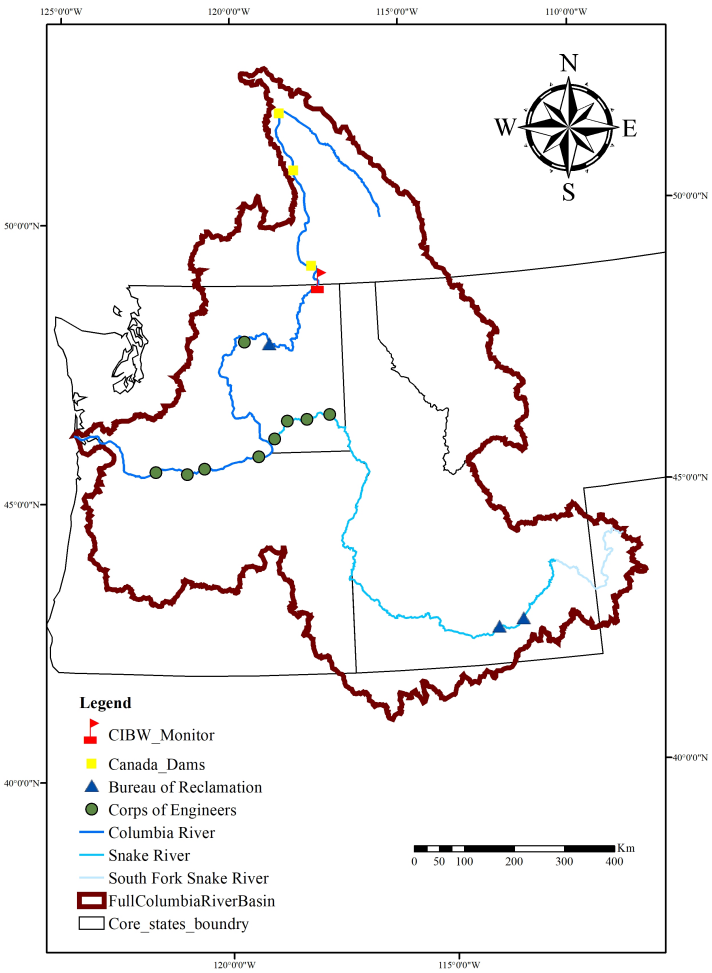


Figure 3.4: Columbia river basin schematic. The geographic location of the CIBW water quality monitor, from which the dataset for this study was collected, is marked by a red flag on the map.

and one-hot encoded seasonal data as input features for our model. The dataset utilized in this research encompasses data recorded at the CIBW monitoring station during the four-year period from 2016 to 2019, totaling 35,040 samples. Of this dataset, the data from the first three years, 2016 to 2018, which represents 75% of the total, is designated as the training set. The data from the 2019 is used as the test set.

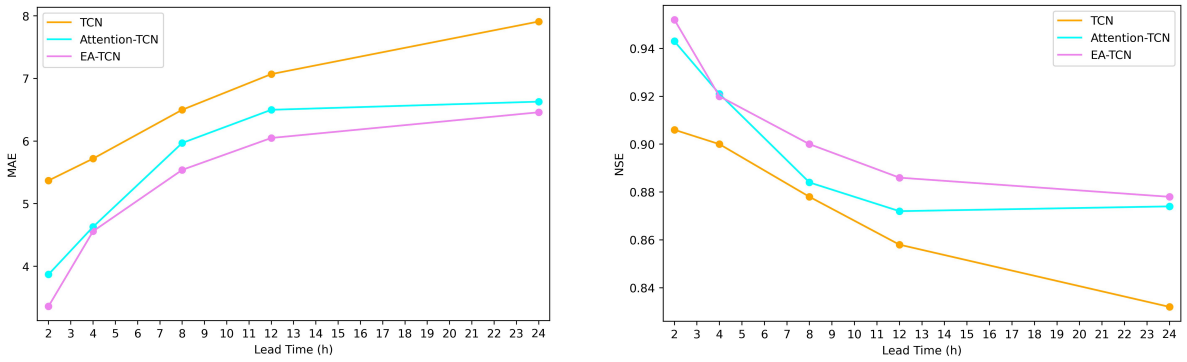
Model	Lead Time	Params	MAE	MAPE	NSE
TCN	2	134K	5.37	8.18	0.906
Attention-TCN		134K	3.87	5.24	0.943
EA-TCN (w/o res)		134K	**3.34**	4.78	**0.955**
EA-TCN (with res)		134K	3.36	**4.69**	0.952
TCN	4	134K	5.72	7.85	0.900
Attention-TCN		134K	4.63	6.45	0.921
EA-TCN (w/o res)		134K	4.71	6.63	0.914
EA-TCN (with res)		134K	**4.54**	**6.45**	**0.921**
TCN	8	134K	6.50	9.03	0.878
Attention-TCN		135K	5.98	8.13	0.884
EA-TCN (w/o res)		135K	5.96	7.99	0.886
EA-TCN (with res)		135K	**5.55**	**7.59**	**0.900**
TCN	12	134K	7.07	9.05	0.858
Attention-TCN		136K	6.50	8.58	0.872
EA-TCN (w/o res)		136K	6.19	**8.18**	0.883
EA-TCN (with res)		136K	**6.05**	8.21	**0.886**
TCN	24	134K	7.91	10.41	0.832
Attention-TCN		144K	6.63	9.28	0.874
EA-TCN (w/o res)		144K	6.82	9.27	0.869
EA-TCN (with res)		144K	**6.46**	**8.81**	**0.878**

Table 3.1: Performance comparison of models with different modules at various lead times. The best results are highlighted in bold.

3.4.2 Ablation Experiment: Performance of the Model with Different Modules

We first perform ablation studies on EA-TCN to evaluate the contribution of its individual modules. Attention-TCN denotes the variant that incorporates the SENet-based attention mechanism along the temporal dimension, while EA-TCN extends it by adding the Snapshot ensemble. The model is tested across lead times from 2 to 24 hours to assess whether the attention and ensemble components help mitigate performance decline as lead time increases.

Table 3.1 presents the ablation results, showing that EA-TCN, which combines the



(a) MAE (b) NSE

Figure 3.5: Visualization of ablation results: line charts of MAE and NSE for TCN with various modules across lead times from 2 to 24 hours.

Activation Function	Lead Time	MAE	MAPE	NSE
Sigmoid		4.96	7.09	0.913
ReLU	4	4.54	**6.45**	**0.921**
Leaky ReLU		**4.48**	6.33	0.920
Sigmoid		6.33	8.46	0.881
ReLU	12	**6.05**	**8.21**	**0.886**
Leaky ReLU		6.17	8.22	0.884
Sigmoid		7.42	9.85	0.846
ReLU	24	**6.46**	**8.81**	**0.878**
Leaky ReLU		6.58	8.88	0.875

Table 3.2: Performance comparison of different activation functions under various lead times. The best results are highlighted in bold.

attention module and Snapshot ensemble, consistently outperforms both the vanilla TCN and Attention-TCN across most lead times. Figure 3.5 further illustrates that models incorporating attention maintain high accuracy as lead time increases. For instance, at 24 hours, EA-TCN reduces MAE by nearly 20% compared to vanilla TCN, highlighting the attention modules strength in focusing on key time steps and improving long-sequence handling. While the ensemble method also enhances robustness and accuracy, its impact is slightly smaller than that of attention. Nonetheless, the Snapshot ensemble achieves these gains without adding parameters or computational cost, making its combination

Perturbation	Perturbation Level	MAE	MAPE	NSE
Attention-TCN	Mild Perturbation	7.03	10.19	0.860
EA-TCN		**6.68**	**9.36**	**0.875**
Attention-TCN	Moderate Perturbation	7.65	10.52	0.843
EA-TCN		**6.89**	**9.53**	**0.866**
Attention-TCN	Extreme Perturbation	8.09	11.06	0.824
EA-TCN		**7.47**	**10.49**	**0.846**

Table 3.3: Robustness evaluation at lead time 24 under varying perturbation levels. The best results are highlighted in bold.

with attention a practical way to improve overall model performance. The residual connections in TCN support information flow across layers, aiding backpropagation and addressing vanishing gradients. Our studies confirm that EA-TCN with residuals performs slightly better than without, emphasizing their role in boosting model efficiency. Parameter-wise, residual connections and the ensemble method contribute minimally to model size, whereas the attention module's effect varies with lead time. As described in Section 3.3.2, each attention module adds parameters computed as $[T \times \frac{T}{r} + \frac{T}{r}]$, where T is the input time steps (four times the lead time) and r is the SENet reduction ratio. Hence, the number of attention module parameters grows with lead time, but the increase is negligible for short horizons. Even at the lead time of 24, the rise is only about 7%, aligning with the module's lightweight design and allowing notable performance gains at minimal computational cost.

Activation functions are crucial to neural network performance. We therefore explore their influence on EA-TCN by comparing three widely used options: Sigmoid, ReLU (the default in this study), and Leaky ReLU. To ensure fair evaluation, experiments take place at lead times of 4, 12, and 24 hours. As shown in Table 3.2, ReLU and Leaky ReLU consistently outperform Sigmoid across all settings, largely due to their superior handling of gradient vanishing and greater computational efficiency. Performance differences

between ReLU and Leaky ReLU are minor and vary by lead time and metric. While Leaky ReLU introduces a negative slope to address the "Dying ReLU" problem, its effectiveness depends on slope choice, which, if suboptimal, can reduce performance relative to ReLU. Based on these results, ReLU is adopted as the activation function in this work.

Robustness is also a key focus of this chapter. To evaluate the robustness of the proposed model under various levels of perturbation, we introduce controlled noise into the training data. To simulate potential measurement errors from real-world hydrological sensors, we inject Gaussian noise with a mean of 0 and standard deviations of 0.01, 0.05, and 0.1, corresponding to mild, moderate, and severe perturbation scenarios, respectively. In addition, 5% of the data is randomly masked to simulate potential data missingness. Table 3.3 reports the performance of the proposed EA-TCN model with the Snapshot ensemble strategy, compared to the baseline Attention-TCN, under the three perturbation levels. The results clearly show that incorporating the ensemble method significantly improves the robustness of the model across all perturbation scenarios, with a maximum reduction in prediction error of up to 10%. This improvement is attributed to the Snapshot ensemble's capability to quickly generate multiple weak learners, which can collectively correct the prediction errors made by individual learners when subjected to noise or missing data, thereby enhancing the model's overall robustness.

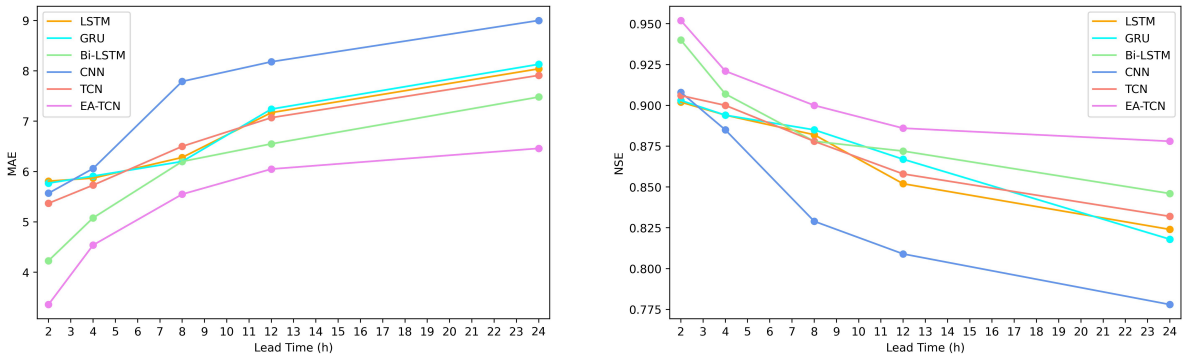
3.4.3 Comparison with Mainstream Time Series Forecasting Models

Table 3.4 shows that both recurrent architecture-based models (such as LSTM/Bi-LSTM, and GRU) and newer time series approaches like CNN and TCN consistently achieve "good" performance under the criteria of [118], highlighting the strong potential of data-driven methods in runoff forecasting.

Model	Lead Time	MAE	MAPE	NSE
LSTM	2	5.81	9.24	0.902
GRU		5.77	9.00	0.903
Bi-LSTM		4.23	6.00	0.940
CNN		5.57	7.26	0.908
TCN		5.37	8.18	0.906
EA-TCN		**3.36**	**4.69**	**0.952**
LSTM	4	5.87	7.64	0.894
GRU		5.91	8.46	0.894
Bi-LSTM		5.08	6.88	0.907
CNN		6.06	8.30	0.885
TCN		5.73	7.85	0.900
EA-TCN		**4.54**	**6.45**	**0.921**
LSTM	8	6.28	8.59	0.882
GRU		6.20	8.41	0.885
Bi-LSTM		6.20	8.51	0.878
CNN		7.79	10.57	0.829
TCN		6.50	9.03	0.878
EA-TCN		**5.55**	**7.59**	**0.900**
LSTM	12	7.17	9.38	0.852
GRU		7.24	10.86	0.867
Bi-LSTM		6.55	9.01	0.872
CNN		8.18	10.93	0.809
TCN		7.07	9.05	0.858
EA-TCN		**6.05**	**8.21**	**0.886**
LSTM	24	8.04	10.35	0.824
GRU		8.13	12.12	0.818
Bi-LSTM		7.48	9.91	0.846
CNN		9.00	11.59	0.778
TCN		7.91	10.41	0.832
EA-TCN		**6.46**	**8.81**	**0.878**

Table 3.4: Comparison with mainstream neural networks at different lead times. Best performance for each group is highlighted in bold.

LSTM, GRU, and TCN demonstrate similar performance across lead times, but TCN's streamlined architecture and parallel computation provide greater efficiency, giving it an advantage over recurrent models. For long time series, GRU shows errors up to 8.13 and



(a) MAE (b) NSE

Figure 3.6: Visualization of comparative results: line chart of MAE and NSE for neural network models across lead times from 2 to 24 hours.

an NSE as low as 0.818, mainly due to difficulties in error backpropagation along time steps, including vanishing gradients. In contrast, TCN propagates errors along depth and uses dilated convolutions to expand its receptive field, enabling better handling of long sequences. Our results also show that the vanilla CNN performs worse than both TCN and recurrent models for extended inputs because its limited receptive field restricts learning. TCN's use of dilated convolutions effectively addresses this limitation. At the 2-hour lead time, TCN and CNN achieve similar results, likely because TCN lacks sufficient sequential information at short horizons while CNN's absence of causal convolution may lead to information leakage. These findings suggest that both models perform comparably in short-term contexts. The temporal attention mechanism allows EA-TCN to focus on key time steps, and the ensemble method enhances robustness and generalization. As shown in Figure 3.6, EA-TCN consistently performs well across lead times, reaching an NSE of 0.877 at 24 hours, demonstrating strong accuracy and stability.

As a powerful LSTM variant, Bi-LSTM demonstrates strong performance across lead times, often outperforming TCN thanks to its bidirectional processing, which allows simultaneous capture of past and future dependencies. This enhances its ability to learn

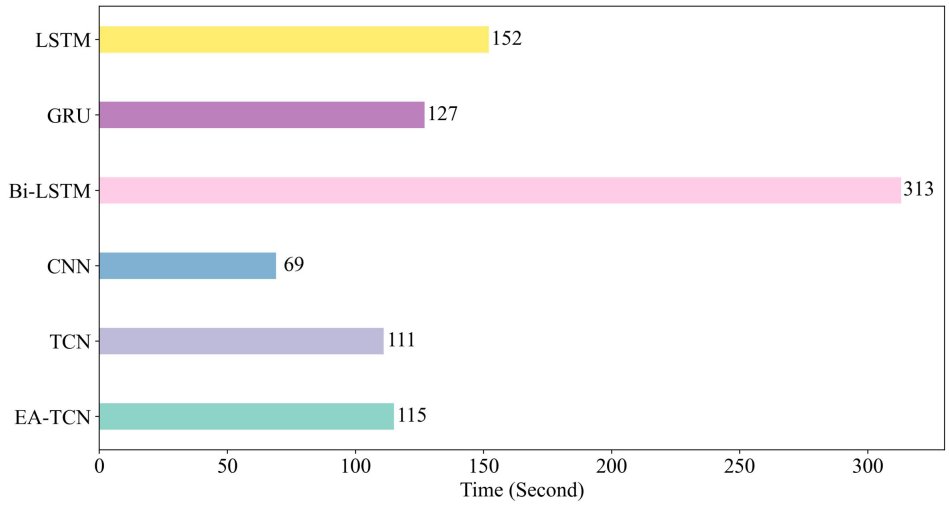


Figure 3.7: Training time of mainstream models for long time series forecasting.

complex sequential features. However, the improved accuracy comes with significant computational cost, particularly for long sequences. As shown in Figure 3.7, Bi-LSTM requires substantially longer training time at a lead time of 24 compared to other models, with LSTM also showing high time consumption. GRU improves efficiency through its simplified gating but sacrifices accuracy at certain lead times relative to LSTM. TCN achieves the fastest training due to parameter sharing and parallelism, making it highly efficient for long sequences. While EA-TCN introduces a modest 3.6% increase in computational time over TCN by adding attention and ensemble modules, it achieves an 18.3% accuracy gain, highlighting its strong balance of accuracy and efficiency.

3.5 Conclusion

This chapter presents EA-TCN as a solution to the limitations of recurrent architecture-based neural networks in runoff forecasting, including weak long-term modeling, lack of parallelism, and poor generalization. By integrating TCN, which supports parallel computation, with the lightweight SENet attention module, EA-TCN explicitly captures temporal dependencies. The Snapshot ensemble further strengthens generalization and

robustness without adding significant computational cost. Experiments on the Columbia River dataset confirm the effectiveness of the design for multi-lead-time short-term runoff forecasting. Comparative analyses with recurrent models highlight the superior accuracy, efficiency, and robustness of EA-TCN. The attention and ensemble components notably enhance TCN's performance across lead times of 2, 4, 8, 12, and 24 hours, enabling accurate predictions for sequences of varying lengths and overcoming the long-term memory limitations of LSTM. The model's parallel structure and reduced parameter count contribute to both accuracy and computational efficiency, setting it apart from recurrent architectures.

The proposed approach offers strong potential for practical deployment and further development. Its ability to capture complex temporal patterns supports applications such as extreme event forecasting, flood management, and water resource optimization. Its capacity for handling long sequences makes it promising for simulating hydrological cycles and watershed dynamics. Nonetheless, there are areas for improvement. Despite TCN's inherent efficiency, the added attention module and increased depth limit its computational advantage over recurrent models. Additionally, performance degrades as lead time extends from 2 to 12 hours, revealing challenges in temporal robustness. Finally, the black-box nature of neural networks limits interpretability. Future work will focus on improving efficiency, strengthening temporal robustness, and enhancing model transparency to foster stakeholder trust and broaden real-world applicability.

CHAPTER 4

RESIDUAL TEMPORAL CONVOLUTIONAL NETWORK WITH DUAL-PATH SPATIOTEMPORAL ATTENTION MECHANISM

RQ2 highlights the structural limitations inherent in current neural network-based runoff forecasting models, which lead to significant performance bottlenecks. In addition, RQ4 also points out that the lack of interpretability has a serious impact on the practical application of the models. To address these problems and fulfill the objectives of RO2 and RO4, this chapter introduces an innovative hybrid model, ResTCN-DAM, which synergizes the strengths of ResNet, TCN, and dual-path spatiotemporal attention mechanism (DAM). ResTCN-DAM is designed to fully exploit the unique properties of these three components. TCN offers strong parallel processing capability for time series data while maintaining a lightweight architecture. By integrating with an enhanced ResNet, multiple TCN layers can be densely stacked to capture higher-level abstract features in the temporal dimension. The proposed DAM module further generalises the traditional focus on a single dimension to multiple dimensions, being able to capture interdependencies between features within both the temporal and spatial dimensions, and subtly highlight relevant time steps/features while weakening less important features

with minimal computational cost. Moreover, ResTCN-DAM incorporates the Snapshot ensemble method, which enables the model to approximate the benefits of training multiple models within one single training cycle, thereby ensuring both accuracy and robustness. The deep integration and synergy among these modules collectively enhance the model's forecasting performance from multiple perspectives. To further improve interpretability, the model-specific local post-hoc explanation technique is employed. Ablation studies prove the effectiveness of each module, and extensive comparative experiments demonstrate that ResTCN-DAM achieves superior accuracy, temporal robustness, and interpretability compared to existing neural network-based runoff forecasting models, effectively overcoming current performance bottlenecks.

Section 4.1 outlines the challenges, motivations, and key innovations of this study. Section 4.2 presents the proposed methodology in detail. Section 4.3 validates the effectiveness and advancement of the methodology through ablation and comparative experiments. Finally, Section 4.4 concludes the chapter and highlights directions for future research.

4.1 Introduction

Hourly runoff forecasting is now widely applied in real-time forecasting scenarios that require rapid responses to sudden hydrological events [120]. In high-risk situations such as flood prevention and emergency water management [121], the accuracy of model forecasts directly influences decision-making by authorities [122]. Short-duration heavy rainfall and rapid runoff variations often exhibit steep rises and falls, frequent extremes, and sudden transitions that can quickly trigger flooding or urban waterlogging. In such cases, low forecasting accuracy can delay emergency response and result in significant threats to public safety and property. In high-frequency short-term operation scenarios such as reservoir regulation and hydropower scheduling [123, 124], forecasting accuracy

is closely tied to cost control and risk management. Higher accuracy enables more efficient scheduling, thereby reducing operational costs. As a result, forecasting at the hourly scale imposes much stricter accuracy requirements compared to monthly or yearly runoff forecasting [125]. Although the EA-TCN proposed in Chapter 3 achieves a great balance among accuracy, efficiency, and robustness, the research also reveals clear performance bottlenecks in existing mainstream models. When the forecasting horizon exceeds 8 hours, almost all models experience a noticeable decline in performance metrics; for some models, MAPE exceeds 10%, and NSE commonly falls below 0.9. With the ongoing advancement of monitoring technologies and rapid iteration of machine learning models, stakeholders in the hydrological domain are demanding increasingly higher accuracy in hourly forecasting [126]. Moreover, several engineering guidelines have started to incorporate quantitative accuracy benchmarks for runoff forecasting, prompting the need for further architectural improvements to meet these rising expectations.

In the field of machine learning, numerous powerful architectures are continually being proposed to enhance model performance. With the continuous expansion of the scale of the neural network model, the feature quantity of the model also gradually increases. In order to apply limited computing resources to more important features, inspired by the way humans observe external things, attention mechanism is proposed to improve the performance of neural network models [127, 128]. Attention mechanism can adaptively learn the weight of each feature in a specific dimension through soft information extraction, and the weight indicates how much the model pays attention to the input information. Han et al. present a runoff forecasting model AT-LSTM [129] that integrates attention mechanism with LSTM. The proposed model embeds attention mechanisms at both the input and hidden layers, enabling it to dynamically identify and assign weights to key factors in real-time. AT-LSTM is applied to long-term runoff forecasting at the Yichang and Pingshan stations in the upper Yangtze River in China.

Experimental results demonstrate that the performance of AT-LSTM significantly surpasses that of traditional LSTM, thereby thoroughly validating the effectiveness of the attention mechanism. Attention mechanism is now moving towards lightness, more and more plug-and-play attention mechanism modules are proposed, which can be embedded in any position in the network to significantly improve the expressive capability with minimal computational cost.

