Journal of Water Resources Research 水资源研究, 2026, 15(3), 306-325
Published Online June 2026 in Hans. https://www.hanspub.org/journal/jwrr
https://doi.org/10.12677/jwrr.2026.153034

Hans汉斯

基于ONNX与测算一体的端侧轻量化洪水预报

程　帅[1]，张佳宁[2]，朱一松[1*]，王增凯[3]，郑晓燕[1]，刘维高[1]，项伍林[1]

[1]武汉大水云科技有限公司，湖北　武汉
[2]山东省水文中心，山东　济南
[3]烟台市水文中心，山东　烟台

收稿日期：2026年4月17日；录用日期：2026年5月6日；发布日期：2026年6月29日

摘　要

洪涝灾害频发且破坏性强，对洪水预报的时效性、经济性与场景适应性提出更高要求。针对传统洪水预报模型计算量大、云端部署存在传输延迟高、运维成本高、偏远地区适配性差等问题，本研究提出一种基于开放神经网络交换(ONNX)的端侧轻量化LSTM测算一体的洪水预报方法。以山东省内6个AiFlow视觉测流监测站点的流量数据为研究基础，通过Yeo Johnson转换将偏态分布的流量数据转换为近似正态分布，结合Z-score标准差标准化构建时序训练样本；采用双层堆叠LSTM模型捕捉水文数据的时序依赖关系，基于ONNX对模型开展计算图优化、算子融合与量化处理，实现模型的端侧轻量化部署与河道流量实时滚动预报。实验结果表明，该方法在6个监测站点均表现出可靠的预报精度，1、3步滚动预报纳什系数(NSE)均高于0.91，平均绝对百分比误差(MAPE)均小于10%，满足短期洪水预报需求；经ONNX轻量化处理后，较原始PyTorch格式模型，模型文件体积减少了67%，可执行文件体积减少了69%，端侧推理冷启动耗时降低超99%，6个站点的平均推理耗时降低7%~31%，模型预报NSE无变化；模型加载内存增量从111.97 MB降低至0.54 MB，完全适配端侧设备的资源约束。本方法实现了“数据采集、本地推理、预报输出”的全流程闭环，显著降低了洪水预报的运维成本和预报延迟，为中小河流、偏远地区的测算一体化洪水预报提供了低成本、轻量化的技术方案。

关键词

测算一体，洪水预报，ONNX，LSTM

ONNX-Based Lightweight On-Device Flood Forecasting with Integrated Measurement and Computation

Shuai Cheng[1], Jianing Zhang[2], Yisong Zhu[1*], Zengkai Wang[3], Xiaoyan Zheng[1], Weigao Liu[1], Wulin Xiang[1]

作者简介：朱一松(1992-)，男，工程师，硕士，主要从事智慧水利工作。
[*]通讯作者 Email: songyi@whu.edu.cn

文章引用: 程帅, 张佳宁, 朱一松, 王增凯, 郑晓燕, 刘维高, 项伍林. 基于 ONNX 与测算一体的端侧轻量化洪水预报[J]. 水资源研究, 2026, 15(3): 306-325. DOI: 10.12677/jwrr.2026.153034

[1]Wuhan Dashuiyun Technology Co., Ltd., Wuhan Hubei
[2]Shandong Provincial Hydrological Center, Jinan Shandong
[3]Yantai City Hydrological Center, Yantai Shandong