Nevertheless, we observe that in current research, the attention mechanism is typically designed to operate on a single dimension, rendering the existing unidirectional attention mechanisms insufficient for efficiently handling time series inputs that incorporate multiple dimensional information. Furthermore, due to the problem of degradation, existing LSTM or TCN-based models are unable to construct deep architectures to further capture high-order abstract features and long-term dependencies within long sequences. The presence of these issues poses a significant challenge for existing models, primarily their lack of temporal robustness. Neural network-based short-term runoff forecasting models often demonstrate commendable forecasting accuracy over singular and relatively short forecast horizons. However, as the forecast horizon extends from a few hours to 24 hours in the future, there is a significant degradation in model performance, rendering it challenging to maintain robust and precise forecasts across varying lead times. For instance, the hourly runoff forecasting model based on TCN proposed by Lin et al. exhibits a serious performance degradation when the forecast horizon extends from t+1 to t+24 [68]. Lastly, it is worth noting that current research on runoff forecasting models is mainly performance-oriented, often overlooking model interpretability [130, 131]. This has resulted in a lack of trust from stakeholders in high-risk areas towards these high-accuracy but low-transparency models. In light of the aforementioned problems, we propose an ensembled multi-lead-time runoff forecasting framework: ResTCN-DAM, which is characterized by a deep densely connected residual

structure, and simultaneously employs attention mechanism in both the temporal and spatial dimensions. The key contributions of this chapter are outlined as follows:

- We innovatively integrate TCN with a deep densely connected residual structure to replace the neural network based on the recurrent architecture as the overall framework. TCN has the advantages of parallel processing and strong feature extraction capabilities, while ResNet Plus further improves model depth through optimized information flow, enabling the learning of high-level abstract features. This integration significantly enhances the forecasting accuracy of the proposed framework.
- Each residual block is embedded with a plug-and-play spatiotemporal attention module DAM, which generalizes lightweight attention mechanisms from a single dimension to the most critical temporal and spatial dimensions in time series forecasting models. This enables the model to adaptively and explicitly model the importance of features across different dimensions, thereby improving long-sequence forecasting accuracy with minimal computational overhead.
- To enhance the interpretability of the forecasting process, the proposed model introduces, for the first time in hourly runoff forecasting, a model-specific local post-hoc explanation technique from interpretable machine learning (IML). This technique enables intuitive visualization and analysis of DAM's attention weights via heatmaps, ensuring that the model's predictions remain human-understandable and trustworthy.

4.2 Methodology

The hourly runoff forecasting framework ResTCN-DAM proposed in this chapter consists of three sub-modules: modified ResNet – ResNet Plus, TCN, and DAM. The overall

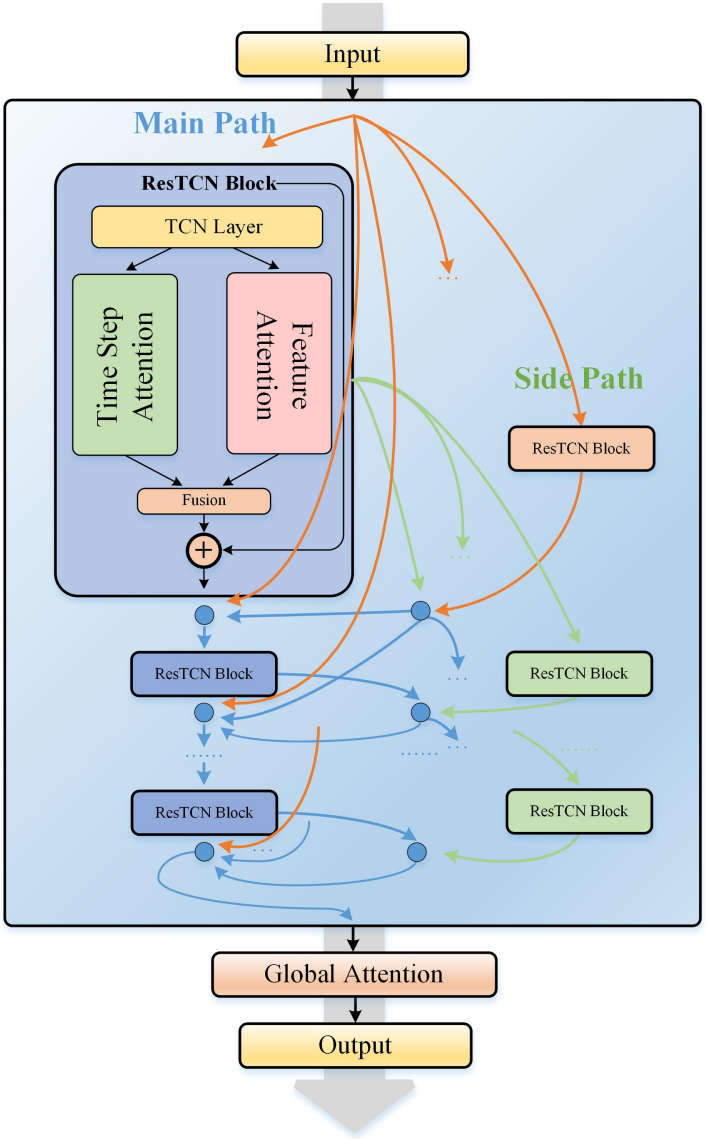


Figure 4.1: The overall architecture of ResTCN-DAM. Each ResTCN block comprises a TCN followed by a DAM module, which explicitly models both the temporal dimension and the feature dimension. Residual blocks are sequentially stacked in a head-to-tail configuration, forming two pathways connected by dense shortcut connections.

framework of ResTCN-DAM can be seen in Figure 4.1. First, a standard convolutional layer will perform preliminary feature extraction on the input, and then pass the output to the subsequent ResTCN-DAM network for further processing.

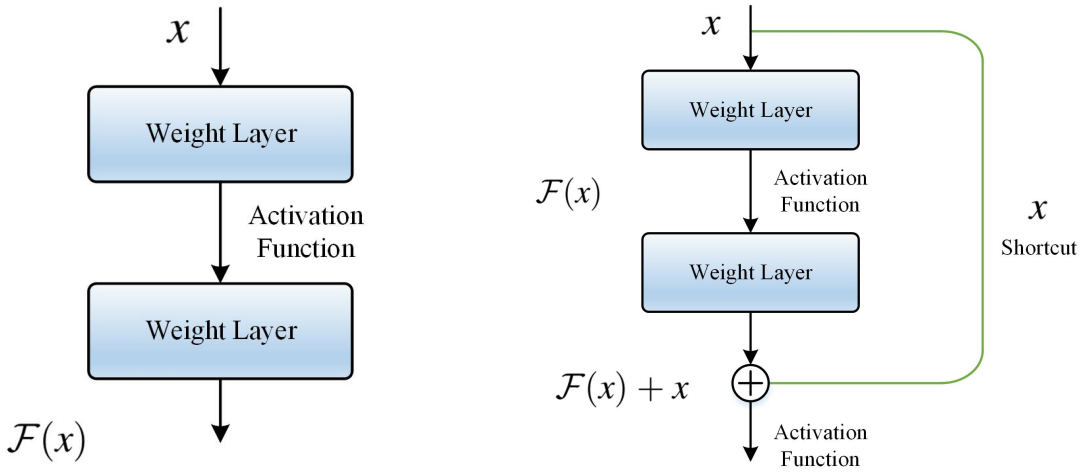
In terms of network architecture, ResTCN-DAM is comprised of four layers of residual

blocks distributed across two distinct paths, with dense shortcut connections established both within and between the residual blocks, which is inspired by the modified version of ResNet, termed ResNet Plus. This configuration significantly elevates the network's depth and backpropagation efficiency relative to the conventional ResNet. Inside each residual block, TCN layers are used to forecast the time series inputs. Two lightweight attention mechanism modules DAM are also inserted after TCN layers. By learning the weight of TCN in temporal dimension and feature dimension respectively, the network can focus on important time steps, amplify useful features, and suppress useless features.

Moreover, NoCycle snapshot ensemble method is incorporated into the ResTCN-DAM framework, allowing us to obtain multiple models for integration via a singular training process to improve model's robustness and generalization capability. In summary, the proposed ResTCN-DAM framework can jointly enhance the performance of runoff forecasting model from multiple aspects such as depth, time step, features and robustness to achieve accurate runoff forecasts. In this section, each module of ResTCN-DAM will be presented in detail.

4.2.1 From ResNet to ResNet Plus

ResNet is a structure proposed to solve the degradation problems in deep neural networks. The proposal of the residual network originated from a concept: when a neural network has the best performance when the number of layers is L, then the network can continue to be deepened on the basis of the L layer. In theory, if the additional layers are the identity mapping of the Lth layer's output, then the performance of the deeper network can be consistent with or even better than that of the original network. In general, the performance of deep neural network should not be outperformed by its shallow counterpart. However, experiments show that the training error and test error of the 56-layer 'plain' network is significantly higher than that of the 20-layer network



(a) The ordinary block.

(b) The ResNet block.

Figure 4.2: The ordinary block and ResNet block. The characteristic that distinguishes ResNet blocks from ordinary blocks is this shortcut connection marked in green, which can transmit the information of the shallow network to the deep layer of the network intact.

on the CIFAR-10 dataset [132]. This phenomenon is the degradation problem in deep neural networks, it is caused by many factors. In addition to the widely known vanishing/exploding gradient, Sandler et al. believe that the existence of nonlinear activation function makes the forward propagation process from input to output irreversible [133], which leads to a decrease in the amount of information acquired by the deep layers in the network.

The existence of the degradation problem proves that deep neural networks cannot be well trained, but the proposal of ResNet makes it possible to address the degradation problem and train deep neural networks. The structure of ResNet is shown in Figure 4.2. Compared with the traditional neural network, ResNet introduces a novel shortcut that directly links the input to the output of nonlinear layers, facilitating an unimpeded flow of information. The core formula of ResNet is:

$$\mathscr{H}(x) = \mathscr{F}(x, W) + x \tag{4.1}$$

where x and W are input and weight respectively. Different from directly fitting a potential identity mapping function $\mathscr{H}(x) = x$, ResNet transforms the learning process into fitting a residual function = $\mathscr{F}(x, W) = \mathscr{H}(x) - x$. As long as $\mathscr{F}(x, W)$ is 0, it constitutes the identity mapping $\mathscr{H}(x) = x$. The Lth layer of ResNet can be expressed as

$$x_L = x_l + \sum_{i=l}^{L-1} \mathscr{F}(x_i, W_i) \tag{4.2}$$

The deep layer x_L can be divided into two parts, one is the shallow layer x_l, and the other is the mapping of the residual function$\sum_{i=l}^{L-1} \mathscr{F}(x_i, W_i)$, which indicates that the model is in the form of residuals at any layer. Because it is easier to fit residuals, ResNet with shortcut connections have better ability to fit functions than traditional neural networks. Moreover, the existence of shortcut connections allows data in shallow layers to be transmitted to the deeper layers of the network intact, which makes a $L+1$-layer network must contain more information than a L-layer network. The back propagation of ResNet can be formulated as:

$$\frac{\partial loss}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \frac{\partial x_L}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \left(1 + \frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} \mathscr{F}(x_i, W_i)\right) \tag{4.3}$$

where $loss$ represents the cost function. When performing backpropagation, the gradient is composed of two parts, one part is the gradient $\frac{\partial loss}{\partial x_L}$ through the shortcut $\mathbf{1}$, and the other part is the gradient $\frac{\partial loss}{\partial x_L} \cdot \left(\frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} \mathscr{F}(x_i, W_i)\right)$ through the weight layer. The existence of the shortcut connection allows the gradient of the deep layers to be backpropagated to the shallow layers of the network intact, and the weight terms $\left(\frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} \mathscr{F}(x_i, W_i)\right)$ will not all be -1, thus avoiding the vanishing gradient. Relying on the superiority of the fitting function and training process, ResNet is able to train a neural network with even one thousands layers, so it has become a milestone in the field of deep learning.

ResNet has made researchers aware of the importance of shortcut in neural networks. In order to create short paths from shallow layers to deep layers, many ResNet variants based on modified shortcuts have been proposed, such as DenseNet, which establishes dense shortcuts that connecs all layers in the network. Inspired by the densely connected shortcuts of DenseNet, this chapter utilizes a novel modified ResNet: ResNet Plus [134], its structure can be seen in Figure 4.1. The characteristic of ResNet Plus is to have two residual block paths, the left one is the main path, and the right one is the side path. The input of the residual block in the first layer on both main path and side path is the input of the entire network. Except for the first layer, the input of each residual block on the side path is the output of the first residual block on the main path. The output of the residual block on two different paths in the same layer will be averaged and densely connected with the subsequent residual blocks on the main path, so the input of each residual block on the main path comes from the output of all previous residual blocks. With additional residual block path and denser shortcuts, feature reuse is further enhanced. Crucially, ResNet Plus replaces the single path of traditional ResNet with a dual-path approach. This ensures that the output of any layer results from a fusion of abstract features from different levels across both paths, significantly improving both forward and backward information flow and facilitating easier training. The incorporation of side residual blocks shortens the path from input to output with the same number of residual blocks, reducing errors and improving backpropagation efficiency due to the shorter paths. Furthermore, the main and side paths are not isolated, residual blocks at each layer within the two paths are merged through the averaging operation. This architecture increases the width (number of channels) of each layer in a relatively straightforward manner, and this width-for-depth strategy enhances model performance cost-effectively.

ResNet Plus can help us train deeper networks to learn more hidden features. Time series forecasting method such as LSTM, GRU or TCN usually contains a large number

of parameters. Therefore, although the hidden relationship between time series inputs can be captured well, the network depth cannot be effectively improved, which limits their performance. In order to further utilize the depth advantages of ResNet Plus for runoff forecasting, we have deeply integrated the advanced time series forecasting method TCN with ResNet Plus. Each ResTCN block contains one core TCN module for time series forecasting, followed by an inserted lightweight DAM module proposed by us. The DAM is designed to model the interdependencies between time steps and features with minimal computational cost. Notably, between TCN and DAM, batch normalization (BN) [135] and dropout [136] are incorporated to reduce internal covariate shift and enhance regularization capability. This lightweight-oriented design significantly reduces the parameter count within each block, thereby diminishing the risk of overfitting and allowing for dense stacking of ResTCN blocks through ResNet Plus. Beyond the shortcut connections between blocks, each ResTCN block internally establishes a shortcut connection linking its input and output. Additionally, within each TCN module, convolutional shortcut connection is implemented, connecting the input and output layers. This strategy forms a three-tiered network of dense shortcut connections, radiating outward from the core, enhancing the efficiency of backpropagation when compared to the traditional ResNet. The integrated architecture of ResNet Plus deeply fused with TCN substantially elevates the model's time series forecasting performance as the network deepens. In the experimental section, the efficacy of this architecture is thoroughly validated through ablation studies.

4.2.2 Dual-Path Spatiotemporal Attention Mechanism

Attention is a mechanism that is widely used in CV and NLP to improve the performance of machine learning models, it exhibits similarities to the human observational mechanism of external phenomena. When humans observe external things, they tend not to

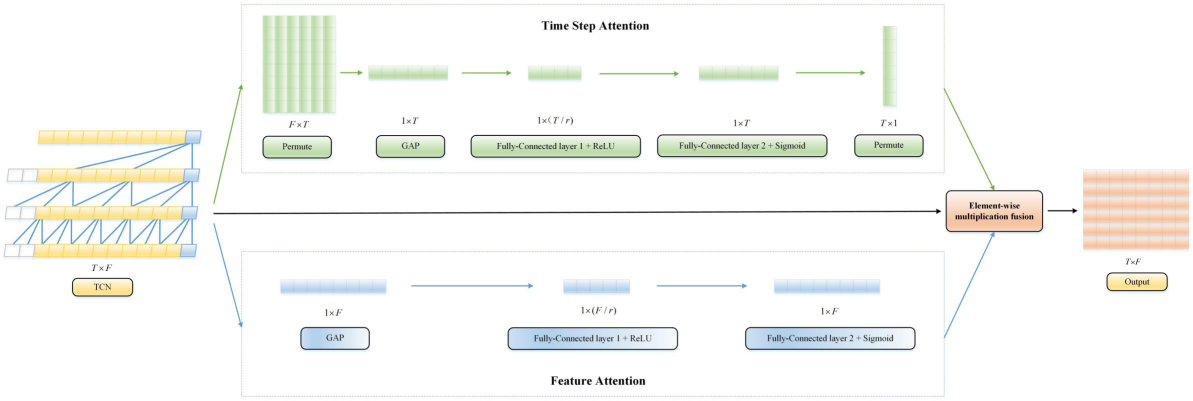


Figure 4.3: The structure of dual attention mechanism. This module adaptively and explicitly models the interdependencies among input elements across both the time step and feature dimensions in an efficient manner. It recalibrates the original output of the TCN in the form of weights. Thanks to its lightweight structural design and parallel processing approach, the module significantly enhances the model's performance with a minimal computational cost.

evenly distribute their attention to the whole, but to pay attention to an important part of things. Attention mechanism has the capability to assign varying weights to distinct segments of inputs. By learning these weights, model can allocate more attention to valuable information while disregarding less relevant information, thereby significantly improving the performance of the model at the cost of a small amount of calculation.

Attention mechanism is also applied in ResTCN. Unlike the traditional attention mechanism that only calculates one specific dimension of the model input, we innovatively propose the DAM module to establish the attention mechanism for the two most important dimensions of time series forecasting models: time steps and features. It can be seen from Figure 4.3 that ResTCN-DAM integrates the lightweight attention mechanism module SENet into each residual block to explicitly model the interdependence between time steps/features, so that network can adaptively amplify useful information and suppress useless information in these two dimension. In addition, another global attention module [137] is added to the output sequence of the network. By comparing the target hidden state at the last time step with the hidden state of each time step, the

attention output of the network is obtained through weighted average and concatenation operations.

SENet was originally designed to adaptively recalibrate channel-wise feature responses in convolutional neural networks. With its lightweight structure and easy deployment, SENet is widely used in various fields. Figure 4.3 demonstrates the structure of DAM composed of improved SENet, we can see that the attention pathway in each dimension consists of two operations: squeeze and excitation. First, a GAP layer is used for the squeeze operation. Similar to LSTM, TCN will output a two-dimensional vector in the shape of $[T \times F]$ when transferring information between layers, where T is the time step, and F represents the number of filters (similar to the units of LSTM). GAP will compress the global features output by TCN into a vector of size $[1 \times T]$ or $[1 \times F]$ according to the temporal/feature dimension, that is, compress each one-dimensional temporal dimension/feature dimension into a real number with a global receptive field that reflects the overall response distribution across time steps or features. Squeeze of the temporal dimension can be formulated as:

$$\tilde{x_1} = \frac{1}{f} \sum_{i=1}^{f} x_1^i \tag{4.4}$$

where f is the number of features. Similarly, squeeze of the feature dimension can be expressed as:

$$\tilde{x_1} = \frac{1}{t} \sum_{i=1}^{t} x_1^i \tag{4.5}$$

where t is the number of time steps. After squeeze, the excitation operation will explicitly model the interdependencies between time steps/feature dimensions. A bottleneck structure containing two fully-connected layers (FC) is used to perform the excitation operation. Excitation can be expressed as:

$$\hat{x}_1 = \sigma(W_2\delta(W_1\tilde{x_1})) \tag{4.6}$$

where δ indicates the ReLU function and σ denotes the Sigmoid function. W_1 and W_2 represent the weight matrices corresponding to the two fully-connected layers on each attention pathway. The first fully-connected layer can reduce dimension by setting scaling parameters, thereby greatly reducing the amount of parameters of DAM, and is then activated by the activation function. The second fully connected layer restores the dimensions and converts the results into weights by Sigmoid function, which represents the importance of each time step/feature. Finally, the results are multiplied by the original output of TCN to re-calibrate the original time steps/features. With the improved DAM module based on SENet, ResTCN can selectively amplify valuable time steps/features and suppress useless time-steps/features at the cost of a small increase in computation, thus improving the performance of the model. It is worth noting that DAM is designed to be embedded at the output of the model rather than at the input because it relies heavily on the global information in the high-level abstract feature maps generated after the feature extraction phase for precise recalibration, and thus cannot perform on raw unprocessed features. Building on this, the proposed DAM possesses a plug-and-play attribute, allowing it to be seamlessly integrated into almost any mainstream neural network model, including CNN, LSTM and TCN. This enables adaptive recalibration of features across multiple dimensions, demonstrating remarkable flexibility and generalization potential.

4.2.3 Global Attention Module

It is worth mentioning that when time series information is transmitted between different TCN layers, the sequence passed to the next layer contains the outputs of all time steps, and the attention module SENet will recalibrate the output of each time step based

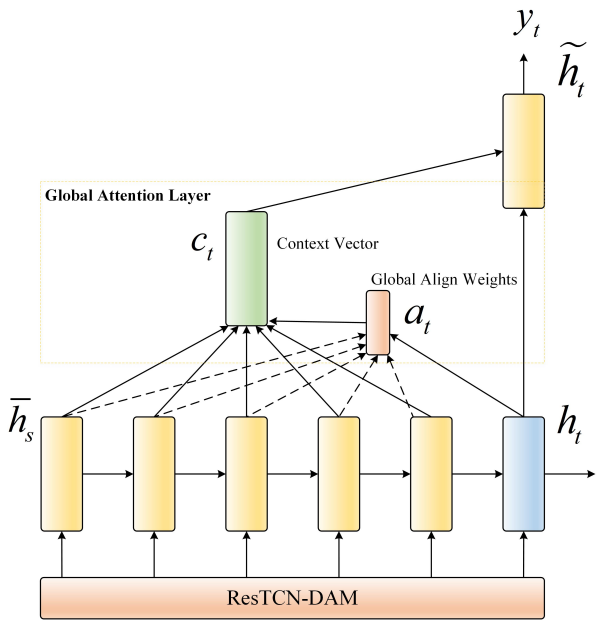


Figure 4.4: The structure of global attention module. This module is inserted into the output part of the model to enhance the final forecast's sensitivity and correlation to the global hidden states.

on the learned weights. However, when the last TCN layer outputs the final forecast, it only retains the output of the last time step and ignores the output of any other time step. Based on the different information flow transmission methods, the attention mechanism of the temporal dimension needs to be adjusted. Another attention mechanism module called global attention is applied to the final output sequence of ResTCN-DAM to capture and calculate the interdependence between the hidden state of the last time step and the hidden states of previous time steps. The integration of global attention endows the hidden state at the final time step with a comprehensive perception of global information across the entire hidden state sequence. This characteristic facilitates an enhancement of the final forecast's accuracy from a holistic perspective. Figure 4.4 demonstrates the structure of global attention module, it combines the hidden state of the last time step h_t and the context vector c_t through a simple concatenation layer to obtain the attention

vector $\tilde{h_t}$. The formula of the context vector c_t is:

$$c_t = \sum_s a_{ts} \overline{h}_s \tag{4.7}$$

The alignment vector a_t is calculated as the weight, and the context vector c_t is the weighted average over the hidden states of all time steps. The attention vector $\tilde{h_t}$ can be formulated as:

$$\tilde{h_t} = f(c_t, h_t) = \tanh(W_c[c_t; h_t]) \tag{4.8}$$

This module is a global attention model, so the model considers all hidden states of the output sequence when deriving the context vector c_t. It compares the target hidden state h_t with the hidden state of each previous time step $\overline{h}_s$ to obtain a variable-length alignment vector a which can be expressed as:

$$a_{ts} = \text{align}\left(h_t, \overline{h}_s\right) = \frac{\exp\left(\text{score}\left(h_t, \overline{h}_s\right)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}\left(h_t, \overline{h}_{s'}\right)\right)} \tag{4.9}$$

where score is determined by a content-based function formulated as:

$$\text{score}\left(h_t, \overline{h}_s\right) = h_t^\top W \overline{h}_s \tag{4.10}$$

The global attention module effectively improves the ability of the final output of the model to perceive the global information, and its synergy with the DAM module can further enhance the expressive power and temporal robustness of the proposed ResTCN-DAM, so as to realize the accurate forecast of runoff across various lead times.

4.2.4 Model Design and Implementation Details

In this chapter, a novel runoff forecasting framework ResTCN-DAM based on ResNet, TCN and DAM for hourly river runoff forecasting is innovatively presented. The proposed framework is capable of accurately forecasting hourly river runoff within different lead times of 2, 4, 8, 12, and 24 hours. To ensure the model can fully capture the

intricate temporal dependencies contained within historical data, the input time steps are configured to be three times the length of the forecast horizons. In the proposed ResTCN-DAM architecture, each TCN module encompasses a 4-layer TCN network with dilation factor of d = 1,2,4. However, when the lead time is greater than or equal to 8, the dilation factor is adjusted to d = 1,2,4,8, and the number of TCN layers in each module is increased to 5. This augmentation facilitates the modeling of longer input sequences, enhancing the network's capacity to capture extended long-term dependencies.