Received: April 17, 2026; accepted: May 6, 2026; published: June 29, 2026

Abstract

Frequent and highly destructive flood disasters have imposed increasingly stringent requirements on the timeliness, cost-effectiveness, and scenario adaptability of flood forecasting. To address the key limitations of traditional flood forecasting models (namely excessive computational overhead) and the inherent drawbacks of cloud-based deployment (including high transmission latency, elevated operation and maintenance (O&M) costs, and poor adaptability to remote areas), this study proposes a lightweight on-device LSTM (Long Short-Term Memory) flood forecasting method with integrated measurement and computation based on the Open Neural Network Exchange (ONNX). Taking the discharge data from 6 AiFlow visual flow measurement monitoring stations in Shandong Province as the research basis, the skewed discharge data are transformed into an approximately normal distribution via the Yeo-Johnson transformation, and time-series training samples are constructed in conjunction with Z-score standardization. A two-layer stacked LSTM model is built to capture the temporal dependencies of hydrological series, and ONNX-based computation graph optimization, operator fusion, and quantization processing are implemented on the trained model, enabling lightweight on-device deployment of the model and real-time rolling forecasting of river channel discharge. The experimental results demonstrate that the proposed method achieves reliable forecasting performance across all 6 monitoring stations. For 1-step and 3-step ahead rolling forecasting, the Nash-Sutcliffe Efficiency (NSE) values are all above 0.91, and the Mean Absolute Percentage Error (MAPE) values are all below 10%, fully satisfying the operational requirements of short-term flood forecasting. After ONNX-enabled lightweight processing, the model code is reduced by 67%, the executable file size is reduced by 69%. The cold start latency of on-device inference is cut by over 99%, and the average inference time across the 6 stations is reduced by 7% to 31%, with no degradation in forecasting NSE. The memory increment required for model loading is reduced from 111.97 MB to 0.54 MB, which fully complies with the resource constraints of on-device edge terminals. The proposed method establishes a full-process closed loop covering "data acquisition, local inference, and forecasting output", thus significantly lowers the O&M costs and forecasting latency of flood forecasting, and provides a low-cost, lightweight technical solution for integrated measurement-computation flood forecasting in medium and small rivers and remote areas.

Keywords

Integrated Measurement and Computation, Flood Forecasting, ONNX, LSTM

1. 引言

洪涝灾害是全球范围内影响最广泛、损失最严重的自然灾害之一，其突发性和强破坏性往往导致重大人员伤亡与经济损失[1]。及时、准确地进行洪水监测与预报是防灾减灾工作的核心，可以为应急调度、人员转移争取关键时间[2]。目前，主流的洪水预报模型主要分为两类：一类是基于物理机制的数值模型如 MIKE [3]、

HEC-RAS [4]、TUFLOW [5]等，这类模型通过刻画水文循环的物理过程实现预报，但需进行复杂的参数率定，且存在分布计算导致计算量大，对硬件算力要求严苛；另一类是基于数据驱动的深度学习模型如 LSTM [6]、Transformer [7]、Informer [8]、GRU [9]等，这类模型无需依赖物理机制，而是通过挖掘时序数据中的潜在关联实现预报，在水文预报中表现出较高的精度和预报效率，但普遍存在参数量大、计算复杂度高的问题；例如 Transformer 模型的多头注意力机制涉及大量矩阵运算[10]，Informer 虽优化了长时序处理效率但仍需较高的内存支持[11]，LSTM 模型的门控结构也导致其计算开销显著高于基础 RNN 模型[12]，对硬件算力同样有着较高的需求。

从部署模式来看，无论是数值模型还是深度学习模型，大多采用“云端集中部署”的方式，预报流程需经过“现场监测设备采集数据、数据回传至云端服务器、云端模型运算、预报结果分发至终端”等多个环节[13]。这种模式存在三大限制：一是时效性不足，数据传输、云端排队计算等过程会产生不可避免的延迟，对于汇流速度快的中小流域，延迟可能导致预报失去实际意义；二是运维成本高，需投入大量资源建设服务器集群、维护网络传输链路，且海量水文数据的重复存储与传输会进一步增加成本[14]；三是端侧场景适应性差，在偏远山区、网络信号薄弱区域，数据传输易中断，导致预报服务中断。

为提升洪水预报实时性和全天候适应性，许多研究聚焦于洪水监测与预报的边缘计算。例如，Lee 等提出融合 RGB 与 LWIR 相机的 DNN 监测系统，通过网络瘦身优化 DeepLabV3+，在 QualcommQCS610 边缘设备实现实时水体分割[15]，Thirugnanasammandamoorthi P 等提出了 FloodNet-Lite 框架，基于优化 UNet 与 MobileNetV3，结合量化剪枝等技术，在 Jetson Nano 平台上实现 14.3FPS 的洪水制图推理[16]，Samikwa E 采用 LSTM 模型结合物联网技术(loT)，在 Raspberry Pi 边缘设备上实现了基于降雨和水位时序数据实现短期洪水预报任务[17]；上述研究均验证了现阶段的端侧设备已经具备支撑轻量化深度学习模型的运行能力，但现有研究普遍将监测任务与预报任务拆分执行，依赖物联网、蓝牙、网络链路等通信技术实现跨设备数据传输。当发生断电、断网等极端突发状况时，这种模式极易出现全面中断风险，造成关键监测数据丢失、预报流程被迫中断，还可能因通信延迟或彻底中断错过黄金预警窗口期，甚至出现“监测端已采集数据却无法上传、预报端已生成结果却无法下发”的脱节问题，严重削弱洪水预警的时效性与可靠性。若能将预报模型直接部署于 AiFlow 等[18]具备数据采集与基础计算能力的监测工作站上，实现基于端侧设备的河道流量“测算一体”，即可实现监测数据的本地处理与预报结果实时输出，从根源上规避云端部署依赖、跨设备通信中断等潜在风险。

开放神经网络交换(ONNX)作为开源的神经网络模型格式标准，为解决端侧部署的轻量化问题提供了有效途径[19]。其核心价值在于打破不同深度学习框架的格式壁垒，实现模型的跨框架转换与跨平台部署，且配套的 ONNX Runtime 推理引擎能够通过计算图优化、算子融合、量化等技术，在不显著损失精度的前提下，显著降低模型的计算开销与存储需求。本研究提出了一种基于开放神经网络交换(ONNX)的端侧轻量化 LSTM 测算一体洪水预报方法：以 LSTM 模型为核心预报模型，通过 ONNX 进行模型轻量化处理，将整个预报流程部署于端侧监测设备，实现“数据采集–本地预处理–预报输出”的全闭环运行，以山东省境内 6 个 AiFlow 监测站点的流量数据为基础，验证该方法的预报精度、轻量化效果与端侧部署性能，为洪水预报的端侧化、实时化、低成本化提供技术支撑。

2. 研究站点及数据来源

选取山东省境内 6 个 AiFlow 视觉测流监测站点为研究对象，AiFlow 视觉测流采样时空图像测速法(STIV)抓取河流表面的涟漪、波纹等水流特征变化来反映河流表面流速大小[20]。6 个站点包含了山东省内大小河流，涵盖了自然流域产流、支流汇入、上游水库调蓄等多种情形，6 个监测站点 AiFlow 视觉测流仪监测的历史流量数据，各站点的数据长度、监测频率、历史最大流量、历史最小流量等基本信息如表 1 所示，历史流量序列过

程如图 1 所示。

3. 研究方法

本研究提出的基于开放神经网络交换(ONNX)的端侧轻量化测算一体洪水预报方法，在训练阶段包含“数据收集、数据处理、模型训练”三个模块，在端侧设备预报阶段包含“监测数据收集、数据处理、模型预报”三个部分，通过 ONNX 将训练完成的洪水预报模型进行轻量化处理并部署到端侧设备上，实现数据监测与实时预报一体进行，具体流程如图 2 所示。

表 1. 研究站点及监测数据基本情况表

站点名称	数据长度	监测频率/min	最大流量/$m^3 \cdot s^{-1}$	最小流量/$m^3 \cdot s^{-1}$	均值/$m^3 \cdot s^{-1}$
S1	17,653	60	505.61	0.00	10.99
S2	47,563	15	36.59	0.00	5.68
S3	26,564	15	119.44	0.00	27.07
S4	30,779	15	51.69	0.00	12.40
S5	26,232	15	134.53	0.10	5.65
S6	21,792	60	84.35	0.10	1.95