The loss function of the model is MSE and the evaluation metrics are MAE, MAPE and NSE which are widely used in hydrological models.The number of model training iterations is 100. When training reaches 90 epochs, the model's error decreases to a lower level and tends towards convergence. Therefore, we take snapshots of the model at 90, 95, and 100 epochs and save the model parameters for ensemble integration. The Adam optimizer is utilized, combining features of both momentum and root mean square propagation optimizers, thereby allowing for adaptive adjustments of the learning rate. The study area and dataset used in this chapter are consistent with Chapter 3, both based on hourly hydrological data from the Columbia River in the United States. Our experiments are conducted in an environment using Keras 2.1.3, based on Tensorflow 1.13.1, with Python version 3.7.0.

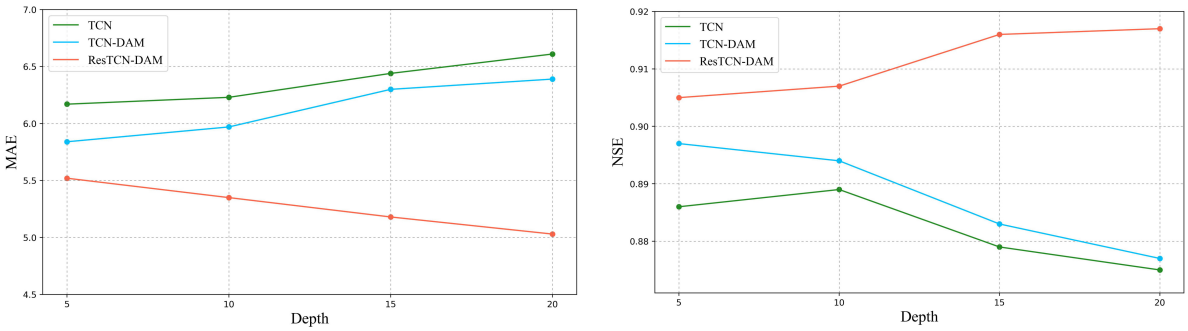
4.3 Experiment Results and Discussion

4.3.1 Validity and Compatibility of Each Module of ResTCN-DAM

First, we design a series of ablation experiments to individually validate the effectiveness of each key module in the proposed ResTCN-DAM. Specifically, these experiments test the overall performance of vanilla TCN, TCN with DAM module (TCN-DAM), and ResTCN-DAM at various model depths, as reflected by evaluation metrics: MAE, MAPE and

Model	Depth	MAE	MAPE	NSE
TCN		6.17	8.91	0.886
TCN-DAM	5	5.84	8.32	0.897
ResTCN-DAM		**5.52**	**7.74**	**0.905**
TCN		6.23	8.63	0.889
TCN-DAM	10	5.97	8.21	0.894
ResTCN-DAM		**5.35**	**7.58**	**0.907**
TCN		6.44	8.80	0.879
TCN-DAM	15	6.30	8.62	0.883
ResTCN-DAM		**5.18**	**7.26**	**0.916**
TCN		6.61	9.25	0.875
TCN-DAM	20	6.39	8.30	0.877
ResTCN-DAM		**5.03**	**7.18**	**0.917**

Table 4.1: Performance of models with different modules at various depths. Best results in each group are highlighted in bold.



(a) MAE (b) NSE

Figure 4.5: MAE and NSE of models with different modules at various depths in ablation experiments.

NSE. It's important to note that the depth mentioned here refers solely to the number of TCN layers in the preliminary design of the model, providing a clear indication of how the stacking of TCN layers affects model performance. The actual depth of these models, including various hidden layers, ranges between 100 to 300 layers. To maintain consistency in hyperparameters, all models in the experiment are set with the lead time of 8. Table 4.1 and Figure 4.5 show the results of the ablation experiments. We observe that as the number of layers increases, TCN and TCN-DAM without the ResNet architecture

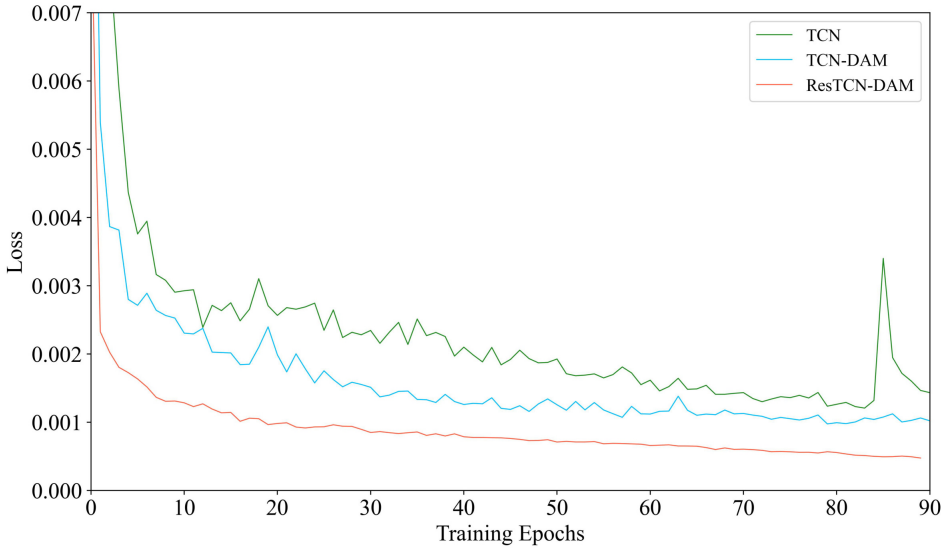
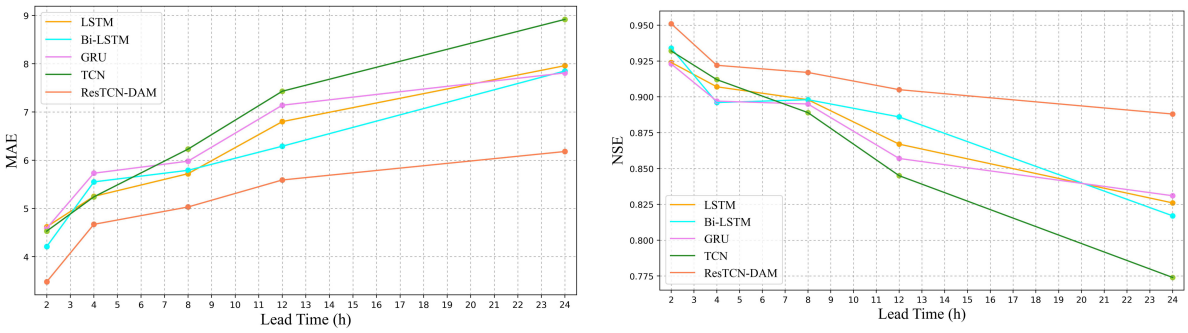


Figure 4.6: Training loss curves of models with different modules at depth of 20 in ablation experiments.

do not exhibit significant changes in accuracy between 5 to 10 layers, and even show some improvements in certain metrics. This indicates that at shallower depths, traditional model structures are capable of learning valuable information from an increase in depth, with minimal negative impact on the overall model. However, as the model depth increases from 10 to 20 layers, TCN and TCN-DAM experience a marked decline in performance, with errors rising by up to 9.4%. This is due to the problems of degradation and vanishing/exploding gradient associated with increased depth, which significantly limit the performance of the models, leading to a scenario where deeper networks perform worse than shallower ones. In contrast, our ResTCN-DAM, featuring a tri-layer dense shortcut connection design from the inside out, shows continual improvement in performance with increased depth, achieving up to an 8.9% increase in accuracy. This demonstrates the ability of the ResNet architecture to effectively address the problems of degradation in deeper models, enabling them to learn more higher-order abstract features through deeper networks.

In comparing TCN with TCN-DAM, it is evident that across all depths, the DAM



(a) MAE (b) NSE

Figure 4.7: MAE and NSE different models at various lead times in comparative experiments.

module significantly enhances the model's accuracy, reducing errors by over 5%. Additionally, TCN-DAM demonstrates notable improvements in NSE metric compared to vanilla TCN. These experimental results indicate that our proposed DAM module effectively models the interdependencies between information across the dimensions of time and space. It encourages the model to focus more on useful time steps and features from a global information perspective, thereby substantially enhancing its forecasting capabilities. Figure 4.6 also clearly indicates that the joint implementation of ResNet architecture and DAM significantly improves the overall performance of ResTCN-DAM. This improvement is reflected in its reduced loss and increased training efficiency, firmly substantiating the efficacy and advanced nature of the proposed modules.

4.3.2 Comparison with Mainstream Machine Learning Models

In the subsequent experiments, we present a comparative experimental analysis of our proposed ResTCN-DAM with mainstream recurrent architecture-based runoff forecasting models such as LSTM, Bi-LSTM [138], and GRU across various lead times. The results showed in Table 4.2, are further visualized in Figure 4.7. It is evident from the experimental results that ResTCN-DAM consistently achieves significantly superior accuracy over other models across all forecast horizons. At shorter lead times, nearly all

Model	Lead Time	MAE	MAPE	NSE
LSTM	2	4.62	6.21	0.924
Bi-LSTM		4.21	5.47	0.934
GRU		4.59	6.19	0.923
TCN		4.53	6.40	0.932
ResTCN-DAM		**3.48**	**5.02**	**0.951**
LSTM	4	5.25	7.36	0.907
Bi-LSTM		5.55	7.40	0.896
GRU		5.73	7.97	0.897
TCN		5.24	7.27	0.912
ResTCN-DAM		**4.67**	**6.71**	**0.922**
LSTM	8	5.72	7.85	0.898
Bi-LSTM		5.79	7.88	0.898
GRU		5.98	8.64	0.895
TCN		6.23	8.63	0.889
ResTCN-DAM		**5.03**	**7.18**	**0.917**
LSTM	12	6.80	9.08	0.867
Bi-LSTM		6.29	8.34	0.886
GRU		7.14	9.71	0.857
TCN		7.43	9.56	0.845
ResTCN-DAM		**5.59**	**7.67**	**0.905**
LSTM	24	7.96	11.51	0.826
Bi-LSTM		7.85	9.86	0.817
GRU		7.81	10.84	0.831
TCN		8.92	11.10	0.774
ResTCN-DAM		**6.18**	**8.35**	**0.888**

Table 4.2: Performance of the mainstream models across various lead times. Best results in each group are highlighted in bold.

neural network models generate relatively accurate forecasts with NSE values exceeding 0.9, indicating a close fit to the original observational data. However, as the lead time increases to 24 hours, a performance degradation is observed in all models. This decline is attributed to the requirement of inputting data spanning a timeframe thrice the length of the lead time to enhance future forecasts, a factor that intensively tests the temporal forecasting capabilities of the models, particularly their ability to capture long-term

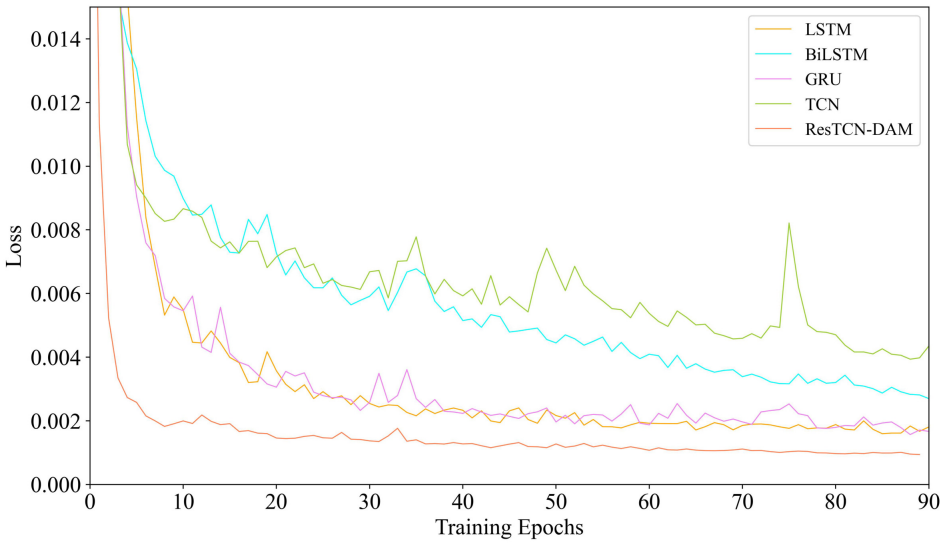
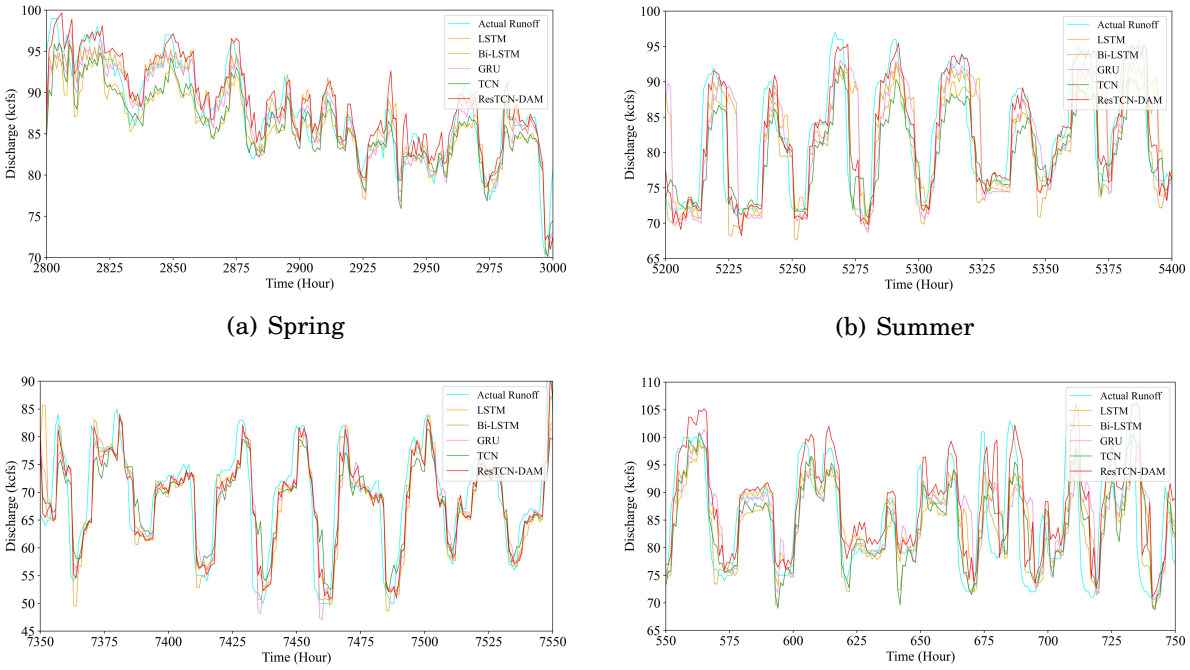


Figure 4.8: Training loss curves of different mainstream models at lead time of 24 in comparative experiments.

dependencies. Featuring a triple-layered dense shortcut connection architecture that radiates from inside out, the proposed ResTCN-DAM effectively captures and learns high-order abstract features in extensive time series by employing hierarchical learning within a deep neural network. Combined with the lightweight DAM module and snapshot ensemble method, it ensures enhanced accuracy while maintaining temporal robustness across varying lead times. Consequently, ResTCN-DAM demonstrates a noticeable improvement of up to 30% in all metrics across different forecast horizons compared to other models.

Additionally, the results indicate that LSTM and its variants, as well as TCN, have their respective strengths across various indices at shorter multiple lead times. This is because these mainstream time-series forecasting models can effectively learn the potential data patterns from the limited temporal features inputted and make relatively accurate forecasts. However, at longer lead times, the performance of Bi-LSTM surpasses other models, second only to ResTCN-DAM. This superiority is attributed to its bi-directional structure comprising two LSTM layers, processing temporal information



(a) Spring

(b) Summer

(c) Autumn

(d) Winter

Figure 4.9: Comparison of observed runoff curve for the Columbia River with forecast curves in different seasons.

in both forward and backward directions, thereby considering both past and future information for enhanced time series forecasting capability. In comparison, simpler structured LSTM, GRU, and TCN show relatively weaker performance in handling long time sequences. Nevertheless, the bi-directional architecture of Bi-LSTM introduces additional parameters, increasing computational costs and hindering the stacking of multiple layers for performance enhancement. Figure 4.8 shows the loss curves of these models during iterative training, distinctly highlighting ResTCN-DAM's lowest error rates and fastest training speed, vividly demonstrating our proposed model's state-of-the-art comprehensive performance.

4.3.3 The Performance of the Porposed ResTCN-DAM in Different Seasons

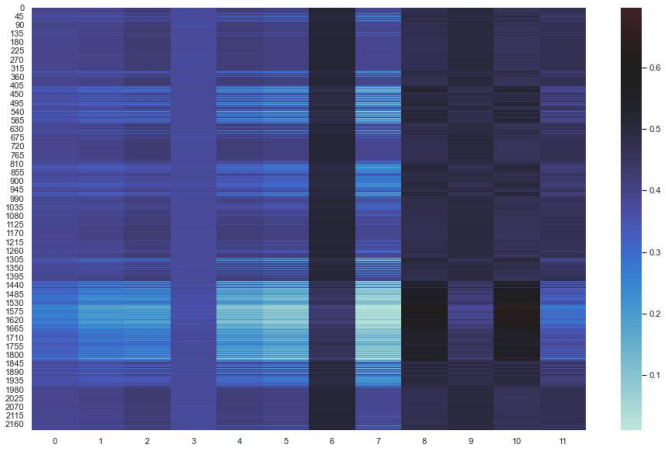
The discharge of the Columbia River is influenced by external factors such as climate change, precipitation, geological variations, and human socio-economic activities, which exhibit strong seasonality and trends. Among these factors, atmospheric precipitation has the most significant impact on discharge. Annually, during the winter season, atmospheric precipitation in the Columbia River basin falls in the form of snow in mountainous areas. In spring, the melted snowwater becomes the most crucial source of water supply, contributing significantly to the river's runoff. Consequently, the annual distribution of the Columbia River's discharge displays evident nonstationarity, with the runoff non-uniformity coefficient of 0.35. The period from April to July represents the flood season, during which these four months account for 68% of the total annual water volume, while water levels remain relatively stable during the autumn and winter seasons. To compare the ability of various neural network-based runoff forecasting models and the proposed ResTCN-DAM to forecast the actual runoff curve during different seasons, we conducted a visualization of the observed river runoff curve and model forecast curves. As shown in Figure 4.9, it is clear that ResTCN-DAM consistently outperforms other mainstream models in forecasting the complex variations of the observed runoff curve during both the high-flow and non-stationary spring season, as well as in the stable water seasons. This demonstrates that the proposed model has captured the seasonal characteristics of the Columbia River in each season, enabling it to make accurate and reliable forecasts throughout the year.

4.3.4 Interpretability of Attention Module

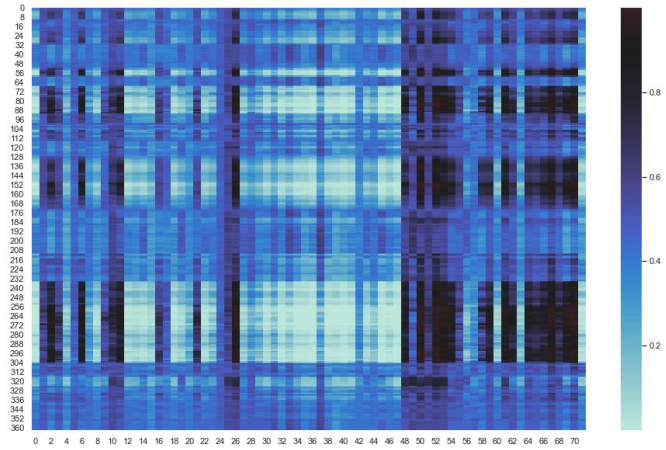
Interpretability is an emerging research focus in the current field of machine learning, often used to assess the extent to which forecasts made by machine learning models can

be understood and trusted by humans. The highly non-linear structure within neural networks brings about superior performance compared to traditional linear models. However, the intricate relationships between network parameters and the complexity of the optimization process make it challenging to mathematically derive the learning process from input to output, hence rendering neural networks as typical black-box models. In the field of runoff forecasting, many stakeholders in high-stakes applications involving economics and policy are reluctant to rely on forecasts from non-interpretable black-box models as trustworthy decision-making tools. This significantly impedes the widespread adoption and practical applications of cutting edge runoff forecasting models.

To enhance the interpretability of ResTCN-DAM, we employ an approach from IML involving model-internal weight visualization to provide model-specific insights into the forecasting process. Previous ablation and comparative experiment results demonstrate that our proposed ResTCN-DAM exhibits significantly higher accuracy and temporal robustness compared to other mainstream runoff forecasting models. The time step attention inside the DAM module plays a crucial role in achieving this, as it can adaptively model interdependencies among time series inputs and recalibrate information along the temporal dimension through weight factors. This enables the model to selectively amplify information from important time steps within the input sequence while suppressing relatively less useful information. For the aforementioned reasons, we choose to visualize the weights learned by the time step attention within the DAM module in the form of heatmaps. Our research focuses on interpreting features along the temporal dimension, primarily because explaining and visualizing the importance of temporal information is more intuitive and easier for humans to comprehend compared with the spatial dimension. Figure 4.10 illustrates the weight heatmaps of time step attention at lead times of 4 and 24, effectively confirming the interpretability of the DAM module across short and long forecast horizons. The DAM module employs the sigmoid



(a) Weight heatmap at the lead time of 4.



(b) Weight heatmap at the lead time of 24.

Figure 4.10: Visualization heatmap of time step attention weights in the proposed DAM module across various forecasting horizons. Deeper colors indicate that the DAM module places greater emphasis on the information from the input sequence at the corresponding time step.

function to transform the importance of learned time steps into weight values within the range $[0,1]$, with higher weight values indicating greater time step importance, resulting in deeper colors in the heatmaps. From both heatmaps, we observe that time step attention tends to assign higher weights to data points closer to the forecast time

point for both short and long forecast horizons (in our experiments, the input sequence length is three times the forecast horizon). This observation aligns well with empirical knowledge in existing research on time series forecasting. In runoff forecasting, the value at the current time point often exhibits stronger temporal correlations with data from the nearest few time points due to short-term influences, such as temperature, precipitation, and sudden natural disasters, which have a significant impact on runoff sequences over short periods. These short-term variations are reflected in changes in data points at adjacent time steps. Furthermore, due to lag effects and long-term trends, data points that are distant from the current time step may still contain valuable information that influences forecasts. Thus, the time step attention within the DAM assigns higher weight values to these corresponding time steps, suggesting that the DAM effectively captures long-term dependencies that exist in the input sequence. Through adaptive recalibration, it amplifies features at these critical time steps that significantly impact the current forecasts. The phenomena observed in the heatmaps strongly demonstrate that our proposed DAM module is able to effectively capture complex hidden relationships in the temporal dimension. The weights reflect time series characteristics that align closely with actual law and human domain expertise, demonstrating a high degree of interpretability.

4.4 Conclusion

This chapter proposes a novel multi-lead-time interpretable runoff forecasting framework ResTCN-DAM, based on TCN, ResNet, and attention mechanism module DAM. The overall architecture of this framework consists of densely connected residual blocks that form a three-tiered shorcut connection network radiating outward from the core. This promotes lossless information transfer from shallow to deep layers, effectively addressing the degradation problem in deep neural networks. The backbone model TCN inside each residual block inherits a streamlined CNN structure while possessing

causal and dilated convolutions suitable for time series forecasting, enabling linear stacking through ResNet to capture high-level abstract features in time series inputs. Each TCN is followed by a newly designed, plug-and-play lightweight attention module, DAM, which explicitly models interdependencies in both the temporal and feature dimensions at a lower computational cost. This encourages the model to focus more on critical information across different dimensions. The efficient collaboration among these modules comprehensively enhances the time series forecasting performance of the proposed ResTCN-DAM from multiple perspectives. Ablation and comparative experiments conducted on the Columbia River dataset validate the effectiveness of each module and superiority of overall framework. Compared to mainstream neural network-based runoff forecasting models, ResTCN-DAM demonstrates significant performance advantages and temporal robustness across various lead times. Furthermore, we employ internal weight visualization using IML methods to visualize the attention weights of the DAM module, enhancing the interpretability of the proposed ResTCN-DAM.