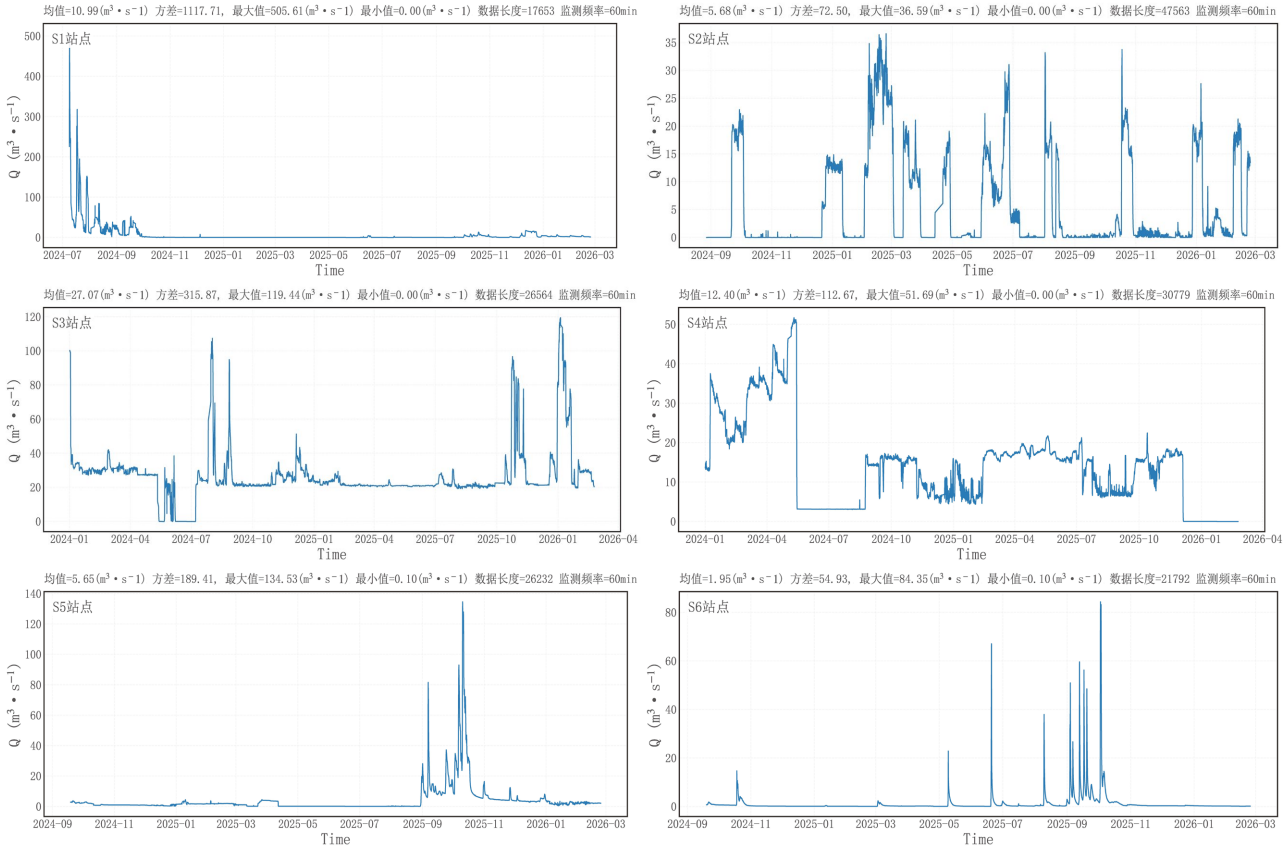


图 1. 研究站点历史流量过程

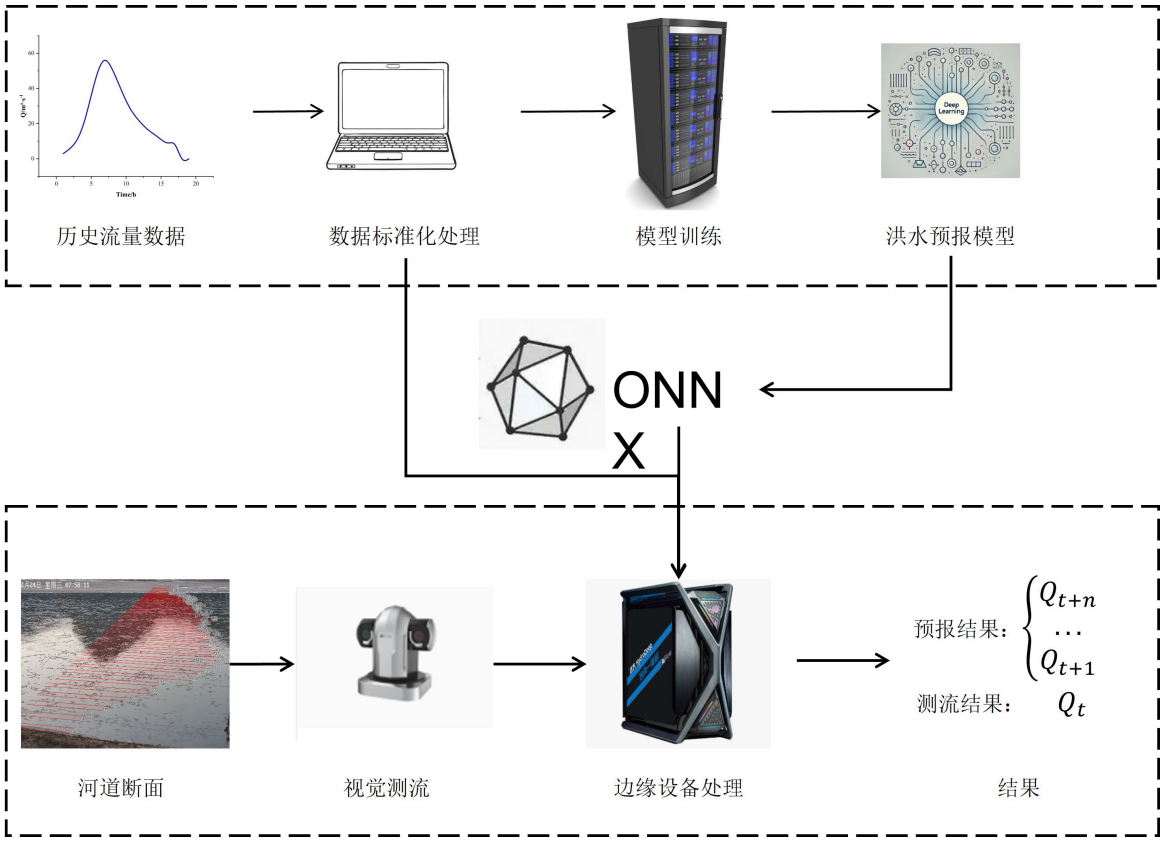


图 2. 基于 ONNX 的端侧轻量化测算一体洪水预报流程

3.1. 数据处理

为保障模型训练效果与端侧预报结果的实时性与准确性，数据预处理环节采用轻量化设计，主要包括空缺值和异常值处理、滑动窗口处理、Yeo Johnson 转换、标准差标准化(Z-score)四个步骤。

3.1.1. 空缺值和异常值处理

在模型训练阶段，对原始数据中的空缺值，根据空缺长度进行分类处理；当空缺长度小于等于 3 个时间步时，为保证训练数据的可用数量，采用线性插值法进行补充，即利用空缺时刻前一有效时刻的流量值和空缺时刻后一有效时刻的流量值通过线性插值填充空缺位置；当空缺值长度超过 3 个时间步时，直接剔除这部分数据，避免长距离插值破坏实测数据的分布规律。对于异常值只进行简单的非负筛选，对于异常长度不超过 3 个时间步的异常值同样采用线性插值方法进行处理，超过 3 个时间步的异常值也直接剔除。

在端侧推理阶段，对由各种原因导致的空缺值和异常值，将使用空缺值(异常值)时刻前的有效数据作为模型输入，推算空缺值(异常值)时刻的流量，并将其作为输入进行未来洪水预报。

3.1.2. 滑动窗口与滚动预报

在本研究中，选择Q_{t-4}、Q_{t-3}、Q_{t-2}、Q_{t-1}、Q_t五个时刻的流量数据作为输入，模型输出为Q_{t+1}，采用固定长度滑动窗口法构建训练样本，即窗口沿时间轴逐时刻滑动，生成足量训练样本进行模型训练。

在端侧设备进行洪水预报时，模型会进行滚动预报，窗口更新采用“实时数据补充”策略：当模型推理出 $t+1$ 时刻的流量Q_{t+1}后，直接将其补充至输入窗口末端，同时剔除窗口最前端的旧数据Q_{t-4}，形成新的输入窗口Q_{t-3}、Q_{t-2}、Q_{t-1}、Q_t、Q_{t+1}，用于预报下一时刻流量Q_{t+2}，直到预报出设定的最大推理步长 m 对应的 $t+m$ 时

刻的流量 Q_{t+m}；通过这一方法，模型可以灵活设定预报步长，大幅提高模型的应用范围，在预报时无需重新构建全量样本，仅需增量更新窗口数据，大幅降低端侧实时预报的计算开销。