During the process of our research, we also recognize certain limitations in the proposed methodology, such as the relatively inefficient training of deep models and the limited interpretability of modules beyond attention mechanism. Therefore, in future research, we will strive to enhance the accuracy and efficiency of deep time series forecasting models and will employ more powerful IML methods to further enhance the global interpretability of model. This will allow us to develop high-performance runoff forecasting models that will be not only more accurate but also more transparent and trustworthy in their forecasting processes.

CHAPTER 5

RESIDUAL BIDIRECTIONAL GATED RECURRENT UNIT WITH SPATIOTEMPORAL SHORTCUTS

RQ2 highlights the performance bottlenecks in current neural network-based runoff forecasting models. Although the model proposed in Chapter 4 achieves high-accuracy predictions within 0 to 12 hours (with NSE > 0.9), most mainstream runoff forecasting models are still built upon recurrent architectures. Therefore, targeted improvements to this widely adopted class of models are urgently needed. To address this problem, we innovatively propose another interpretable hourly deep runoff forecasting framework: **S**patio-**T**emporal **Res**idual **Bi**directional **G**ated **R**ecurrent **U**nit (**STResBiGRU**). This framework fundamentally relies on a significantly enhanced BiGRU network, which not only captures the rich hidden relationships in long sequences through bidirectional traversal of the input sequence but also innovatively introduces unique temporal shortcut connections between cells. In the overall network architecture, the novel design of dual residual block pathways and dense spatial shortcut connections considerably deepens the recurrent architecture-based neural network. The integration of spatiotemporal shortcut connections effectively improves the mode of information transmission in different

dimensions, mitigates the vanishing gradient problem caused by deep networks and long sequences, and enhances feature reuse. In addition, we incorporate the lightweight spatiotemporal attention module DAM into the bidirectional information fusion process of BiGRU to explicitly model the interdependencies among features in the forward and backward GRU output maps across both temporal and spatial dimensions. This mechanism selectively amplifies salient features to enhance the model's temporal robustness when handling long sequences. Furthermore, to improve interpretability, heatmap visualizations based on IML techniques are applied to both the forward and backward branches of the BiGRU, providing insight into the model's prediction process. Ablation and comparative experiments on a real-world dataset from the Columbia River in the United States demonstrate that the proposed STResBiGRU framework significantly enhances accuracy and temporal robustness over mainstream models, providing reliable decision support for stakeholders.

Section 5.1 outlines the chapter's motivation and key contributions. Section 5.2 details the proposed methodology. Section 5.3 offers a comprehensive model evaluation, and Section 5.4 presents the conclusions.

5.1 Introduction

In hydrology, runoff data inherently form a multivariate time series, where future runoff values over a specific forecasting horizon are highly correlated with historical variables such as runoff, precipitation, temperature, and evaporation [139]. These influences are often transmitted with temporal lags across multiple time steps [140]. Compared to traditional neural networks, recurrent architecture-based neural networks such as RNN possess a unique capacity for modeling temporal dependencies. At each time step, the output depends not only on the current input but also on the hidden state from previous steps, endowing them with memory capability. Building upon RNN, LSTM is proposed

as a milestone neural network model. Its distinctive gating mechanism enables selective retention and forgetting of information from past inputs, significantly enhancing the ability to capture long-term dependencies and the delayed effects inherent in hydrological processes [141]. Furthermore, LSTM naturally supports multivariate inputs and multi-step forecasting, and can be flexibly integrated with other machine learning components such as CNN and attention mechanism to form powerful hybrid models. Owing to these advantages, LSTM-based architectures have been widely adopted in runoff forecasting applications. Deng et al. propose a hybrid neural network model that combines 1-D CNN and LSTM for daily rainfall-runoff forecasting [142]. The model initially applies CNN with convolution and pooling to derive high-dimensional features, followed by LSTM to uncover temporal dependencies within these local patterns. Experimental results indicate that CNN-LSTM significantly outperforms individual LSTM models in terms of accuracy and efficiency over longer lead times. Wu et al. further exploit the advantages of such hybrid models for quantitative precipitation estimation (QPE) in China [143]. The proposed hybrid model uses CNN to model and process two-dimensional gridded data obtained from satellites and rain gauges to extract spatial features, with LSTM capturing the temporal dependencies of precipitation. This allows CNN-LSTM to simultaneously model and learn the spatial and temporal correlations of the inputs. Compared to individual models like CNN and multi-layer perceptron (MLP) [144], the proposed CNN-LSTM hybrid model reduces errors by up to 17%.

The widespread application of recurrent architecture-based neural networks in runoff forecasting highlights their significant potential for further development [145]. However, these mainstream time series forecasting models exhibit notable limitations in real-world scenarios, particularly along spatial and temporal dimensions. From a spatial architecture perspective, the increasing demand for predictive accuracy among hydrological stakeholders necessitates continual improvements in model design to meet rising perfor-

mance expectations. As one of the most widely used models in runoff forecasting, LSTM suffers from substantial parametric complexity due to its gating units. When multiple layers are stacked to learn high-level abstract features, the resulting degradation issues make the network difficult to train, severely limiting its performance scalability. In the temporal dimension, existing models often struggle with vanishing gradient and long-term dependencies when processing long input sequences containing complex contextual information [146]. These shortcomings prevent models from fully capturing the intricate patterns and relationships within time series data. Consequently, recent research has focused on hybridizing LSTM with models such as CNN to enhance its feature extraction capability [147]. Moreover, the black-box nature of neural networks poses a significant challenge to interpretability. The inability of stakeholders to understand the forecasting process undermines trust in the model's predictions, thereby hindering its adoption in critical hydrological decision-making.

To address the above problems, this chapter introduces a novel short-term runoff forecasting framework: STResBiGRU. This framework, which is based on BiGRU, spatiotemporal shortcut connections, and dual-path attention mechanism, encapsulates the following innovative features:

- The proposed STResBiGRU framework utilizes our redesigned BiGRU capable of simultaneously processing bidirectional information flows, which not only innovatively incorporates dense shortcut connections built around dual residual block pathways in its spatial architecture, but also establishes unique temporal shortcut connections between GRU cells to facilitate efficient information transfer.

- We further integrate the lightweight dual-path spatiotemporal attention module DAM with the bidirectional architecture, enabling adaptive recalibration of temporal and feature weights during the fusion of forward and backward information

flows in BiGRU. By selectively emphasizing key information in each branch, the attention mechanism facilitates efficient bidirectional fusion, thereby enhancing the model's forecasting capability over long input sequences.

- The IML-based heatmap visualization method is extended from unidirectional to bidirectional architecture, providing local post-hoc explanations for the forecasting processes of each unidirectional branch in STResBiGRU, thereby enhancing the interpretability of the proposed model.

5.2 Methodology

We present the STResBiGRU framework for interpretable hourly runoff forecasting, as shown in Figure 5.1. In the spatial architecture of the overall network, we adopt dual residual block pathways instead of the traditional single pathway of ResNet. Within each residual block, the BiGRU models the temporal input both forward and backward in parallel, capturing hidden relationships within the bidirectional sequence from a more comprehensive perspective. Notably, we innovatively introduce temporal residual connections between BiGRU cells, allowing each cell to establish connections with earlier cells, thereby enhancing the reuse of temporal information and mitigating problems of vanishing gradient and long-term dependencies in the temporal dimension. To facilitate the effective integration of bidirectional information in the BiGRU and enhance the model's capacity to handle long sequences, a dual-path attention mechanism module is inserted after each BiGRU within the residual blocks. This module also explicitly models the importance of time steps and features within the bidirectional information flow of the BiGRU across both temporal and spatial dimensions, selectively amplifying key information in the feature maps while suppressing irrelevant details. The deep integration of spatiotemporal shortcut connections, BiGRU, and dual-path attention

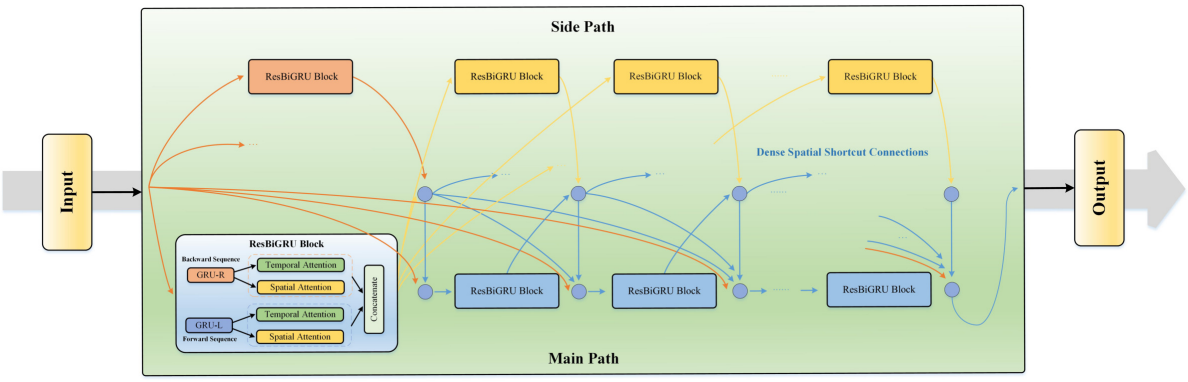


Figure 5.1: The overall architecture of the proposed STResBiGRU hourly runoff forecasting framework.

mechanism significantly enhances the depth and capacity of the time series forecasting model to handle long sequences, enabling us to achieve accurate and robust runoff forecasting.

5.2.1 Gated Recurrent Unit with Temporal Shortcuts

Traditional neural networks cannot perform time series forecasting, but the proposal of RNN allows researchers to learn and use the hidden relationship of input in the time dimension. However, RNN has a serious long-term dependencies problem, which makes the network lose the ability to learn information in a large time interval. LSTM is a milestone in the development of RNN, it can selectively memorize data by adding gating units on the basis of RNN. Because LSTM has powerful performance, it has basically replaced RNN as one of the most widely used neural network structures. Figure 5.2.1 demonstyrates the typical LSTM structure. The core of LSTM is the cell state which determines the retained information and the forgotten information. The formulas of the LSTM structure are as follows:

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \tag{5.1}$$

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \tag{5.2}$$

$$g_t = \phi\left(W_{gx}x_t + W_{gh}h_{t-1} + b_g\right) \tag{5.3}$$

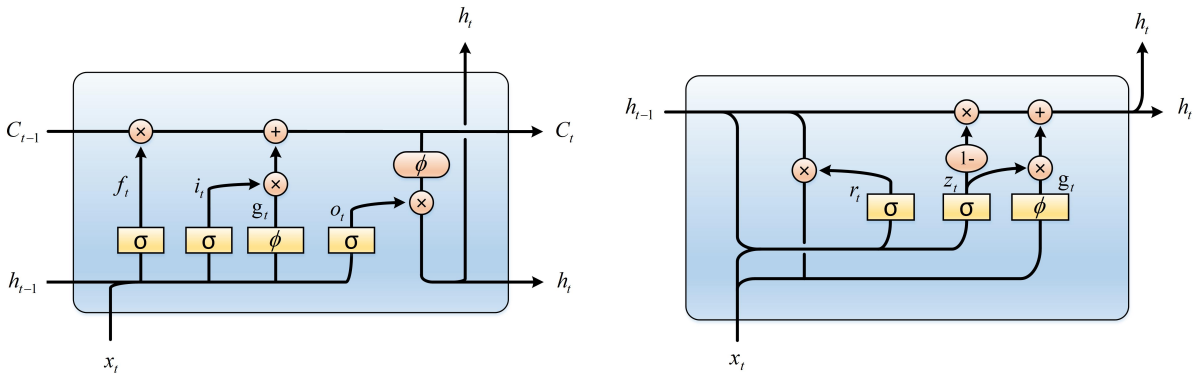
$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \tag{5.4}$$

$$C_t = g_t \odot i_t + C_{t-1} \odot f_t \tag{5.5}$$

$$h_t = \phi\left(C_t\right) \odot o_t \tag{5.6}$$

where W is the weight matrix for the corresponding input; b is the bias term; σ and ϕ represent sigmoid and tanh nonlinear activation function; C_t represents the cell state, which is the core parameter and can determine the forgotten and stored information. C_t needs to be maintained and updated through three gating units: forget gate f_t, input gate i_t and output gate o_t. The forget gate f_t is used to determine the information that needs to be removed from C_t. The input gate i_t chooses the stored information. The output gate o_t controls how much information the cell state C_t has to output to the current hidden state h_t. g_t is the newly generated cell state; h_t represents the intermediate information transmitted between cells; x_t is the time series input at tth step; $\odot$ is the element-wise multiplication. Through the above three gating units, LSTM can selectively forget and store information, thus having better performance than RNN.

Although LSTM has excellent performance, it comes at the cost of increasing a huge amount of parameters. Since the gating units contain a large number of weights and bias term parameters, the number of parameters of LSTM is greatly increased. The parameter amount of a typical LSTM is 4 times that of a naive RNN, which greatly increases the training time and computational cost, and easily leads to overfitting. To solve this problem, many variants of LSTM have also been proposed, the most famous of which is GRU [148, 149]. The structure of GRU is shown in Figure 5.2.1. Like LSTM, GRU can also selectively memorize data, and it has a more streamlined architecture, which



(a) The typical LSTM structure. (b) The GRU structure.

Figure 5.2: The comparison of typical LSTM and GRU. GRU merges the gating units of LSTM, which significantly reduces the parameters, and improves the model efficiency while having the ability to selectively memorize and forget information.

reduces the amount of calculation while achieving the same level of performance as LSTM. The proposed STResBiGRU is designed to develop a deep recurrent architecture-based time series forecasting model capable of learning higher-order abstract features from input sequences. Consequently, each layer in the network needs to maintain lightweight characteristics to reduce the number of parameters. To this end, each residual block is built upon the lightweight GRU architecture, which offers a favorable trade-off between complexity and performance. The GRU consists of a series of sequentially connected cells along the temporal dimension, with each cell formulated as follows:

$$r_t = \sigma\left(W_{rx}x_t + W_{rh}h_{t-1} + b_r\right) \tag{5.7}$$

$$z_t = \sigma\left(W_{zx}x_t + W_{zh}h_{t-1} + b_z\right) \tag{5.8}$$

$$g_t = \phi\left[W_{gx}x_t + W_{grh}(r_t \odot h_{t-1}) + b_g\right] \tag{5.9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot g_t \tag{5.10}$$

where x_t represents the input at the tth time step to the GRU Cell; W and b respectively denote the weight matrix and bias terms associated with the gating units; σ and ϕ

are the sigmoid and tanh activation functions, respectively. A significant improvement of the GRU over the LSTM is the consolidation of the input and forget gates into a single update gate z_t, which selectively memorizes the current input information and determines the extent to which past hidden state information is retained. The reset gate r_t controls the amount of information to be forgotten from the previous hidden state h_{t-1} when generating the new hidden state h_t in conjunction with the current input x_t, which is crucial for capturing short-term dependencies within the input sequence. Furthermore, unlike the LSTM, the GRU does not have a separate cell state but uses the hidden state h_t to control information flow, thereby reducing the number of parameters. Consequently, GRU maintains a performance level comparable to that of LSTM but with a more lightweight structure. This results in lower computational costs and enhanced computational efficiency, thereby laying the foundation for constructing bidirectional, deep time series forecasting model [150].

Although GRU optimizes the model architecture based on LSTM, the vanilla GRU still encounters some of the same limitations present in the LSTM. Specifically, while the gating units selectively retain crucial information, each GRU Cell is significantly influenced by the hidden state of the previous time step, leading to a gradual forgetting of earlier antecedent information as the sequence length increases, exacerbating information loss. Moreover, recurrent architecture-based neural network models update weights incrementally by backpropagation through time (BPTT) [151]. Although gating units can mitigate the vanishing gradient to some extent, when dealing with very long time sequences, the accumulation of gradients over multiple time steps can still result in exceedingly small effective gradients, causing a substantial decline in training efficiency. The proposed STResBiGRU aims to enhance the temporal robustness of time series forecasting models across multiple forecast horizons of varying lengths. To address the decline in performance of the core GRU architecture over extended lead times, we

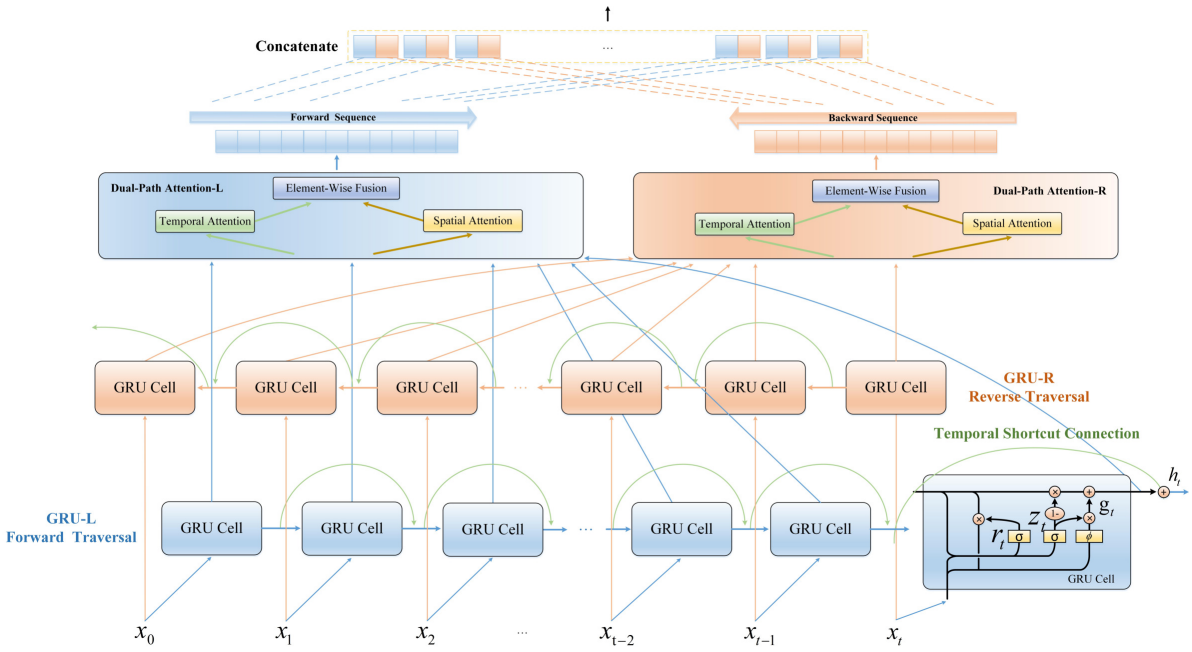


Figure 5.3: The structure of the core module BiGRU. Green lines indicate the unique temporal shortcut connections between cells.

draw inspiration from ResNet and innovatively introduce temporal shortcut connections between GRU cells. As illustrated in Figure 5.3, each GRU Cell not only receives the hidden state h_{t-1} from the previous time step as input but also establishes shortcut connections that span cells, directly linking input to output. Consequently, the output of each GRU Cell at any time step can be expressed as follows:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot g_t + h_{t-1} \tag{5.11}$$

The incorporation of unique temporal residual connections effectively suppresses gradient vanishing along the temporal dimension and ensures that information from previous time steps can be transmitted intact across intermediate cells to subsequent time steps. This feature significantly enhances the training efficiency and performance of the model when handling longer time sequences, thereby enabling the proposed STResBiGRU to achieve robust forecasts across various forecasting horizons.

5.2.2 Bidirectional Architecture Integrated with Dual-Path Spatiotemporal Attention Mechanism

Traditional time series forecasting models also have a significant limitation in that they only process sequences unidirectionally from past to present. To forecast hourly runoff up to 24 hours in the future, the input sequences often contain longer previous information, which can include complex dependencies that are difficult for traditional unidirectional models to capture fully. Research by Siami-Namini et al. demonstrates that BiLSTM significantly outperforms traditional unidirectional LSTM in time series forecasting tasks by additionally training on the raw data in reverse order [152]. Inspired by this, we further modify the GRU layers within each residual block of the STResBiGRU by adding an additional GRU to traverse the input sequence both forward and backward, forming the BiGRU as shown in Figure 5.3. The proposed BiGRU comprises two structurally identical GRU branches, where GRU-L for forward information flow traverses the input sequence from x_0 to x_t, and the newly added GRU-R for backward information flow traverses the sequence from x_t to x_0 in reverse order. The two sets of unidirectional modeling processes of BiGRU are performed in parallel. It is noteworthy that in each unidirectional GRU, the proposed temporal residual connections are also incorporated to mitigate vanishing gradient and enhance feature reuse.

However, when modeling long sequences, the complex multivariate features contained in bidirectional sequences make it challenging for conventional models to accurately identify key elements. To address this, as demonstrated in Figure 5.4, a lightweight spatiotemporal attention module DAM, previously introduced in Chapter 4, is further embedded after each unidirectional GRU. This module is embedded after each unidirectional GRU rather than after the bidirectional sequence fusion, primarily because the fused sequence already contains bidirectional information, which may obscure the unique and important components within each branch. The proposed module explicitly models

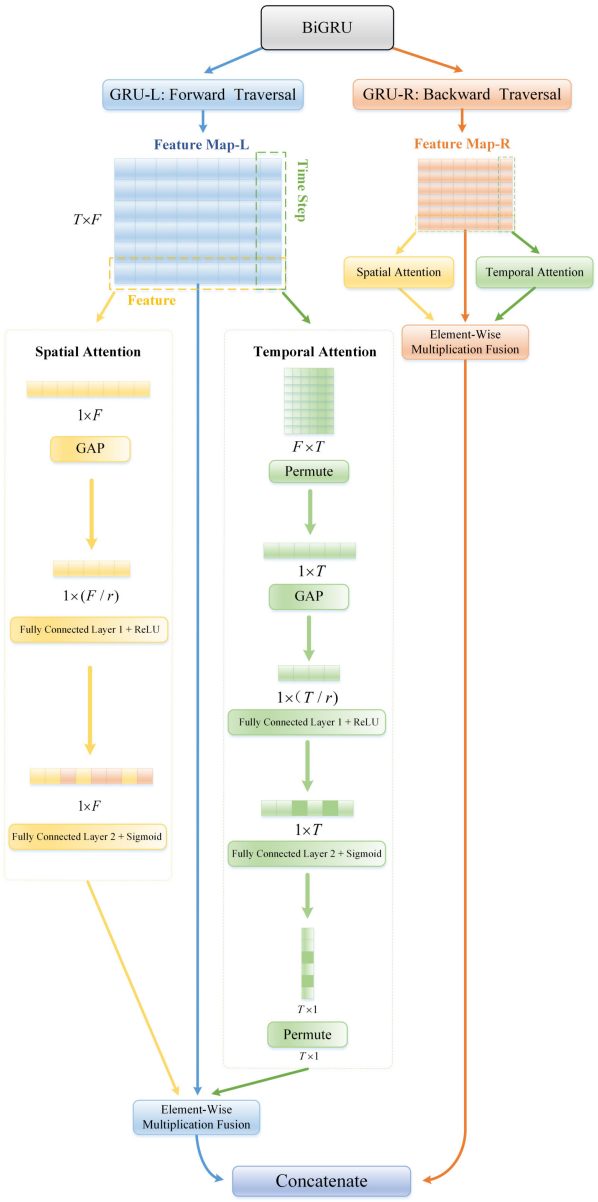


Figure 5.4: The plug-and-play spatiotemporal attention module DAM facilitates the effective fusion of bidirectional sequences in BiGRU.

feature importance across both temporal and spatial dimensions in the 2D feature maps. By dynamically recalibrating feature weights, it selectively emphasizes critical features in both forward and backward sequences, thereby facilitating effective bidirectional information fusion. Subsequently, the adaptively recalibrated bidirectional forecasts are then merged step-by-step to produce the final output of the BiGRU.

It is noteworthy that the bidirectional structure employed by BiGRU only operates on the all known observations $\{x_0, x_1, ..., x_t\}$ available at timestep t, thereby ensuring that this structure does not lead to leakage of future data in time series forecasting. The adoption of the bidirectional structure in BiGRU allows each timestep element to perceive both forward and backward information, providing a richer and more comprehensive feature representation. This bidirectional information flow enables BiGRU to more effectively learn and capture complex patterns and long-term dependencies within time series compared to traditional unidirectional models. Furthermore, thanks to the lightweight characteristic of GRU, the increase in model complexity and computational cost with BiGRU remains within acceptable bounds compared to other bidirectional models, such as BiLSTM. This feasibility allows BiGRU, as a core component, to be integrated with the ResNet Plus architecture, thereby facilitating the construction of the deep time series focecasting model STResBiGRU.

5.2.3 Model Design and Implementation Details

The proposed STResBiGRU is designed for 24-hour ahead runoff forecasting. In the proposed attention module DAM, the hyperparameter for dimensionality reduction r is set to 4. Except for certain modules with specific functions, the default nonlinear activation function in STResBiGRU is ReLU. The sliding window strategy is employed for rolling forecasting, with the input sequence length set to three times the forecasting horizon. The model is trained using the MSE as the loss function, and its performance is evaluated using MAE, MAPE, and NSE metrics. During training, the model undergoes 100 iterations with Adam optimizer, and the snapshot ensemble method is applied to enhance robustness. All other hyperparameter settings and implementation details remain consistent with those described in Chapter 4.

5.3 Experiment Results and Discussion

5.3.1 Ablation Studies on Key Modules of STResBiGRU

The proposed runoff forecasting framework STResBiGRU addresses the degradation problem in deep networks by incorporating additional residual block pathway and dense spatial shortcut connections. The BiGRU with temporal shortcut connections, alongside the plug-and-play attention module DAM, significantly enhances the model's forecasting performance over longer lead times. We specifically design a series of ablation studies to validate the effectiveness of these core modules, with the lead time uniformly set to the designed maximum forecasting horizon of 24 hours, and the core BiGRU layers incrementally increasing from one to four layers (excluding other hidden layers such as those from the attention module). In the ablation studies, we design three model structures containing different modules: BiGRU as the baseline model, BiGRU-DAM incorporating the lightweight attention module DAM, and STResBiGRU, which includes both spatiotemporal shortcut connections and the DAM module.

The results of the ablation studies are shown in Table 5.1, from which it can be clearly seen that the performance of both BiGRU and BiGRU-DPA without residual structures initially improves then declines. Traditional architecture-based BiGRU networks reach a performance threshold after stacking two layers, and then when the network depth continues to increase, the vanishing gradient and degradation problem seriously restrict the performance of the model, making the performance of the deep network even inferior to that of the shallow network. Taking BiGRU-DPA as an example, the error increases by 5.4% when the depth is increased from 2 to 4 layers, proving that the existence of the degradation problem prevents the additional depth increase from effectively obtaining positive gains. Notably, at each depth, the performance of BiGRU-DPA, which incorporates the lightweight spatiotemporal attention module, consistently outperforms

Model	Depth	MAE	MAPE	NSE
BiGRU		7.89	10.34	0.812
BiGRU-DAM	1	7.66	10.08	0.840
STResBiGRU		**7.47**	**9.75**	**0.847**
BiGRU		7.60	10.76	0.845
BiGRU-DAM	2	7.35	10.13	**0.852**
STResBiGRU		**7.33**	**9.83**	0.851
BiGRU		7.62	10.27	0.845
BiGRU-DAM	3	7.52	10.01	0.844
STResBiGRU		**7.16**	**9.61**	**0.858**
BiGRU		7.89	10.89	0.834
BiGRU-DAM	4	7.75	10.24	0.837
STResBiGRU		**6.92**	**9.31**	**0.866**

Table 5.1: Performance comparison of the proposed STResBiGRU and models integrating various modules at different depths. Best results in each group are highlighted in bold.

the baseline BiGRU, demonstrating the effectiveness of DPA in adaptively recalibrating multidimensional features. Unlike the other two models, STResBiGRU, which deeply integrates the ResNet Plus architecture, consistently achieves the best performance at any depth due to its unique temporal shortcut connections between GRU cells and dense spatial shortcut connections, and it can be stacked up to 4 layers without degradation, thereby far surpassing traditional models in depth. When STResBiGRU is increased from 1 to 4 layers, its performance improves by 7.4%, validating that the ResNet Plus architecture mitigates degradation in deep networks, allowing the additional layers to learn higher-order abstract features from the sequence and thus achieve richer feature representations. Furthermore, Figure 5.5 demonstrates the loss curves of different model during the training process, showing that STResBiGRU, with its additional residual block pathway and dense shortcut connections, optimizes the backpropagation path of gradients. As a result, our proposed model exhibits faster convergence and lower loss.

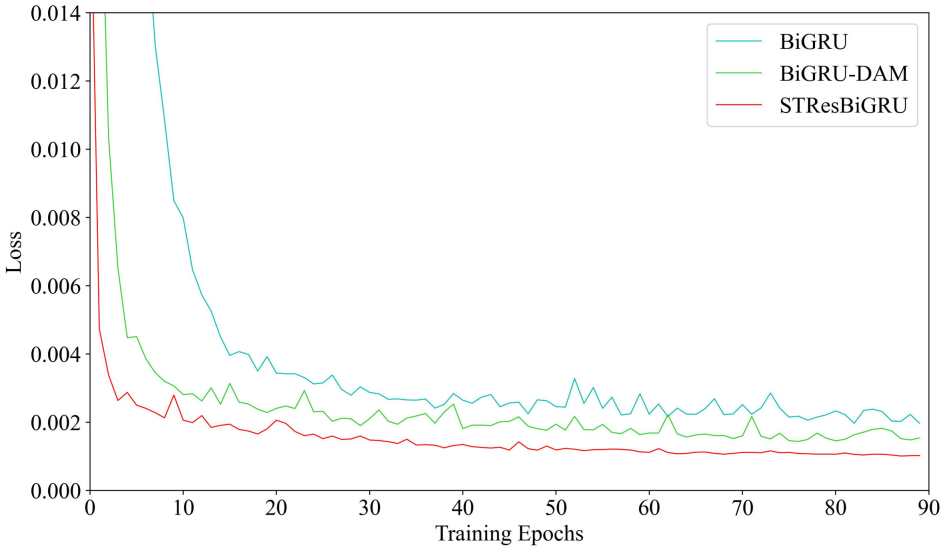


Figure 5.5: Visualization of training loss curves for models integrating various modules at depth of 4.

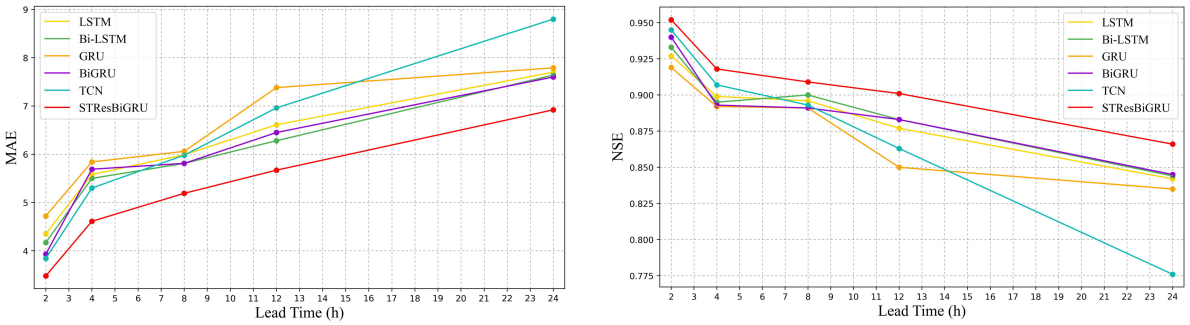
5.3.2 Comparative Experiments with Mainstream Models

The main problem with existing runoff forecasting models is that they perform well at short lead times, but their performance deteriorates severely as the forecast horizon becomes longer. One of the research objectives of the proposed STResBiGRU is to address the lack of temporal robustness in models. Therefore, we test the performance of mainstream runoff forecasting models over five different lead times from $t+2$ to $t+24$, as shown in Table 5.2, and visualized in Figure 5.6. It is evident that at shorter lead times, the NSE values for the models are above 0.9, indicating that these neural network-based models can make accurate forecasts. However, as the lead time continues to increase, the performance of all the models decreases to varying degrees, mainly due to the fact that the forecast for longer lead times relies on longer previous information, and these sequences as inputs to the model may often be several times the forecast horizon. Therefore, the capability to adequately model and learn the rich contextual information and complex relational patterns in long sequences becomes the key to determine the performance of the model. As shown in Table 5.2, the proposed STResBiGRU

Model	Lead Time	MAE	MAPE	NSE
LSTM	2	4.35	5.88	0.927
BiLSTM		4.17	5.42	0.933
GRU		4.72	6.39	0.919
BiGRU		3.93	5.23	0.940
TCN		3.84	5.39	0.945
STResBiGRU		**3.48**	**4.92**	**0.952**
LSTM	4	5.58	7.71	0.899
BiLSTM		5.50	7.53	0.895
GRU		5.84	8.17	0.892
BiGRU		5.69	7.66	0.893
TCN		5.30	7.32	0.907
STResBiGRU		**4.61**	**6.62**	**0.918**
LSTM	8	5.99	8.26	0.896
BiLSTM		5.81	7.70	0.900
GRU		6.06	8.78	0.891
BiGRU		5.81	7.71	0.891
TCN		5.98	8.10	0.893
STResBiGRU		**5.19**	**7.32**	**0.909**
LSTM	12	6.61	8.82	0.877
BiLSTM		6.28	8.44	0.883
GRU		7.38	10.09	0.850
BiGRU		6.45	9.20	0.883
TCN		6.96	9.41	0.863
STResBiGRU		**5.67**	**7.89**	**0.901**
LSTM	24	7.70	10.24	0.842
BiLSTM		7.64	10.63	0.844
GRU		7.79	10.71	0.835
BiGRU		7.60	10.76	0.845
TCN		8.80	11.24	0.776
STResBiGRU		**6.92**	**9.31**	**0.866**

Table 5.2: Performance comparison of the proposed STResBiGRU with mainstream models at different lead times. Best results in each group are highlighted in bold.

consistently outperforms other models at all lead times, achieving SOTA performance and maintaining excellent temporal robustness. Benefiting from its novel bidirectional architecture, STResBiGRU effectively utilizes bidirectional information flows to learn



(a) MAE (b) NSE

Figure 5.6: MAE and NSE of models at various lead times.

more feature representations, while the attention module DAM enhances the model's capacity to perceive the importance of multidimensional features. In addition, the unique spatiotemporal shortcut connections effectively mitigate the vanishing gradient problem in corresponding dimensions, which not only improves the model's capability to model long sequences, but also effectively enhances its depth. The integration of these modules provides STResBiGRU with a distinct performance advantage over other mainstream neural network models, with approximately a 20% improvement in accuracy at each lead time, thoroughly validating the advanced nature of the proposed framework.

5.3.3 Interpretability of Attention Module

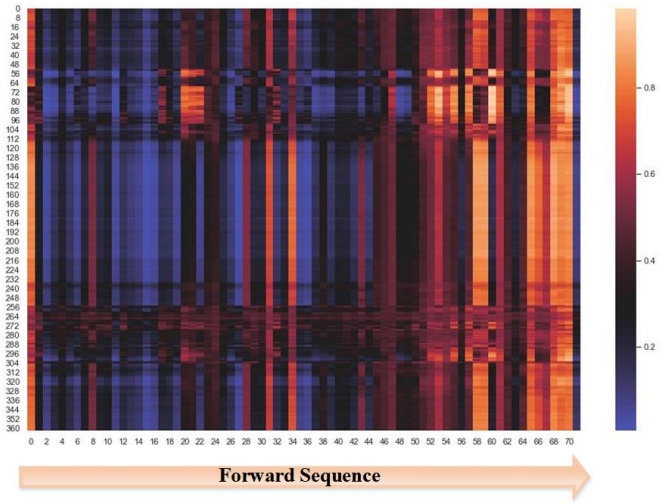
In order to enhance the interpretability of the proposed STResBiGRU, we visualize the feature weights learned by the attention module DAM embedded behind the forward and reverse branches of BiGRU in the temporal dimension using heat map. It can be seen from Figure 5.7 that for both the forward and backward outputs of BiGRU, DAM tends to assign higher weights to features on time steps closer to the forecasted time point (nearer to the right in the forward sequence and nearer to the left in the backward sequence), and this observation is consistent with empirical regularities gained by time series forecasting in practice. In most practical application scenarios, neighboring time points in a time series tend to have strong correlation directly, but at the same time, due

to the existence of long-term dependencies, i.e., certain time points may have specific connections with others far apart, and these real-world objective laws are well reflected by the heatmap, which proves that the modeling process of the features in our proposed DAM module conforms to the objective laws and can be understood by humans. Through this model-specific post hoc explanation, the forecasting process of STResBiGRU becomes more transparent, significantly enhancing the model's interpretability.

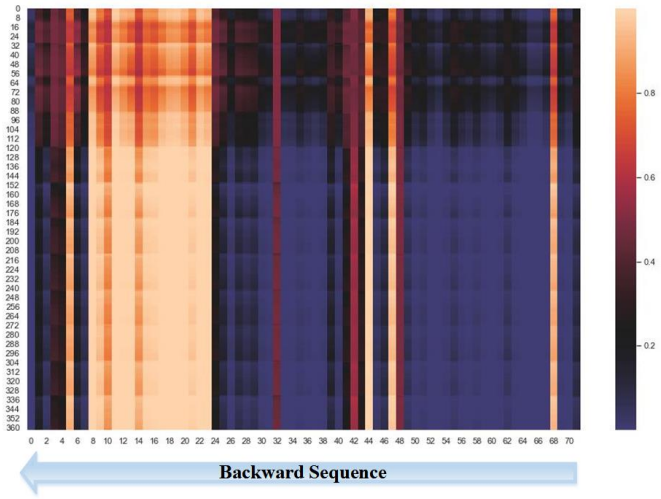
5.4 Conclusion

This chapter demonstrates STResBiGRU, an hourly runoff forecasting framework with high temporal robustness. Its overall architecture is based on BiGRU, which utilizes a novel bidirectional recurrent architecture to traverse input sequences in both directions, capturing more comprehensive feature representations. Innovatively, unique temporal residual connections between GRU Cells are introduced to mitigate the problem of vanishing gradient over long sequences in the temporal dimension. To enhance the fusion of bidirectional sequences in BiGRU, the lightweight spatiotemporal attention module DPA is further integrated. Owing to its plug-and-play nature, this module can be seamlessly embedded after each unidirectional GRU branch to adaptively recalibrate the features of the forward and backward output sequences along both temporal and spatial dimensions. The redesigned network architecture features dual residual block pathways and dense shortcut connections, optimizing the backpropagation path of gradients and significantly deepening the model to learn higher-order abstract features. Additionally, IML-based heatmap visualization is extended to each branch of the BiGRU to enhance the interpretability of the forecasting process. Ablation studies and comparative experiments prove the effectiveness and advancement of the core components within the proposed STResBiGRU, enabling robust and accurate runoff forecasting.

However, the research also reveals that although the proposed STResBiGRU achieves



(a) Weight heatmap of the forward GRU.



(b) Weight heatmap of the backward GRU.

Figure 5.7: Heatmap of weights obtained by modeling the forward and backward outputs of STResBiGRU in the time dimension through the DPA module at lead time of 24.

greater depth and higher accuracy than the vanilla GRU and other recurrent architecture-based models, its overall performance at certain forecasting horizons remains slightly inferior to that of the TCN-based ResTCN-DAM, indicating potential for further improvement. Moreover, the first three models proposed in this thesis exhibit NSE values below 0.9 when the forecasting horizon reaches 24 hours, indicating that they have not yet

fully achieved robust multi-lead-time forecasting. Addressing this limitation will be a key focus of our future work.

CHAPTER 6

Deep Bidirectional Mamba for Robust Multi-Lead-Time Runoff Forecasting

RQ3 highlights the urgent demand among hydrological stakeholders for outstanding models capable of multi-lead-time runoff forecasting with strong temporal robustness. As runoff forecasting continues to evolve from single-step to multi-step prediction, the expectations placed on models have shifted from achieving high performance over selected intervals to maintaining temporal robustness across extended forecasting horizons of up to 24 hours. To address this challenge and achieve RO3, we propose a robust and interpretable fine-grained runoff forecasting framework: ResBi-Mamba Plus. The model is built upon a redesigned bidirectional Mamba (Bi-Mamba) backbone. As an emerging architecture grounded in state space theory, Bi-Mamba inherits both convolutional characteristics from CNN and recurrent properties from RNN, while maintaining linear complexity, enabling efficient training and inference on GPU accelerators. The incorporation of bidirectional architecture allows the model to simultaneously traverse input sequences in both forward and backward directions, thereby capturing richer contextual dependencies. To further enhance the modeling of multivariate long-sequence inputs,

we integrate the dual-path spatiotemporal attention module after each unidirectional Bi-Mamba branch. This lightweight module adaptively recalibrates original features in temporal and spatial dimensions based on channel-dependence (CD) and channel-independence (CI) strategies, thereby allowing each unidirectional branch to highlight its critical information during the fusion process. Additionally, the ResNet Plus architecture is retained to optimize information flow in deep networks, enabling the model to achieve unprecedented performance gains through depth enhancement that surpass the vanilla Mamba architecture. Extensive ablation and comparative experiments conducted on the hourly runoff dataset of the Columbia River Basin in the United States validate the effectiveness of each module. Notably, the proposed ResBi-Mamba Plus establishes new SOTA performance for extended forecasting horizons, significantly outperforming current leading Transformer-based models in hydrology.

Section 6.1 describes the research background and key challenges addressed in this chapter. Section 6.2 presents the technical details of the proposed ResBi-Mamba Plus in detail. Section 6.2 presents the technical details of the proposed ResBi-Mamba Plus framework. Section 6.3 reports the model's performance metrics across various forecasting horizons. Section 6.4 concludes the contributions of this chapter.

6.1 Introduction

As a key branch of deep learning, time series forecasting has received considerable attention from the research community, leading to the emergence of numerous representative models [153]. Neural network-based models applied to runoff forecasting can be primarily classified into two main categories according to different stages of development and their corresponding characteristics.

The first category is the classical time series forecasting methods, represented by

RNN and TCN, which constitute the prevailing paradigm and are widely adopted. Recurrent architecture-based neural networks, such as LSTM and GRU, possess certain capabilities for selective information retention and forgetting. However, these models still suffer from vanishing gradient and limited ability to capture long-term dependencies in the temporal dimension [154], which significantly constrains their performance on long sequences. Moreover, the inherently sequential computation also seriously affects their computational efficiency. Although TCN supports parallel processing, its performance remains constrained to short or medium horizons due to limited receptive fields [155]. Through the work in Chapters 3 to 5, we construct a diversified runoff forecasting framework incorporating various neural network architectures based on these mainstream models. The proposed models address key limitations of traditional architectures and achieve high accuracy within certain forecasting horizons. However, when the horizon is extended to 24 hours, performance degradation becomes evident, suggesting that these models still fall short of delivering SOTA performance for multi-lead-time forecasting [156].

The second category involves Transformer-based models [157], including Informer [158] and iTransformer [159], which represent an emerging paradigm in runoff forecasting. These models utilize self-attention mechanisms to capture arbitrary dependencies across input sequences, enabling both parallel processing and effective feature extraction. In theory, Transformer can handle sequences of any length, effectively overcoming the long-term dependencies modeling limitations of classical methods. This makes it foundational to large-scale language models (LLM) [158] such as the famous Generative Pre-trained Transformer (GPT) [160]. However, despite demonstrating exceptional time series forecasting capability, Transformer suffers from quadratic complexity relative to input sequence length due to its multi-head attention. This squared complexity leads to prohibitive computational costs when processing long sequences, severely constraining

its effectiveness for extended forecasting horizons [161].

To achieve robust multi-lead-time runoff forecasting as outlined in RO4, it is imperative to overcome the inherent structural limitations of existing models. This necessity prompts a shift toward exploring new architectural paradigms. Recently, Gu et al. propose a novel time series forecasting architecture Mamba [162, 163], characterized by its dual attributes of convolution and recursion while still maintaining linear complexity. Moreover, it enhances long-sequence contextual understanding by incorporating innovative memory initialization and selection mechanisms. To further boost computational efficiency, Mamba adopts two hardware-aware algorithms, parallel associative scan and memory recomputation, tripling its computational efficiency. Due to its outstanding performance, Mamba has garnered significant attention and is widely regarded as a strong contender to Transformer-based architectures [164], with promising potential for broad application in time series forecasting. Inspired by this, we innovatively propose a powerful hourly runoff forecasting framework: **ResBi-Mamba Plus**, which deeply integrates redesigned bidirectional Mamba (**Bi-Mamba**), improved deep residual network (**Res**Net **Plus**), and lightweight spatiotemporal attention mechanism DAM, capable of robust and interpretable multi-lead-time runoff forecasting. The main contributions of this chapter are as follows:

- We propose a novel Bi-Mamba architecture based on the modified Mamba-2, which can process bidirectional information flows in parallel. Serving as the backbone of our overall framework, Bi-Mamba not only surpasses the performance of Mamba-1 in each unidirectional branch but also boosts forecasting capability for long sequences through richer bidirectional contextual representations.

- The proposed Bi-Mamba integrates the spatiotemporal attention module into its bidirectional fusion process, enhancing the parallel modeling of complex interdepen-

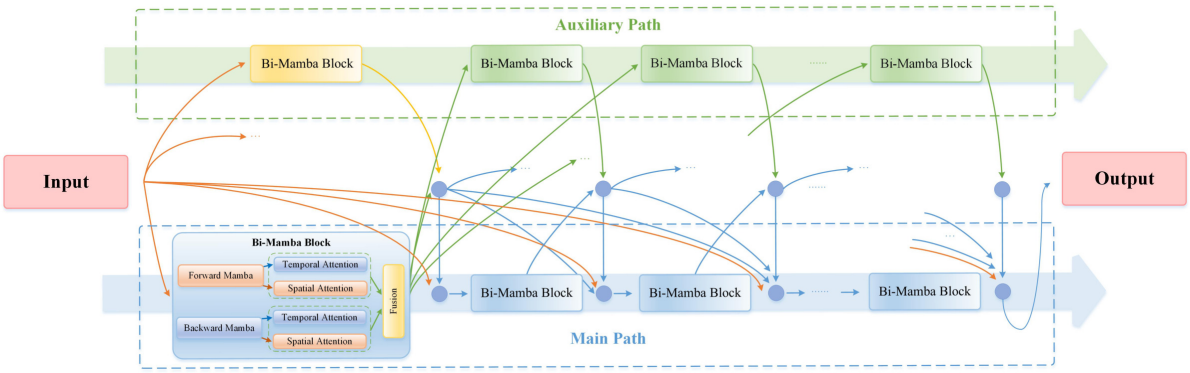


Figure 6.1: The overall architecture of the proposed ResBi-Mamba Plus network. The model is composed of residual blocks that integrate Bi-Mamba and attention modules as fundamental building units. These blocks are stacked and interact along two parallel residual pathways, enhancing information flow throughout the network and enabling the construction of deep Mamba-based models.

dencies across different time steps and features in multivariate time series along both temporal and spatial dimensions. Additionally, the ResNet Plus architecture is also employed to increase the model's depth, making the proposed ResBi-Mamba Plus significantly more effective than current Mamba-family models in both depth and long-sequence modeling.

- To the best of our knowledge, this chapter is the first to explore the potential applications of the Mamba architecture for fine-grained hourly runoff forecasting. We demonstrate the effectiveness and superiority of the deeply enhanced ResBi-Mamba Plus framework through detailed ablation studies and comparative experiments.