3.1.3. 数据转换与标准化处理

根据图 3 可知，6 个 AiFlow 监测站点的历史监测流量呈现出显著的偏态分布情况且流量序列方差较大，流量波动较为剧烈；直接以原始流量数据作为模型输入会导致模型训练时出现梯度消失、收敛缓慢等问题。本研究先使用 Yeo Johnson [21]方法将原始流量序列转换为近似正态分布，再采用 Z-score 标准化(标准差标准化)对数据进行标准化处理，得到服从均值为 0、方差为 1 的正态分布数据，消除偏态分布和数值波动对模型的影响，提升模型训练稳定性，其计算公式如下：

Yeo Johnson 转换：

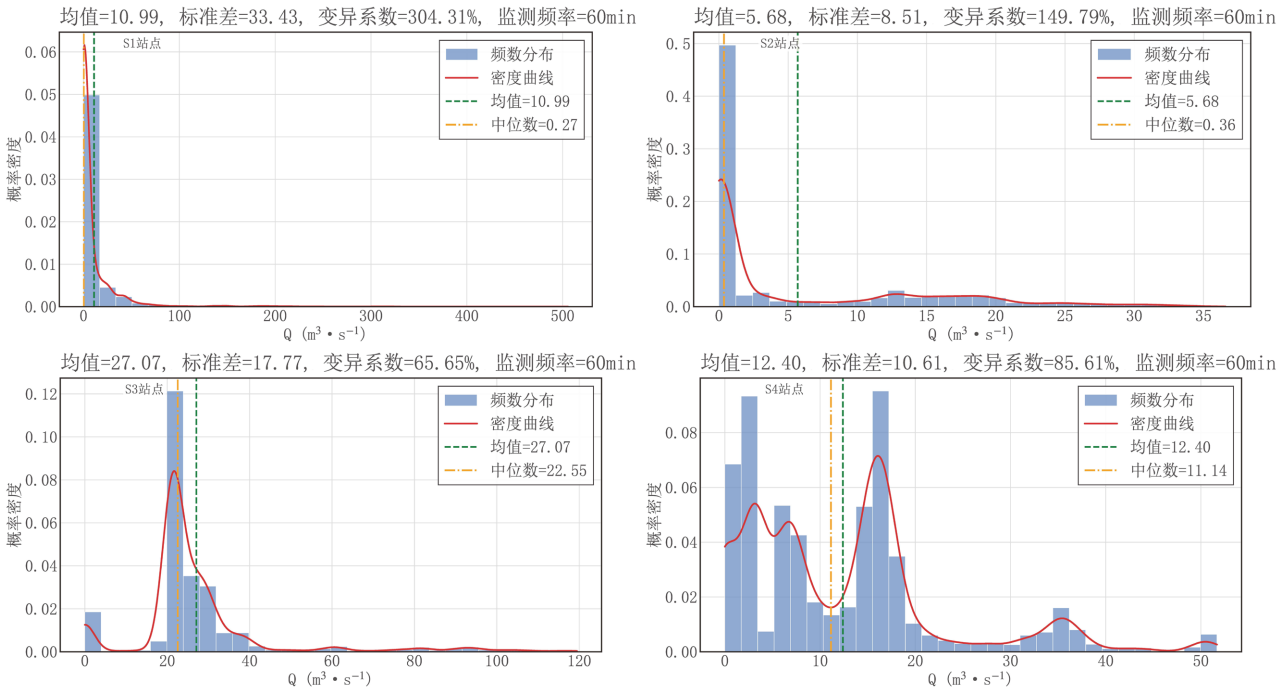
$$T_\lambda(y)=\begin{cases}\dfrac{(y+1)^{\lambda}-1}{\lambda}, & y\geq 0,\lambda\neq 0\\ \lg(y+1), & y\geq 0,\lambda=0\\ -\dfrac{(1-y)^{2-\lambda}-1}{2-\lambda}, & y<0,\lambda\neq 2\\ -\lg(1-y), & y<0,\lambda=2\end{cases} \tag{1}$$

Z-score 标准化：

$$Z=\frac{Q-\mu}{\sigma} \tag{2}$$

式中，Q 为原始流量数据，μ 为训练集的样本均值，σ 为训练集的样本标准差，Z 为标准化后的输出数据。

为避免数据泄露，数据处理时需严格遵循“训练集单独标准化，测试集复用训练集参数”的原则；利用训练集数据计算 Yeo Johnson 转换和 Z-score 标准化的参数并保存至文件，在端侧推理时直接加载文件，对实时监测数据进行转换与标准化处理。