6.1.1 Proposed Method

The overall architecture of the proposed multivariate hourly runoff forecasting framework ResBi-Mamba Plus is shown in Figure 6.1. The primary strengths of ResBi-Mamba Plus lie in its depth and scalability, enabling the construction of deeper models and the

processing of longer input sequences. Its backbone is built upon the data-driven SSM Model: Mamba, which combines parallel processing with linear complexity. To overcome the depth limitations of vanilla Mamba, we incorporate the ResNet Plus architecture into the network design. This architecture features two parallel residual paths interconnected via dense shortcut connections, effectively mitigating gradient vanishing and performance degradation issues, and substantially increasing the model's effective depth. To further enhance the performance of each residual block, we draw inspiration from the bidirectional structure proposed in Chapter 5 and redesign the novel Bi-Mamba modules within each residual block to capture richer contextual information through additional backward processing. Moreover, a multi-dimensional attention module is embedded to adaptively and explicitly model the importance of features across temporal and spatial dimensions for both directional branches of Bi-Mamba. This ensures that critical elements are prominently represented when the two branches are fused. With these features, ResBi-Mamba Plus not only can stack to deep layers to learn high-order abstract features but also maintains robust performance over extended periods, setting a new benchmark for hourly runoff forecasting.

6.1.2 From State Space Model to Mamba

Mamba is a novel milestone in time series forecasting architectures, following RNN and Transformer, capable of striking a better balance between performance and computational complexity in tasks involving long sequences. The Mamba architecture is fundamentally based on State Space Model (SSM) [165], a mathematical framework originating from modern control system theory, used to describe the state representation of systems at each time step. The core of SSM is to map the input x_t to the output y_t through the hidden state h_t using first-order differential equations. This process can be

formulated as:

$$h(t)' = \mathbf{A}h(t) + \mathbf{B}x(t), \tag{6.1}$$

$$y(t) = \mathbf{C}h(t) \tag{6.2}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times D}$ and $\mathbf{C} \in \mathbb{R}^{N\times D}$ represent three learnable parameter matrices, corresponding to the state transition matrix, input matrix, and output matrix, respectively. N denotes the state dimension, while D represents the variable dimension. The original SSM was designed to deal with continuous functions. However, to model the real-world discrete time series , it is necessary to transform the continuous parameters $(\Delta, \mathbf{A}, \mathbf{B})$ into discrete parameters $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ using the zero-order hold (ZOH) method, where Δ is a special time scale parameter used to control the updating of the state at each time step. The discretization process can be expressed as:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}) \tag{6.3}$$

$$\overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \tag{6.4}$$

The formula for the discretised SSM is then given by:

$$h_k = \overline{\mathbf{A}}h_{k-1} + \overline{\mathbf{B}}x_k \tag{6.5}$$

$$y_k = \mathbf{C}h_k \tag{6.6}$$

where k represents the discrete time step. The discretized SSM exhibits unique dual attributes of convolution and recursion. During the training phase, it employs convolutional operation to compute in parallel, enhancing training efficiency. In the inference

phase, outputs can be rapidly generated through recursion. However, SSM still suffers from the long-term dependencies problem. To address this, Gu et al. replace the randomly initialized matrix $\mathbf{A}$ with the HiPPO matrix [166], significantly improving the SSM's retention of historical information. This development laid the foundation for the precursor to Mamba: the structured state space model (S4). Building upon S4, Mamba introduces two significant improvements: a data-driven selection mechanism and a new hardware-aware parallel algorithm. The S4 is fundamentally a Linear Time-Invariant (LTI) model, meaning that Δ, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ remain constant at all time steps, thus the model lacks context-aware ability specific to given inputs. To address this, Mamba implements a unique selection mechanism that parameterizes Δ and matrices $\mathbf{B}$, $\mathbf{C}$ according to the input, transforming S4 into a data-driven, time-varying model. This process can be described as follows:

$$\mathbf{B} \rightarrow S_{\mathbf{B}}(x) = \text{Linear}_N(x) \tag{6.7}$$

$$\mathbf{C} \rightarrow S_{\mathbf{C}}(x) = \text{Linear}_N(x) \tag{6.8}$$

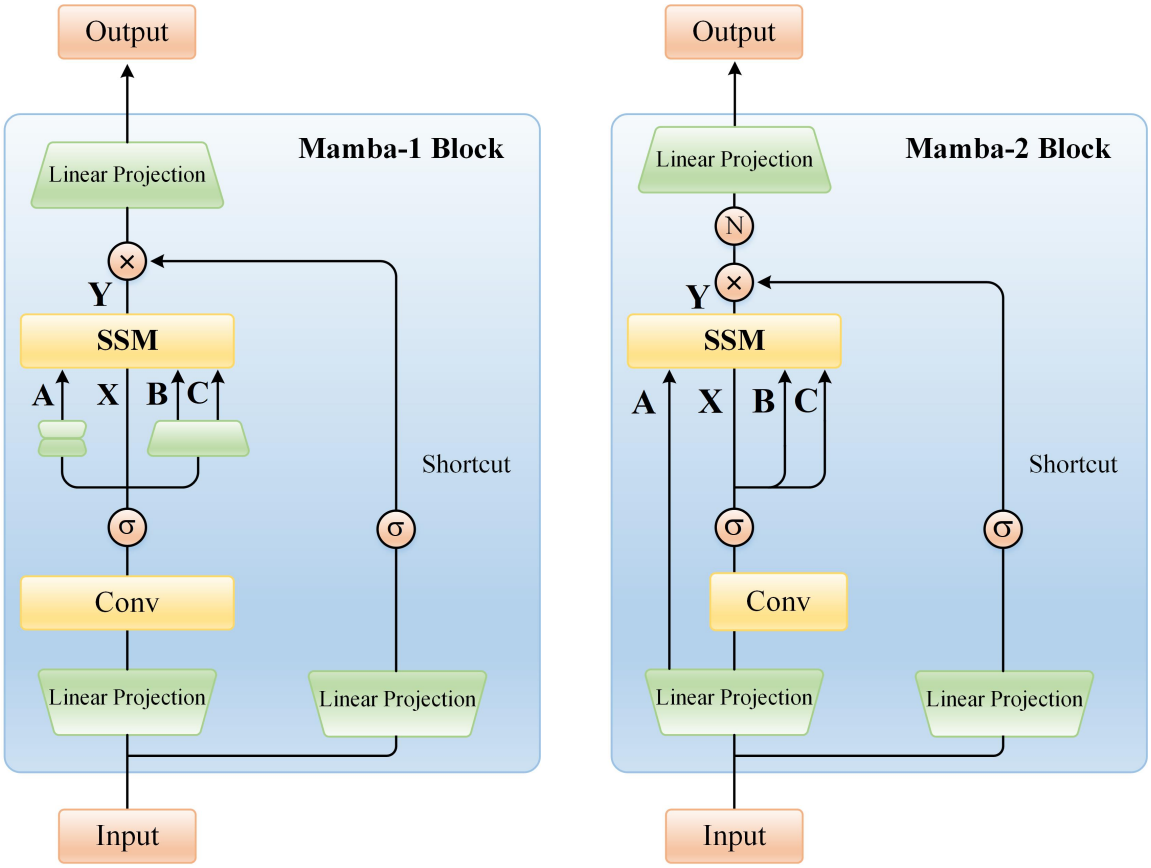
$$\Delta \rightarrow S_{\Delta}(x) = \tau_{\Delta} \cdot \text{Broadcast}_D(\text{Linear}_1(x)) \tag{6.9}$$

where Linear_N represents a parameterized projection to dimension N, and τ_{Δ} denotes the Softplus function. The selection mechanism in Mamba is similar to a combination of attention and gating mechanisms, enabling selective memory and forgetting of inputs and their projections. The interaction between parameters Δ and $\mathbf{A}$ determines the information retained by the model, while $\mathbf{B}$ and $\mathbf{C}$ make the model selective and able to filter information more finely. However, the time-varying selection mechanism also compromises Mamba's equivalence to convolution. To offset the loss in efficiency, Mamba

uses a new recursive scanning algorithm as an alternative to the original convolutional operation. This method computes the input sequence in chunks and combines them iteratively by constructing a balanced binary tree, ensuring that the model can still utilize modern accelerators for parallel processing. Additionally, Mamba performs discretization and recursive computations directly in GPU SRAM to reduce I/O operations, and employs the recomputation technique during backpropagation to recalculate intermediate states, thereby reducing memory requirements and significantly enhancing the efficiency of processing long sequences. Mamba block serves as the fundamental component of the model and its structure is shown in Figure 6.1.2. In terms of architectural design, the simplified Mamba block combines the H3 architecture, which is the basis of SSM, with the gated MLP and can be stacked flexibly like residual blocks.

6.1.3 Bidirectional Mamba-2 with Spatiotemporal Attention Mechanism

Building on Mamba-1, Dao et al. further unify the SSM architecture with attention mechanism and propose the Mamba-2 model based on Structured State Space Duality (SSD) [167]. The SSD framework views both the linear time-invariant system of SSM and the attention mechanism as semiseparable matrix transformations, allowing SSM operations to be converted into structured matrix multiplications. To further enhance model's training efficiency on accelerators, the SSD layer in Mamba-2 optimizes SSM matrix computations through the block decomposition algorithm. Diagonal blocks utilize dual attention for intra-block computations, while low-rank off-diagonal blocks are decomposed according to the properties of semi-separable matrices for inter-block computations. As a result, Mamba-2, which is based on matrix operations, is able to accelerate SSM using the tensor core, resulting in a 2-8 times increase in training speed. In addition, as shown in Figure 6.1.2, the architecture of the Mamba-2 block is also



(a) Mamba-1. (b) Mamba-2.

Figure 6.2: The internal architectures of the Mamba-1 and Mamba-2 blocks. The green trapezoid denotes linear projection, the yellow rectangle represents sequence transformations, and the orange dot represents nonlinearity.

optimised by replacing the SSM-centric sequential linear projection with the parallel parametric projection. In this configuration, (**A**, **B**, **C**) are no longer considered as auxiliary parameters but rather the SSD layer acts as a map from (**A**, **X**, **B**, **C**) to **Y**, which allows for better applicability of tensor-parallelism to the input projection. Moreover, Mamba-2 incorporates an extra normalization layer before the final output projection to ensure training stability. Given these advantages, our proposed ResBi-Mamba Plus employs Mamba-2 as its core backbone to construct the overall model.

The advantages of the Mamba-2 model in terms of efficiency and long sequence modelling enable it to maintain high temporal robustness over extended forecasting

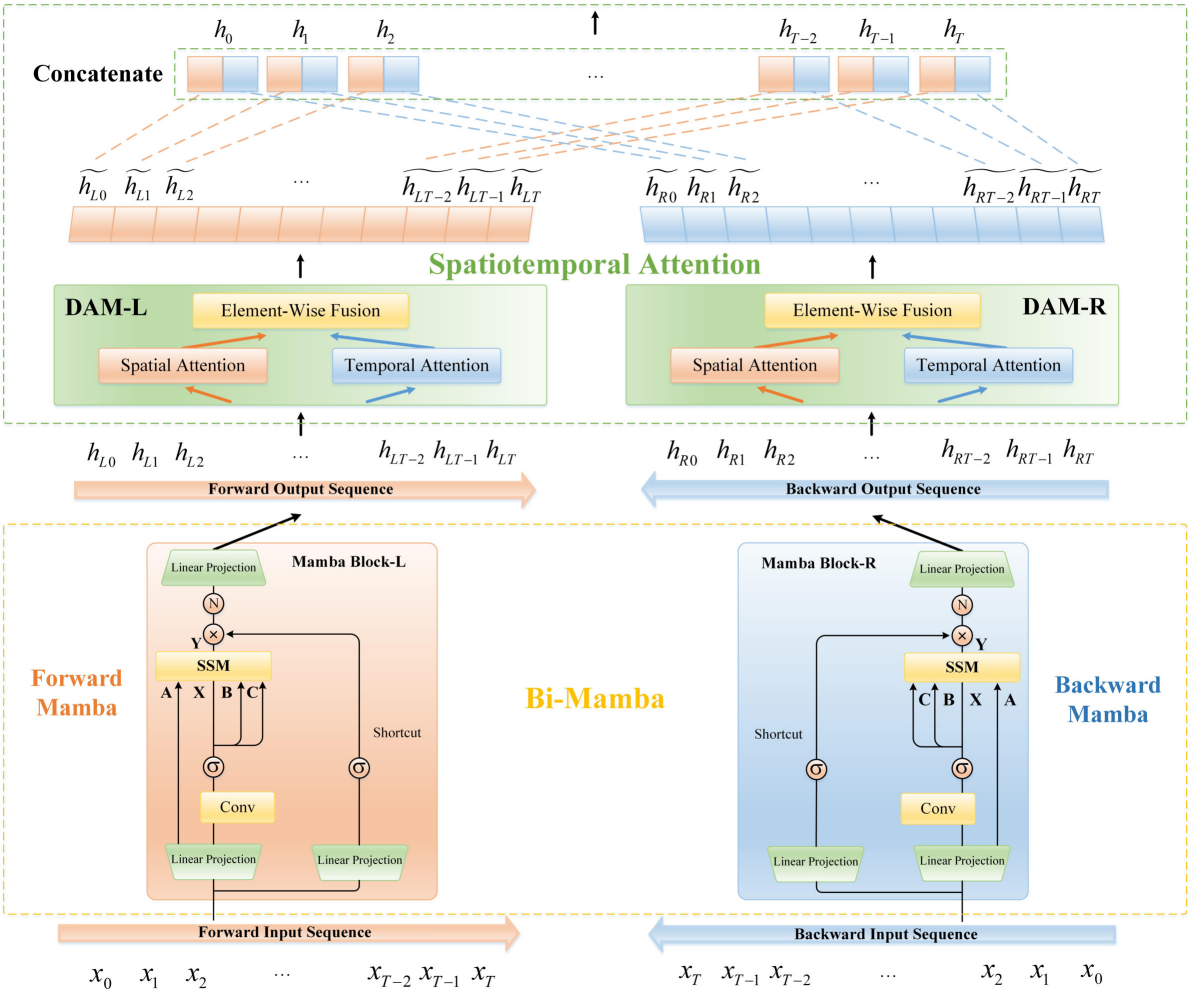


Figure 6.3: The proposed Bi-Mamba architecture, which consists of two parallel branches: Forward Mamba and Backward Mamba. These branches traverse the input sequence in opposite directions, and their outputs are adaptively recalibrated through the subsequently embedded spatiotemporal attention module.

horizons. However, increasing the length of input sequences typically brings richer contextual information, while existing Mamba models mostly rely on traditional unidirectional structures, traversing from past to present, which inevitably overlooks some complex interdependencies among sequences. Siami-Namini et al. and Liang et al. have demonstrated that LSTM and SSM models enhanced with bidirectional architectures outperform their unidirectional counterparts on multiple tasks[152, 168]. Moreover, the results presented in Chapter 5 further substantiate the advantages of this design for

modeling long sequences. Inspired by this, we further improve the bidirectional architecture and propose the first Bi-Mamba model based on the latest Mamba-2 backbone, as illustrated in Figure 6.3. Within each block, an additional Mamba has been incorporated to traverse the input sequence in reverse order, from x_t to x_0. This newly added reverse traversal is also based solely on historical data known at the current time step, thereby preventing any leakage of future information during training and inference. Both forward and backward computations are independent and parallel, and benefiting from Mamba's linear complexity and hardware acceleration, Bi-Mamba achieves a favorable balance between accuracy and computational efficiency. This ensures that it can be linearly stacked in multiple layers to enhance performance like traditional Mamba blocks.

Time series forecasting models typically input historical information that is multiple times the length of the forecasting horizon to ensure the sequence contains sufficient contextual information for modeling. To address the challenges posed by long sequences, we further integrate the attention mechanism into each Bi-Mamba block [169], enabling the model to dynamically focus on critical parts of the input feature map. The proponents of Mamba-2 believe that the integration of attention mechanisms will be a significant future development direction for the Mamba family models. Essentially, there is a complementary relationship between the Mamba layer and the attention layer in the model, i.e., the SSM establishes an initial mapping from input to output, and then the attention module retrieves and labels the important elements in the sequences, which can work synergistically to improve the efficiency and accuracy. Dao et al. have also demonstrated through empirical studies that combining attention with SSM significantly improves the forecasting performance of the model compared to the vanilla Mamba model [167]. In real-world applications, time series data often contain multiple variables. Consequently, in multivariate time series forecasting, original sequences are considered as multi-channel signals where each channel represents an independently

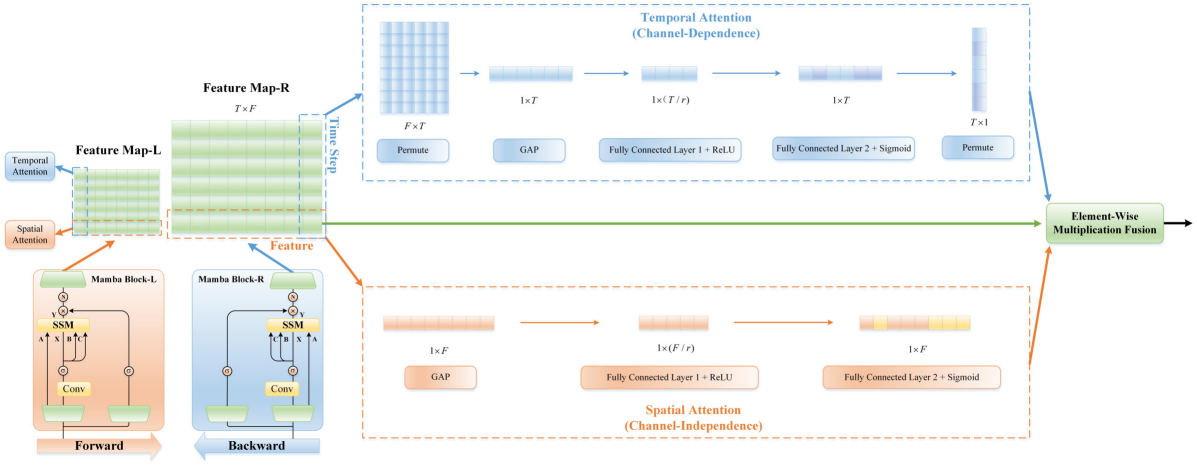


Figure 6.4: The architecture of the proposed multidimensional attention module DAM. This plug-and-play module is designed to be inserted after the Bi-Mamba layer, enabling explicit modeling of feature importance in both temporal and spatial dimensions based on CD and CI strategies, respectively.

recorded univariate series. Mainstream attention mechanisms employed in time series forecasting models, such as the self-attention of Transformer, mainly focus on a single dimension, usually the temporal dimension. This channel-dependence (CD) modeling strategy inherently amalgamates distinct variables into hybrid temporal features at each time step. The consequent loss of independence and receptive field characteristics potentially constrains model performance in complex hydrological and meteorological scenarios requiring long-term forecasting capability. Emerging SOTA architectures like iTransformer have progressively adopted the channel-independence (CI) strategy, which treats the entire sequence of each variable as the primary subject of the model's description. This inverted modeling approach has demonstrated empirical efficacy in enhancing forecasting performance. However, the CI strategy struggles to explicitly model inter-channel dependencies, which becomes particularly problematic when the number of channels is large.

To overcome the limitation of existing attention mechanisms that primarily focus on a single dimension, it is necessary to further quantify the importance of output

elements from the forward and backward Mamba branches across different dimensions. Therefore, we deeply integrate the spatiotemporal attention module DAM, proposed in Chapters 4 and 5 , with Bi-Mamba. Specifically, the CD strategy is applied in the temporal dimension to emphasize the sequential correlations among features, while the CI strategy is used in the spatial dimension to assess the relative importance of individual variable channels. The structure is demonstrated in Figure 6.4. Moreover, conventional bidirectional architectures often adopt simple element-wise concatenation to fuse outputs from two independent branches. In contrast, the plug-and-play nature of the DAM module allows it to be flexibly embedded at arbitrary positions within time series forecasting models, enabling it to replace traditional concatenation-based fusion. In the proposed ResBi-Mamba Plus, DAM is inserted after each unidirectional branch in Bi-Mamba, where it explicitly recalibrates the importance of elements in the output feature maps. The two branches recalibrated by attention weights can highlight their respective key elements from a global perspective, enabling more efficient fusion of bidirectional multivariate time series. This enhancement further improves ResBi-Mamba Plus's temporal robustness when modeling long sequences.

6.1.4 Quantitative Evaluation of Multi-Lead-Time Runoff Forecasting

There is no universally accepted standard for evaluating the performance of runoff forecasting models in hydrology. As a result, existing studies often adopt diverse criteria based on specific application scenarios and subjective assessments. Among these criteria, most are grounded in the Nash-Sutcliffe Efficiency (NSE), a widely used quantitative metric. NSE intuitively measures the goodness-of-fit between observed and predicted values. It is dimensionless and conceptually straightforward, making it convenient for direct comparison across different regions, models, and variables. For these reasons,

Performance Rating	**Ecrepont Criterion**	**Ritter Criterion**
Very Good	> 0.66	≥ 0.90
Good	$0.33 - 0.66$	$0.80 - 0.90$
Average/Acceptable	$0 - 0.33$	$0.65 - 0.80$
Poor/Unsatisfactory	< 0	≤ 0.65

Table 6.1: Evaluation Criteria for Hydrological Models.

NSE is regarded as a core evaluation indicator in international hydrological research. Ecrepont et al. suggest that the performance of a hydrological model can be classified into four categories based on the NSE value: "very good" when > 0.66, "good" when $0.33 - 0.66$, "average" when $0 - 0.33$ and "poor" when < 0 [170]. Ritter et al. propose a more stringent evaluation standard, where NSE value ≥ 0.90 is considered "very good", $0.80 - 0.90$ is "good", $0.65 - 0.80$ is "acceptable", and ≤ 0.65 is classified as "unsatisfactory" [118]. The comparison between these two evaluation criteria is presented in Table 6.1.

It can be observed that the two evaluation metrics differ substantially, primarily because the rapid development of deep learning has rendered less stringent standards inadequate for capturing the fast growth in model performance. Therefore, in this thesis, the more rigorous Ritter evaluation criterion is adopted for assessing the proposed high-performance neural network models. Drawing on the work of Ritter et al. and considering the current state of research in the field, we further refine the performance evaluation for hourly runoff forecasting models across multiple horizons (i.e., temporal robustness). A model is considered to achieve SOTA temporal robustness in multi-lead-time forecasting only if it consistently maintains NSE≥ 0.90 across all intervals within the 0-24 hour horizon. This quantitative standard imposes stricter requirements on both short-term and long-term forecasting capabilities. Models with true multi-lead-time forecasting ability must not only achieve "very good" accuracy in a single interval but also sustain this level of performance throughout the entire 24-hour horizon without noticeable degradation.

6.1.5 Model Design and Implementation Details

As a multi-lead-time forecasting model, ResBi-Mamba Plus is further extended to forecast hourly runoff from 2 hours up to 48 hours in the future. In terms of network architecture, ResBi-Mamba Plus supports the construction of dual residual block pathways with a depth of up to 6 layers, while in DAM, the dimensionality reduction hyperparameter r is set to 4. The dropout regularization is also added after each unidirectional Mamba layer to prevent overfitting, with the dropout rate set to 0.2. The study area of this chapter is the Columbia River in Western North America. The model is iterated 100 times using the Adam optimizer. Experiments are conducted on NVIDIA RTX 3070, based on Pytorch 1.12.1 + CUDA 12.8 environment. Due to the adoption of a new deep learning framework, some experimental results in this chapter differ significantly from those presented in the previous chapters.

6.2 Experiment Results and Discussion

6.2.1 Ablation Study

Compared to the baseline model Mamba-2, our proposed ResBi-Mamba Plus introduces multiple principal enhancements, including the bidirectional architecture combined with the intra-block design that integrates dual path multidimensional DAM module. At the network level, it further improves information flow in deep layers through an additional auxiliary residual pathway and dense shortcuts. To validate the effectiveness of our methodology, we conduct comprehensive ablation studies examining the performance contributions of individual components at varying network depths. The forecasting horizon is consistently set to 48 hours, and the model depth refers to the number of residual blocks rather than the total number of network layers.

Table Table 6.2 presents the results of the ablation studies, clearly illustrating that

Model	Depth	MAE	MAPE	NSE
Mamba-2		7.38	10.69	0.845
Bi-Mamba-DAM	1	**7.02**	**9.77**	0.856
ResBi-Mamba Plus		7.04	10.25	**0.858**
Mamba-2		7.24	10.11	0.852
Bi-Mamba-DAM	2	7.08	9.85	0.857
ResBi-Mamba Plus		**6.79**	**9.42**	**0.868**
Mamba-2		7.28	10.11	0.851
Bi-Mamba-DAM	4	7.13	10.33	0.854
ResBi-Mamba Plus		**6.43**	**9.29**	**0.878**
Mamba-2		7.71	11.43	0.831
Bi-Mamba-DAM	6	7.39	10.39	0.845
ResBi-Mamba Plus		**6.22**	**8.86**	**0.887**

Table 6.2: Performance comparison at different depths. Best results per group are highlighted in bold.

ResBi-Mamba Plus outperforms other models in most experimental settings, particularly in deeper network configurations. Specifically, while the baseline model Mamba-2 and the Bi-Mamba-DAM variant exhibit slight performance gains at shallower depths, their errors increase substantially as the network depth reaches six residual blocks. Notably, the error of Mamba-2 rises by 6.5%, and that of Bi-Mamba-DAM increases by 5.3%, indicating that even SOTA models within the Mamba family struggle to mitigate issues such as vanishing gradient and performance degradation in deep networks. In contrast, ResBi-Mamba Plus, which adopts the ResNet Plus architecture, continues to benefit from increased depth. Its error decreases by approximately 11.6%, demonstrating that the model effectively addresses the adverse effects of deep networks and significantly enhances forecasting accuracy by learning higher-level abstract representations. While mainstream Mamba-based models typically consist of only 2-3 Mamba blocks, ResBi-Mamba Plus more than doubles this depth. It is also worth noting that each Bi-Mamba block comprises dozens of internal layers performing diverse functions, meaning that at the depth of 6, the actual number of hidden layers in the model exceeds one hundred.

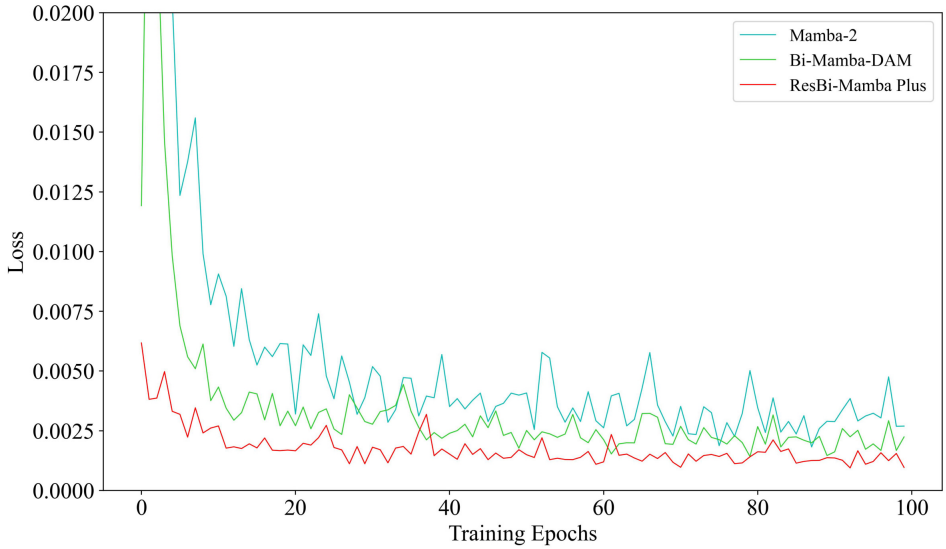


Figure 6.5: Training loss curves of different models in the ablation study (depth = 6, forecasting horizon = 48).

Moreover, Bi-Mamba-DAM consistently outperforms the baseline model across all depths, proving that the integration of the additional backward traversal and the adaptive recalibration of features in different dimensions via the DAM module effectively enhances forecasting accuracy by enabling richer contextual modeling. Figure 6.5 further demonstrates the loss curves of models with different modules during the training process. It is evident that ResBi-Mamba Plus achieves the fastest convergence, which can be attributed to the dense shortcut connections between parallel residual pathways. These connections establish shorter routes for gradient backpropagation, thereby accelerating the optimization process. In addition, the convergence speed of Bi-Mamba-DAM is also significantly improved. This enhancement is due to the application of attention mechanisms in both temporal and spatial dimensions, which enables the model to rapidly and adaptively focus on critical features. As a result, the error signal can more efficiently propagate in directions that have a greater impact on loss reduction, further contributing to faster and more stable training.

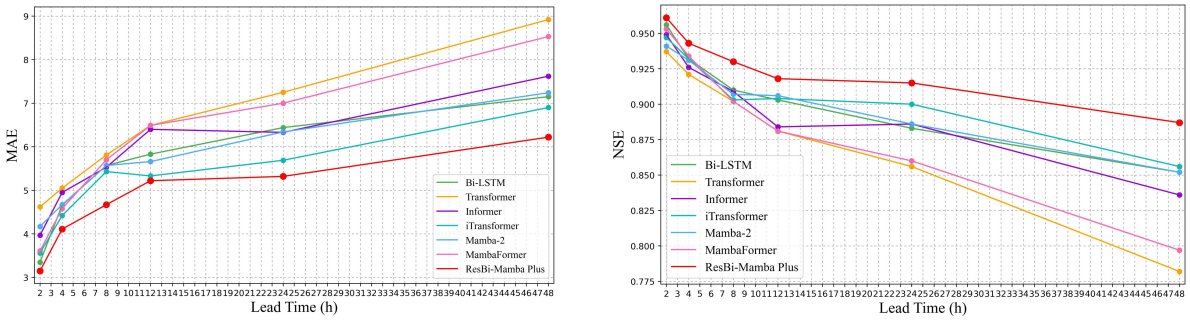
6.2.2 Comparative Experiments with Mainstream Models

We conduct comprehensive comparative experiments between the proposed ResBi-Mamba Plus and two representative categories of mainstream models that are most widely used in the field of runoff forecasting, as introduced in Section 6.1. To evaluate the temporal robustness of the models across different forecasting horizons, we test them under six lead times ranging from short to long: 2, 4, 8, 12, 24, and 48 time steps. The models included in the comparison are as follows:

- **Bi-LSTM** adopts a novel bidirectional recurrent architecture and is regarded as one of the most advanced and high-performing variants of LSTM.
- **Transformer** replaces traditional recurrent architectures with the self-attention mechanism and has become one of the most widely used models in deep learning.
- **Informer** builds upon the traditional Transformer by incorporating the Prob-Sparse self-attention mechanism and the self-attention distilling strategy to reduce computational complexity and parameter count. It is considered one of the SOTA time series forecasting models.
- **iTransformer** adopts the inverted modeling method based on the CI strategy, treating variables as the primary dimension to learn feature representations of multivariate time series. It stands as one of the SOTA models for long-term forecasting tasks.
- **Mamba-2** is the latest paradigm within the SSM family and also serves as the backbone of our proposed methodology.
- **MambaFormer** is a popular hybrid model in time series forecasting [171], combining the respective strengths of the Mamba and Transformer architectures. It continues to evolve with the emergence of new variants.

Mainstream runoff forecasting models are currently undergoing a paradigm shift from single-step to multi-step forecasting. As a result, model performance across various forecasting horizons has become a critical focus of our analysis. Table 6.3 and Figure 6.6 present the results of comparative experiments. Specifically, at short to medium lead times, traditional forecasting methods achieve comparable or even slightly superior performance to outstanding Transformer and Mamba-family models. Notably, Bi-LSTM, which also adopts the bidirectional architecture, exhibits strong performance at multiple lead times, indirectly validating the effectiveness of the additional backward traversal. As the lead time increases, iTransformer and Mamba-2, both representing the latest SOTA models within their respective branches, demonstrate remarkable temporal robustness in long-term forecasting. The CI-based iTransformer demonstrates substantial performance improvement over vanilla Transformer through capturing intra-channel dependencies, establishing itself as the second-best model across several lead times. According to the criteria outlined in Section 6.1.4, iTransformer is the only baseline model that achieves SOTA multi-lead-time performance over the 24-hour forecasting horizon. In contrast, the vanilla Transformer suffers from its quadratic complexity, where the growth of the attention matrix in longer sequences impedes effective training. Consequently, even the hybrid MambaFormer, which integrates Mamba-2 and Transformer components, still underperforms in long-term horizons. The results clearly show that across all six evaluated lead times, our proposed ResBi-Mamba Plus outperforms all competing models in terms of various evaluation metrics, highlighting its superior overall performance. It is worth noting that ResBi-Mamba Plus consistently maintains SOTA-level temporal robustness with NSE ≥ 0.90 across all forecasting horizons within the next 24 hours. This effectively addresses the problem of performance degradation over long forecasting horizons.

Furthermore, compared with the three models previously proposed in this thesis,



(a) MAE (b) NSE

Figure 6.6: MAE and NSE of mainstream time series forecasting models at various lead times.

ResBi-Mamba Plus achieves a substantial extension of the forecasting horizon through progressive architectural improvements while maintaining comparable performance levels. Specifically, the horizon for which temporal robustness is sustained has been tripled relative to EA-TCN presented in Chapter 3, and has more than doubled compared to the stronger deep models ResTCN-DAM and STResBiGRU. These results further demonstrate the remarkable performance advantage of the SSM-based Mamba architecture over mainstream convolutional and recurrent architectures in time series forecasting tasks.

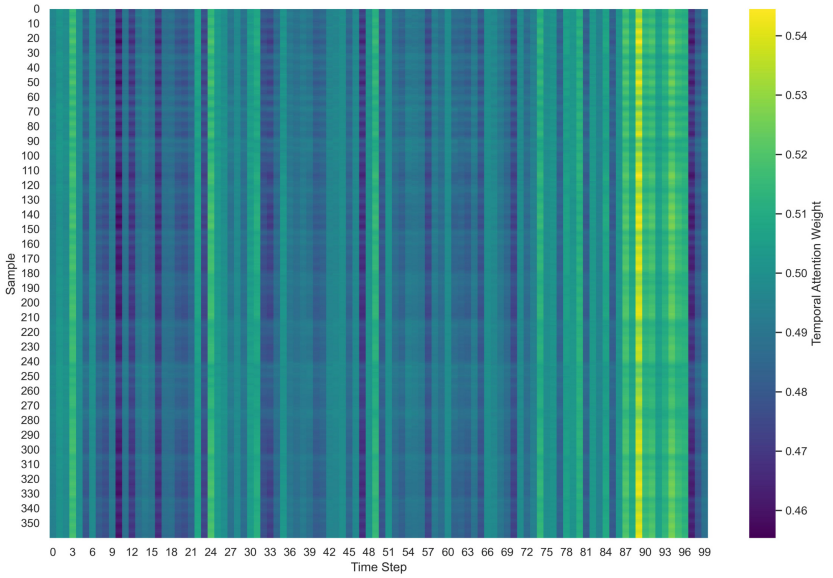
To rigorously assess the models' temporal robustness at the extremes of its forecasting capability, the forecasting horizon is extended beyond conventional limits. At the longest horizon (lead time = 48), ResBi-Mamba Plus achieves a 9.9% reduction in error compared to iTransformer, the widely recognized SOTA model for long-term forecasting. This demonstrates that our proposed methodology based on both CD and CI strategies is more suitable for long-term, fine-grained runoff forecasting than single-strategy models like iTransformer. Furthermore, the performance advantage of ResBi-Mamba Plus is even more pronounced when compared to other mainstream models: it achieves 30.3% and 18.4% lower errors than Transformer and Informer, respectively, strongly validating the effectiveness and advancement of our proposed framework.

Model	Lead Time	MAE	MAPE	NSE
Bi-LSTM	2	3.35	4.62	0.956
Transformer		4.62	6.76	0.937
Informer		3.97	5.65	0.949
iTransformer		3.56	5.09	0.947
Mamba-2		4.17	6.63	0.941
MambaFormer		3.61	5.49	0.953
ResBi-Mamba Plus		**3.15**	**4.52**	**0.961**
Bi-LSTM	4	4.66	6.75	0.932
Transformer		5.05	7.51	0.921
Informer		4.95	7.10	0.926
iTransformer		4.42	6.33	0.932
Mamba-2		4.67	6.61	0.931
MambaFormer		4.58	6.43	0.934
ResBi-Mamba Plus		**4.11**	**5.89**	**0.943**
Bi-LSTM	8	5.57	7.89	0.910
Transformer		5.81	8.10	0.902
Informer		5.53	7.77	0.909
iTransformer		5.43	7.85	0.902
Mamba-2		5.57	8.17	0.907
MambaFormer		5.70	8.31	0.902
ResBi-Mamba Plus		**4.67**	**6.60**	**0.930**
Bi-LSTM	12	5.83	7.80	0.903
Transformer		6.49	9.07	0.881
Informer		6.40	9.59	0.884
iTransformer		5.33	7.60	0.904
Mamba-2		5.66	8.07	0.906
MambaFormer		6.49	9.03	0.881
ResBi-Mamba Plus		**5.22**	**7.19**	**0.918**
Bi-LSTM	24	6.44	8.77	0.883
Transformer		7.25	9.95	0.856
Informer		6.33	9.11	0.886
iTransformer		5.69	8.03	0.900
Mamba-2		6.34	8.69	0.886
MambaFormer		7.00	9.45	0.860
ResBi-Mamba Plus		**5.32**	**7.74**	**0.915**
Bi-LSTM	48	7.15	10.69	0.852
Transformer		8.92	12.9	0.782
Informer		7.62	11.18	0.836
iTransformer		6.90	9.83	0.856
Mamba-2		7.24	10.11	0.852
MambaFormer		8.53	11.45	0.797
ResBi-Mamba Plus		**6.22**	**8.86**	**0.887**

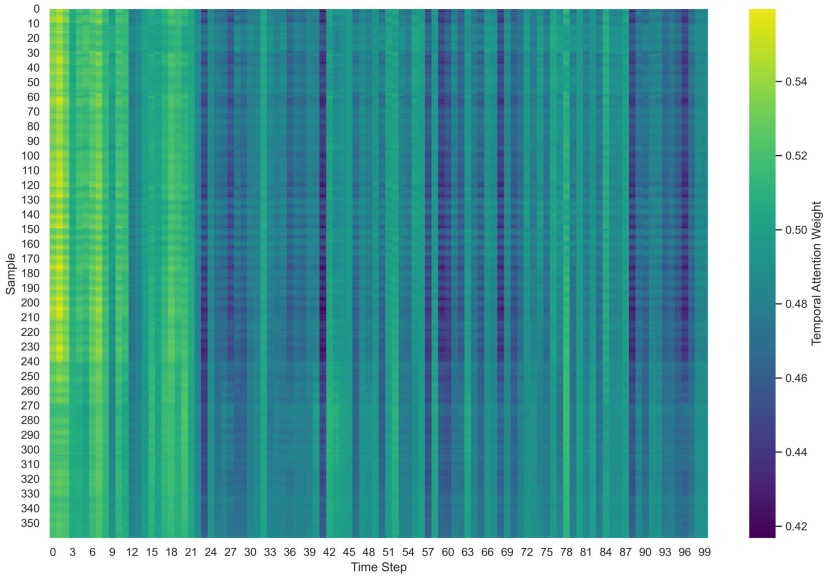
Table 6.3: Performance comparison at different lead times.

6.2.3 Interpretability of Attention Module

To further enhance the transparency of the forecasting process, we employ the IML approach to visualize the temporal attention weights of the DAM module. Figure 6.7 shows



(a) Weight heatmap of the forward Mamba.



(b) Weight heatmap of the backward Mamba.

Figure 6.7: Temporal attention weight heatmaps of forward and backward Bi-Mamba branches.

the weight heatmaps for the forward and reverse Bi-Mamba branches, where brighter colours represent higher weights assigned to features at this time step. It is evident that in the forward branch's heatmap, features toward the right (near the forecasting horizon) are given higher weights. Conversely, in the backward branch which processes the

input sequence in reverse, features on the left side of the heatmap (corresponding to the same region near the forecasting horizon) are emphasized. These patterns indicate that the attention module's temporal weighting preferences align closely with fundamental time-series characteristics. Specifically, correlation between time-series elements decays as their temporal distance increases. Thus, observations closer in time typically exhibit stronger correlations and better reflect the series' current trend, a short-term memory effect that the DAM module effectively captures. It is noteworthy that time series may also exhibit specific long-term dependencies characterized by a relatively slow decay in the autocorrelation function, resulting in strong correlations even across large temporal intervals. This phenomenon is clearly reflected in the heatmaps as well.

6.3 Conclusion

In this chapter, we propose a novel multi-lead-time runoff forecasting framework: ResBi-Mamba Plus, which demonstrates exceptional temporal robustness and achieves accurate forecasts across all forecasting horizons within the next 24 hours. For the first time, we introduce the emerging SSM-based deep learning architecture Mamba-2 to fine-grained hourly runoff forecasting. This architecture exhibits dual attributes of convolution and recursion while maintaining linear computational complexity. Furthermore, the ResNet Plus architecture with dense shortcut connections is incorporated to complement the Mamba model, enabling substantial performance gains through increased network depth. To comprehensively capture the rich contextual information within input sequences, we designe a bidirectional backbone, Bi-Mamba, and insert our designed DAM module after each unidirectional branch. The DAM module adaptively recalibrates features along the temporal and spatial dimensions based on CD and CI strategies, respectively, thereby enhancing the model's temporal robustness, particularly for long-term forecasting tasks. Multiple sets of ablation and comparative experiments prove the effectiveness and

superiority of ResBi-Mamba Plus, demonstrating that it consistently outperforms current SOTA models across various evaluation metrics. Our framework establishes a new paradigm for applying Mamba-family models to runoff forecasting tasks.

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH

7.1 Conclusion

Accurate forecasting of fine-grained hourly runoff, which belongs to non-stationary time series, has long been a challenging research direction in hydrology. This thesis conducts a comprehensive optimization and enhancement of existing neural network-based runoff forecasting models by addressing their common limitations in terms of accuracy, efficiency, and long-term forecasting capability. Building upon these improvements, a series of high-performance hourly runoff forecasting frameworks are progressively proposed, incorporating diverse emerging deep learning techniques. The primary contributions of this work are summarized as follows:

Lightweight Robust Forecasting Framework: We propose an innovative forecasting framework, EA-TCN, which achieves a balanced trade-off among performance, efficiency, and robustness. This framework replaces the parameter-heavy recurrent architecture with the lightweight TCN and integrates a plug-and-play attention module to enhance accuracy. The Snapshot ensemble method is also used to achieve the effect

of training multiple individual models through one single training process, thereby improving the model's robustness against data perturbations.

Dual-Pathway ResNet and Spatiotemporal Attention: To overcome the performance bottlenecks caused by inherent structural limitations in existing models, we propose ResTCN-DAM, a novel architecture built upon the enhanced ResNet Plus framework featuring dual residual block pathways and dense shortcut connections. This design effectively mitigates the degradation problem by optimizing information flow in deep networks, thereby enabling consistent performance gains as model depth increases. Furthermore, the conventional attention mechanisms, which typically focus on a single dimension, are extended to both the temporal and spatial dimensions. Leveraging CD and CI strategies, the proposed multidimensional DAM module can simultaneously and explicitly model the interdependencies between features in different dimensions, thereby selectively emphasizing critical features while suppressing irrelevant ones.

Bidirectional Recurrent Architecture and Temporal Shortcuts: The STResBi-GRU framework is developed in response to the limitations of recurrent-based neural networks in modeling long-term dependencies within sequences. Its backbone incorporates the redesigned BiGRU architecture that processes inputs in both forward and backward directions, thereby capturing richer contextual information. To promote effective integration of the two directional branches, the spatiotemporal attention module DAM is embedded into the architecture. Moreover, temporal shortcut connections are inserted between GRU cells to enable lossless transmission of information across time steps, which effectively mitigates the vanishing gradient problem in the temporal dimension.

Multi-Lead-Time Forecasting Based on State Space Model: To ensure temporal robustness in multi-step forecasting scenarios, ResBi-Mamba Plus is proposed to maintain leading performance across varying forecasting horizons. This framework introduces the advanced Mamba-2 into fine-grained hourly runoff forecasting for the first

time. Mamba-2 inherits the dual attributes of convolution and recursion from SSM while preserving parallel processing capability and linear complexity. These properties enable its seamless integration with bidirectional architecture and facilitate the construction of deep networks to enhance long-sequence modeling. Experimental results demonstrate that ResBi-Mamba Plus achieves superior multi-lead-time forecasting performance, significantly outperforming existing mainstream models.

Interpretable Forecasting Process: The excessive emphasis on complex model architectures and predictive accuracy often comes at the expense of transparency in the forecasting process. To address this concern, this thesis incorporates a model-specific local post-hoc explanation technique grounded in IML. Specifically, the temporal attention weights of the multidimensional attention module DAM are visualized in the form of heatmaps. The visual analysis confirms that the patterns learned by the model align with human cognition and objective principles, thereby significantly enhancing the interpretability of the proposed forecasting framework.

7.2 Future Research

Our future research will focus on the following key directions:

Multimodal Data Fusion: The data used in this thesis is derived from single-modal historical time series. In future research, we plan to incorporate multimodal data at the model input stage by integrating diverse types and sources of information, such as satellite remote sensing, meteorological radar observations, and various forms of external prior knowledge. Multimodal forecasting is expected to compensate for the limitations of single-source data and enhance the model's robustness and generalization capability by leveraging richer and comprehensive information. From the engineering perspective, multimodal data fusion offers significant potential for real-world applications. In oper-

ational hydrology and environmental management, assimilating diverse data streams in near real time can enhance predictive accuracy, improve robustness to missing or noisy measurements, and support scalable, automated forecasting systems for practical deployment.

Global Feature Interpretation: The interpretability analysis conducted in this thesis is currently limited to model-specific local explanations. In future work, we plan to extend the current interpretability methods by incorporating advanced IML techniques such as LIME and SHAP. These approaches will enable model-agnostic global interpretations from the perspective of feature contributions, offering a more comprehensive understanding of the forecasting process.

General-Purpose Time Series Forecasting: Apart from the hydrological domain, time series forecasting plays a critical role in numerous real-world applications. A key focus of our future research is to enhance the cross-domain generalization capability of the proposed model, thereby enabling the high-performance multi-lead-time forecasting framework to be effectively applied to various tasks such as high-resolution financial market forecasting, patient health monitoring in healthcare, and energy consumption trend forecasting.

References

[1] A. F. Hamlet and D. P. Lettenmaier, "Effects of climate change on hydrology and water resources in the columbia river basin 1," *JAWRA Journal of the American Water Resources Association*, vol. 35, no. 6, pp. 1597–1623, 1999.

[2] P. Qin, H. Xu, M. Liu, C. Xiao, K. E. Forrest, S. Samuelsen, and B. Tarroja, "Assessing concurrent effects of climate change on hydropower supply, electricity demand, and greenhouse gas emissions in the upper yangtze river basin of china," *Applied Energy*, vol. 279, p. 115694, 2020.

[3] Y.-p. Chen, B.-j. Fu, Y. Zhao, K.-b. Wang, M. M. Zhao, J.-f. Ma, J.-H. Wu, C. Xu, W.-g. Liu, and H. Wang, "Sustainable development in the yellow river basin: Issues and strategies," *Journal of Cleaner Production*, vol. 263, p. 121223, 2020.

[4] Z. Sheng, S. Wen, Z.-k. Feng, J. Gong, K. Shi, Z. Guo, Y. Yang, and T. Huang, "A survey on data-driven runoff forecasting models based on neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1083–1097, 2023.

[5] X.-H. Le, D.-H. Nguyen, S. Jung, M. Yeon, and G. Lee, "Comparison of deep learning techniques for river streamflow forecasting," *IEEE Access*, vol. 9, pp. 71 805–71 820, 2021.

[6] Q. Zhao, Y. Liu, W. Yao, and Y. Yao, "Hourly rainfall forecast model using supervised learning algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–9, 2021.

[7] Y. Man, Q. Yang, J. Shao, G. Wang, L. Bai, and Y. Xue, "Enhanced lstm model for daily runoff prediction in the upper huai river basin, china," *Engineering*, vol. 24, pp. 229–238, 2023.

[8] X. Yuan, C. Chen, X. Lei, Y. Yuan, and R. Muhammad Adnan, "Monthly runoff forecasting based on lstm–alo model," *Stochastic environmental research and risk assessment*, vol. 32, pp. 2199–2212, 2018.

[9] W.-c. Wang, D.-m. Xu, K.-w. Chau, and S. Chen, "Improved annual rainfall-runoff forecasting using pso–svm model based on eemd," *Journal of Hydroinformatics*, vol. 15, no. 4, pp. 1377–1390, 2013.

[10] L. Ren, M. Wang, C. Li, and W. Zhang, "Impacts of human activity on river runoff in the northern area of china," *Journal of Hydrology*, vol. 261, no. 1-4, pp. 204–217, 2002.

[11] T. Xie, G. Zhang, J. Hou, J. Xie, M. Lv, and F. Liu, "Hybrid forecasting model for non-stationary daily runoff series: A case study in the han river basin, china," *Journal of Hydrology*, vol. 577, p. 123915, 2019.

[12] N. Farsi, N. Mahjouri, and H. Ghasemi, "Breakpoint detection in non-stationary runoff time series under uncertainty," *Journal of Hydrology*, vol. 590, p. 125458, 2020.

[13] Z. Sheng, Y. Cao, Y. Yang, Z.-K. Feng, K. Shi, T. Huang, and S. Wen, "Residual temporal convolutional network with dual attention mechanism for multilead-time interpretable runoff forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[14] Z. Jiang, R. Li, A. Li, and C. Ji, "Runoff forecast uncertainty considered load adjustment model of cascade hydropower stations and its application," *Energy*, vol. 158, pp. 693–708, 2018.

[15] J. Zhang, X. Chen, A. Khan, Y.-k. Zhang, X. Kuang, X. Liang, M. L. Taccari, and J. Nuttall, "Daily runoff forecasting by deep recursive neural network," *Journal of Hydrology*, vol. 596, p. 126067, 2021.

[16] M. B. Wagena, D. Goering, A. S. Collick, E. Bock, D. R. Fuka, A. Buda, and Z. M. Easton, "Comparison of short-term streamflow forecasting using stochastic time series, neural networks, process-based, and bayesian models," *Environmental Modelling & Software*, vol. 126, p. 104669, 2020.

[17] K. Douglas-Mankin, R. Srinivasan, and J. Arnold, "Soil and water assessment tool (swat) model: Current developments and applications," *Transactions of the ASABE*, vol. 53, no. 5, pp. 1423–1431, 2010.

[18] L. Ma, C. He, H. Bian, and L. Sheng, "Mike she modeling of ecohydrological processes: Merits, applications, and challenges," *Ecological Engineering*, vol. 96, pp. 137–149, 2016.

[19] H. Fan, M. Jiang, L. Xu, H. Zhu, J. Cheng, and J. Jiang, "Comparison of long short term memory networks and the hydrological model in runoff simulation," *Water*, vol. 12, no. 1, p. 175, 2020.

[20] A. Ahani, M. Shourian, and P. Rahimi Rad, "Performance assessment of the linear, nonlinear and nonparametric data driven models in river flow forecasting," *Water resources management*, vol. 32, pp. 383–399, 2018.

[21] Y. Wang, S. Guo, L. Xiong, P. Liu, and D. Liu, "Daily runoff forecasting model based on ann and data preprocessing techniques," *Water*, vol. 7, no. 8, pp. 4144–4160, 2015.

[22] Y.-c. Wu and J.-w. Feng, "Development and application of artificial neural network," *Wireless Personal Communications*, vol. 102, pp. 1645–1656, 2018.

[23] N. Xue, I. Triguero, G. P. Figueredo, and D. Landa-Silva, "Evolving deep cnn-lstms for inventory time series prediction," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1517–1524.

[24] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[25] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.

[26] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence*, 2019, pp. 49–55.

[29] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent lstm neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[30] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[31] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[32] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou, "Deep learning with a long short-term memory networks approach for rainfall-runoff simulation," *Water*, vol. 10, no. 11, p. 1543, 2018.

[33] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 631–648.

[34] D. Yang, Y. Yang, and J. Xia, "Hydrological cycle and water resources in a changing world: A review," *Geography and Sustainability*, vol. 2, no. 2, pp. 115–122, 2021.

[35] Y. Pokhrel, M. Burbano, J. Roush, H. Kang, V. Sridhar, and D. W. Hyndman, "A review of the integrated effects of changing climate, land use, and dams on mekong river hydrology," *Water*, vol. 10, no. 3, p. 266, 2018.

[36] P. W. Downs and H. Piégay, "Catchment-scale cumulative impact of human activities on river channels in the late anthropocene: implications, limitations, prospect," *Geomorphology*, vol. 338, pp. 88–104, 2019.

[37] Y. Liu, J. Zhang, and Y. Zhao, "The risk assessment of river water pollution based on a modified non-linear model," *Water*, vol. 10, no. 4, p. 362, 2018.

[38] W.-j. Niu and Z.-k. Feng, "Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management," *Sustainable Cities and Society*, vol. 64, p. 102562, 2021.

[39] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[40] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International journal of neural systems*, vol. 31, no. 03, p. 2130001, 2021.

[41] S. Gao, S. Zhang, Y. Huang, J. Han, T. Zhang, and G. Wang, "A hydrological process-based neural network model for hourly runoff forecasting," *Environmental Modelling & Software*, vol. 176, p. 106029, 2024.

[42] C. Paniconi and M. Putti, "Physically based modeling in catchment hydrology at 50: Survey and outlook," *Water Resources Research*, vol. 51, no. 9, pp. 7090–7129, 2015.

[43] C. B. Uvo and N. E. Graham, "Seasonal runoff forecast for northern south america: A statistical model," *Water Resources Research*, vol. 34, no. 12, pp. 3515–3524, 1998.

[44] V. Nourani, H. Najafi, A. B. Amini, and H. Tanaka, "Using hybrid wavelet-exponential smoothing approach for streamflow modeling," *Complexity*, vol. 2021, no. 1, p. 6611848, 2021.

[45] M. West, "Time series decomposition," *Biometrika*, vol. 84, no. 2, pp. 489–494, 1997.

[46] K. Kalpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of arima time-series," in *Proceedings 2001 IEEE international conference on data mining*. IEEE, 2001, pp. 273–280.

[47] M. Valipour, "Long-term runoff study using sarima and arima models in the united states," *Meteorological Applications*, vol. 22, no. 3, pp. 592–598, 2015.

[48] R. Lewis and G. C. Reinsel, "Prediction of multivariate time series by autoregressive model fitting," *Journal of multivariate analysis*, vol. 16, no. 3, pp. 393–411, 1985.

[49] H. RP, "The moving average model for spatial interaction," *Transactions of the Institute of British Geographers*, pp. 202–225, 1978.

[50] J. C. Palomares-Salas, J. J. De la Rosa, J. G. Ramiro, J. Melgar, A. Aguera, and A. Moreno, "Arima vs. neural networks for wind speed forecasting," in *2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. IEEE, 2009, pp. 129–133.

[51] Q. Zhang, B.-D. Wang, B. He, Y. Peng, and M.-L. Ren, "Singular spectrum analysis and arima hybrid model for annual runoff forecasting," *Water resources management*, vol. 25, no. 11, pp. 2683–2703, 2011.

[52] Z.-Y. Wang, J. Qiu, and F.-F. Li, "Hybrid models combining emd/eemd and arima for long-term streamflow forecasting," *Water*, vol. 10, no. 7, p. 853, 2018.

[53] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[54] S. Kavitha, S. Varuna, and R. Ramya, "A comparative analysis on linear regression and support vector regression," in *2016 online international conference on green engineering and technologies (IC-GET)*. IEEE, 2016, pp. 1–5.

[55] X. Zhao, X. Chen, Y. Xu, D. Xi, Y. Zhang, and X. Zheng, "An emd-based chaotic least squares support vector machine hybrid model for annual runoff forecasting," *Water*, vol. 9, no. 3, p. 153, 2017.

[56] A. Girard, C. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," *Advances in neural information processing systems*, vol. 15, 2002.

[57] W. Yan, H. Qiu, and Y. Xue, "Gaussian process for long-term time-series forecasting," in *2009 international joint conference on neural networks*. IEEE, 2009, pp. 3420–3427.

[58] A. Y. Sun, D. Wang, and X. Xu, "Monthly streamflow forecasting using gaussian process regression," *Journal of Hydrology*, vol. 511, pp. 72–81, 2014.

[59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[60] S. P. Van, H. M. Le, D. V. Thanh, T. D. Dang, H. H. Loc, and D. T. Anh, "Deep learning convolutional neural network in rainfall–runoff modelling," *Journal of Hydroinformatics*, vol. 22, no. 3, pp. 541–561, 2020.

[61] X. Li, Z. Du, and G. Song, "A method of rainfall runoff forecasting based on deep convolution neural networks," in *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*. IEEE, 2018, pp. 304–310.

[62] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[63] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, 2014.

[64] L. Yan, J. Feng, T. Hang, and Y. Zhu, "Flow interval prediction based on deep residual network and lower and upper boundary estimation method," *Applied Soft Computing*, vol. 104, p. 107228, 2021.

[65] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE transactions on neural networks*, vol. 22, no. 3, pp. 337–346, 2010.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[67] D. Liu, W. Jiang, L. Mu, and S. Wang, "Streamflow prediction using deep learning neural network: case study of yangtze river," *IEEE Access*, vol. 8, pp. 90 069–90 086, 2020.

[68] K. Lin, S. Sheng, Y. Zhou, F. Liu, Z. Li, H. Chen, C.-Y. Xu, J. Chen, and S. Guo, "The exploration of a temporal convolutional network combined with encoder-decoder framework for runoff forecasting," *Hydrology Research*, vol. 51, no. 5, pp. 1136–1149, 2020.

[69] A. Gu, C. Gulcehre, T. Paine, M. Hoffman, and R. Pascanu, "Improving the gating mechanism of recurrent neural networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3800–3809.

[70] P. Li, J. Zhang, and P. Krebs, "Prediction of flow based on a cnn-lstm combined deep learning approach," *Water*, vol. 14, no. 6, p. 993, 2022.

[71] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 62–74, 2016.

[72] Y. Liu, T. Zhang, A. Kang, J. Li, and X. Lei, "Research on runoff simulations using deep-learning methods," *Sustainability*, vol. 13, no. 3, p. 1336, 2021.

[73] Z.-M. Wang, O. Batelaan, and F. De Smedt, "A distributed model for water and energy transfer between soil, plants and atmosphere (wetspa)," *Physics and Chemistry of the Earth*, vol. 21, no. 3, pp. 189–193, 1996.

[74] X.-y. Bi, B. Li, W.-l. Lu, and X.-z. Zhou, "Daily runoff forecasting based on data-augmented neural network model," *Journal of Hydroinformatics*, vol. 22, no. 4, pp. 900–915, 2020.

[75] N. M. Noor, M. M. Al Bakri Abdullah, A. S. Yahaya, and N. A. Ramli, "Comparison of linear interpolation method and mean method to replace the missing values in environmental data set," in *Materials Science Forum*, vol. 803. Trans Tech Publ, 2015, pp. 278–281.

[76] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1127–1137.

[77] Z. Xiang, J. Yan, and I. Demir, "A rainfall-runoff model with lstm-based sequence-to-sequence learning," *Water resources research*, vol. 56, no. 1, p. e2019WR025326, 2020.

[78] H. Yin, X. Zhang, F. Wang, Y. Zhang, R. Xia, and J. Jin, "Rainfall-runoff modeling using lstm-based multi-state-vector sequence-to-sequence model," *Journal of Hydrology*, vol. 598, p. 126378, 2021.

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[80] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[81] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled, "Overview of the transformer-based models for nlp tasks," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020, pp. 179–183.

[82] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "Multi-head attention: Collaborate instead of concatenate," *arXiv preprint arXiv:2006.16362*, 2020.

[83] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[84] H. Yin, Z. Guo, X. Zhang, J. Chen, and Y. Zhang, "Rr-former: Rainfall-runoff modeling based on transformer," *Journal of Hydrology*, vol. 609, p. 127781, 2022.

[85] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *arXiv preprint arXiv:2106.04554*, 2021.

[86] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[87] P. Bühlmann, "Bagging, boosting and ensemble methods," in *Handbook of computational statistics*. Springer, 2012, pp. 985–1022.

[88] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[89] F. Huang, J. Ash, J. Langford, and R. Schapire, "Learning deep resnet blocks sequentially using boosting theory," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2058–2067.

[90] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[91] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[92] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[93] Y.-Y. Chang, F.-Y. Sun, Y.-H. Wu, and S.-D. Lin, "A memory-network based solution for multivariate time-series forecasting," *arXiv preprint arXiv:1809.02105*, 2018.

[94] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *arXiv preprint arXiv:1912.09363*, 2019.

[95] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, p. 108218, 2022.

[96] F. Wang, Q. Ge, Q. Yu, H. Wang, and X. Xu, "Impacts of land-use and land-cover changes on river runoff in yellow river basin for period of 1956–2012," *Chinese Geographical Science*, vol. 27, pp. 13–24, 2017.

[97] Y. Zhou, C. Lai, Z. Wang, X. Chen, Z. Zeng, J. Chen, and X. Bai, "Quantitative evaluation of the impact of climate change and human activity on runoff change in the dongjiang river basin, china," *Water*, vol. 10, no. 5, p. 571, 2018.

[98] C. J. Talbot, E. M. Bennett, K. Cassell, D. M. Hanes, E. C. Minor, H. Paerl, P. A. Raymond, R. Vargas, P. G. Vidon, W. Wollheim *et al.*, "The impact of flooding on aquatic ecosystem services," *Biogeochemistry*, vol. 141, pp. 439–461, 2018.

[99] F. Chiew, S. Zhou, and T. McMahon, "Use of seasonal streamflow forecasts in water resources management," *Journal of Hydrology*, vol. 270, no. 1-2, pp. 135–144, 2003.

[100] S. Gao, S. Zhang, Y. Huang, J. Han, H. Luo, Y. Zhang, and G. Wang, "A new seq2seq architecture for hourly runoff prediction using historical rainfall and runoff as input," *Journal of Hydrology*, vol. 612, p. 128099, 2022.

[101] G. Ayzel, "Does deep learning advance hourly runoff predictions," in *Proceedings of the V international conference information technologies and high-performance computing (ITHPC-2019), Khabarovsk, Russia*, 2019, pp. 16–19.

[102] F. Pappenberger, H. L. Cloke, D. J. Parker, F. Wetterhall, D. S. Richardson, and J. Thielen, "The monetary benefit of early flood warnings in europe," *Environmental Science & Policy*, vol. 51, pp. 278–291, 2015.

[103] Y. Zhang, Z. Zhou, J. Van Griensven Thé, S. X. Yang, and B. Gharabaghi, "Flood forecasting using hybrid lstm and gru models with lag time preprocessing," *Water*, vol. 15, no. 22, p. 3982, 2023.

[104] S. K. Jain, P. Mani, S. K. Jain, P. Prakash, V. P. Singh, D. Tullos, S. Kumar, S. P. Agarwal, and A. P. Dimri, "A brief review of flood forecasting techniques and their applications," *International journal of river basin management*, vol. 16, no. 3, pp. 329–344, 2018.

[105] S. H. Silva and P. Najafirad, "Opportunities and challenges in deep learning adversarial robustness: A survey," *arXiv preprint arXiv:2007.00753*, 2020.

[106] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[107] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[108] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[109] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[110] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[111] X. Cheng, X. Li, J. Yang, and Y. Tai, "Sesr: Single image super resolution with recursive squeeze and excitation networks," in *2018 24th International conference on pattern recognition (ICPR)*. IEEE, 2018, pp. 147–152.

[112] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[113] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE / CVF conference on computer vision and pattern recognition*, 2020, pp. 11 534–11 542.

[114] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[115] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[116] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[117] J. E. Nash and J. V. Sutcliffe, "River flow forecasting through conceptual models part i‚Äîa discussion of principles," *Journal of hydrology*, vol. 10, no. 3, pp. 282–290, 1970.

[118] A. Ritter and R. Muñoz-Carpena, "Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments," *Journal of Hydrology*, vol. 480, pp. 33–45, 2013.

[119] Columbia River DART, "Hourly water quality measurements," http://www.cbr.washington.edu/dart/query/wqm_hourly, 2021, columbia Basin Research, University of Washington.

[120] J. C. Bennett, D. E. Robertson, P. G. Ward, H. P. Hapuarachchi, and Q. J. Wang, "Calibrating hourly rainfall-runoff models with daily forcings for streamflow forecasting applications in meso-scale catchments," *Environmental Modelling & Software*, vol. 76, pp. 20–36, 2016.

[121] H. Badrzadeh, R. Sarukkalige, and A. Jayawardena, "Hourly runoff forecasting for flood risk management: Application of various computational intelligence models," *Journal of Hydrology*, vol. 529, pp. 1633–1643, 2015.

[122] D. Masseroni, A. Cislaghi, S. Camici, C. Massari, and L. Brocca, "A reliable rainfall–runoff model for flood forecasting: Review and application to a semi-urbanized watershed at high flood risk in italy," *Hydrology Research*, vol. 48, no. 3, pp. 726–740, 2017.

[123] X. Zhang, Y. Peng, W. Xu, and B. Wang, "An optimal operation model for hydropower stations considering inflow forecasts with different lead-times," *Water resources management*, vol. 33, pp. 173–188, 2019.

[124] E. Ogliari, A. Nespoli, M. Mussetta, S. Pretto, A. Zimbardo, N. Bonfanti, and M. Aufiero, "A hybrid method for the run-of-the-river hydroelectric power plant energy forecast: hype hydrological model and neural network," *Forecasting*, vol. 2, no. 4, pp. 410–428, 2020.

[125] H. Han and R. R. Morrison, "Improved runoff forecasting performance through error predictions using a deep-learning approach," *Journal of Hydrology*, vol. 608, p. 127653, 2022.

[126] H. Sanikhani, M. R. Nikpour, and F. Jamshidi, "Advanced framework for predicting rainfall-runoff: Comparative evaluation of ai models for enhanced forecasting accuracy," *Water Resources Management*, pp. 1–22, 2025.

[127] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[128] D. Soydaner, "Attention mechanism in neural networks: where it comes and where it goes," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13 371–13 385, 2022.

[129] D. Han, P. Liu, K. Xie, H. Li, Q. Xia, Q. Cheng, Y. Wang, Z. Yang, Y. Zhang, and J. Xia, "An attention-based lstm model for long-term runoff forecasting and factor recognition," *Environmental Research Letters*, vol. 18, no. 2, p. 024004, 2023.

[130] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[131] Q.-s. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[132] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[133] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[134] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2018.

[135] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[136] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[137] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[138] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[139] D. Machiwal and M. K. Jha, "Time series analysis of hydrologic data for water resources planning and management: a review," *Journal of Hydrology and Hydromechanics*, vol. 54, no. 3, pp. 237–257, 2006.

[140] L. Yunpeng, H. Di, B. Junpeng, and Q. Yong, "Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network," in *2017 14th web information systems and applications conference (WISA)*. IEEE, 2017, pp. 305–310.

[141] T. Lees, S. Reece, F. Kratzert, D. Klotz, M. Gauch, J. De Bruijn, R. Kumar Sahu, P. Greve, L. Slater, and S. Dadson, "Hydrological concept formation inside long short-term memory (lstm) networks," *Hydrology and Earth System Sciences Discussions*, vol. 2021, pp. 1–37, 2021.

[142] H. Deng, W. Chen, and G. Huang, “Deep insight into daily runoff forecasting based on a cnn-lstm model,” *Natural Hazards*, vol. 113, no. 3, pp. 1675–1696, 2022.

[143] H. Wu, Q. Yang, J. Liu, and G. Wang, “A spatiotemporal deep fusion model for merging satellite and gauge precipitation in china,” *Journal of Hydrology*, vol. 584, p. 124664, 2020.

[144] J. Singh and R. Banerjee, “A study on single and multi-layer perceptron neural network,” in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019, pp. 35–40.

[145] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, “Deep learning for time series forecasting: a survey,” *Big data*, vol. 9, no. 1, pp. 3–21, 2021.

[146] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, A. Muneer *et al.*, “Lstm inefficiency in long-term dependencies regression problems,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 30, no. 3, pp. 16–31, 2023.

[147] R. Rick and L. Berton, “Energy forecasting model based on cnn-lstm-ae for many time series with unequal lengths,” *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104998, 2022.

[148] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[149] F. Furizal, A. B. Fawait, H. Maghfiroh, A. Ma,Äôarif, A. A. Firdaus, and I. Suwarno, “Long short-term memory vs gated recurrent unit: A literature review on the performance of deep learning methods in temperature time series forecasting,” *International Journal of Robotics and Control Systems*, vol. 4, no. 3, pp. 1506–1526, 2024.

[150] S. Gao, Y. Huang, S. Zhang, J. Han, G. Wang, M. Zhang, and Q. Lin, “Short-term runoff prediction with gru and lstm networks without requiring time step optimization during sample generation,” *Journal of Hydrology*, vol. 589, p. 125188, 2020.

[151] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[152] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*. IEEE, 2019, pp. 3285–3292.

[153] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018.

[154] J. Zhao, F. Huang, J. Lv, Y. Duan, Z. Qin, G. Li, and G. Tian, "Do rnn and lstm have long memory?" in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 365–11 375.

[155] S. Gopali, F. Abri, S. Siami-Namini, and A. S. Namin, "A comparison of tcn and lstm models in detecting anomalies in time series data," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 2415–2420.

[156] M. Cheng, F. Fang, T. Kinouchi, I. Navon, and C. Pain, "Long lead-time daily and monthly streamflow forecasting using machine learning methods," *Journal of Hydrology*, vol. 590, p. 125376, 2020.

[157] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[158] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[159] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[160] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[161] B. Peng, S. Narayanan, and C. Papadimitriou, "On limitations of the transformer architecture," in *First Conference on Language Modeling*, 2024.

[162] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[163] H. Qu, L. Ning, R. An, W. Fan, T. Derr, H. Liu, X. Xu, and Q. Li, "A survey of mamba," *arXiv preprint arXiv:2408.01129*, 2024.

[164] S. Das, R. Sen, and S. Devendiran, "Mamba models a possible replacement for transformers?" *Proceedings of the 23nd*, 2024.

[165] X. Wang, S. Wang, Y. Ding, Y. Li, W. Wu, Y. Rong, W. Kong, J. Huang, S. Li, H. Yang *et al.*, "State space model for new-generation network alternative to transformers: A survey," *arXiv preprint arXiv:2404.09516*, 2024.

[166] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, vol. 33, pp. 1474–1487, 2020.

[167] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," *arXiv preprint arXiv:2405.21060*, 2024.

[168] A. Liang, X. Jiang, Y. Sun, X. Shi, and K. Li, "Bi-mamba+: Bidirectional mamba for time series forecasting," *arXiv preprint arXiv:2404.15772*, 2024.

[169] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3279–3298, 2021.

[170] S. Ecrepont, C. Cudennec, F. Anctil, and A. Jaffrézic, "Pub in québec: A robust geomorphology-based deconvolution-reconvolution framework for the spatial transposition of hydrographs," *Journal of Hydrology*, vol. 570, pp. 378–392, 2019.

[171] R. Waleffe, W. Byeon, D. Riach, B. Norick, V. Korthikanti, T. Dao, A. Gu, A. Hatamizadeh, S. Singh, D. Narayanan *et al.*, "An empirical study of mamba-based language models, 2024," *URL https://arxiv. org/abs/2406.07887*.