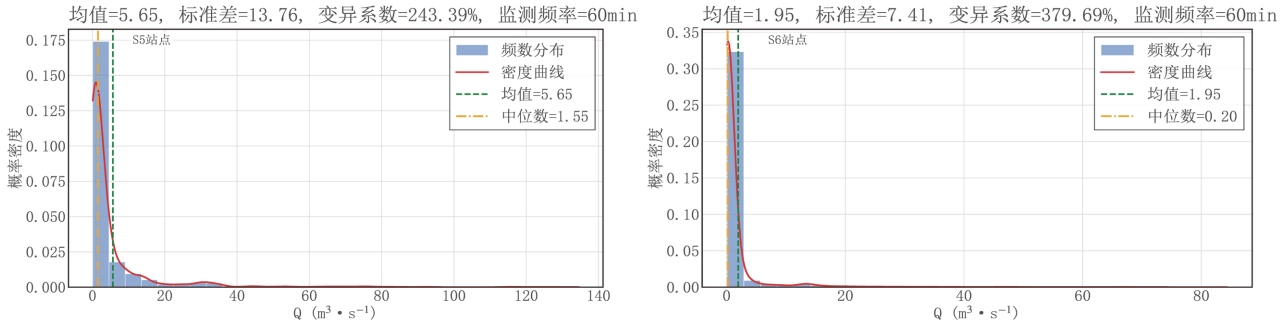


图 3. 各站点流量数据分布情况

3.2. LSTM 洪水预报模型

LSTM 通过其独特的门控结构(遗忘门、输入门、输出门)实现对长时序数据的有效处理，精准捕捉时序数据间的动态关联。LSTM 模型核心数学公式如下：

遗忘门：

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{3}$$

输入门：

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{4}$$

候选细胞状态：

$$C_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{5}$$

细胞状态更新：

$$C_t = f_t \odot C_t - 1 + i_t \odot C_t \tag{6}$$

输出门：

$$O_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{7}$$

隐藏状态更新：

$$h_t = O_t \odot \tanh\left(C_t\right) \tag{8}$$

式中，W_f、W_i、W_c、W_o 分别为遗忘门、输入门、候选细胞状态、输出门的权重矩阵；b_f、b_i、b_c、b_o 分别为对应的偏置向量；h_{t-1} 为上一时刻的隐藏状态；x_t 为当前时刻的输入数据；σ 为 sigmoid 激活函数；tanh 为双曲正切激活函数。

考虑到历史流量变化较为剧烈且流量极大值与极小值差距较大，为更好地提取时序关联特征，本研究分别使用 1 层 LSTM、2 层 LSTM 堆叠和 3 层 LSTM 堆叠的方式构建了洪水预报模型，并在 6 个研究站点进行了初步实验，试验结果如表 2 所示，不同 LSTM 层数的洪水预报模型单步预测结果如图 4 所示。

表 2. 各站点不同层数 LSTM 模型单步预报结果统计指标

站点名称	LSTM 层	NSE	MAE	RMSE	MAPE (%)	实测峰现位置	预报峰现位置	峰现时间差/min
S1	1	0.9645	0.4129	0.8544	6.46	332	333	60
	2	0.9803	0.2909	0.6373	3.86	332	333	60
	3	0.9968	0.1644	0.2551	4.99	332	333	60

续表

S2	1	0.9950	0.2845	0.5380	1.49	637	637	0
	2	0.9873	0.4077	0.8567	1.93	637	637	0
	3	0.9940	0.3121	0.5895	1.67	637	637	0
S3	1	0.6650	7.9025	16.4076	9.21	264	90	2610
	2	0.9689	2.2367	4.9995	3.48	264	267	45
	3	0.5914	9.1769	18.1187	10.90	264	170	1410
S4	1	0.9914	0.2260	0.4340	1.25	1765	1767	30
	2	0.9720	0.3964	0.7809	1.56	1765	1768	45
	3	0.8383	1.1507	1.8771	5.35	1765	1768	45
S5	1	0.9552	0.2134	0.2463	9.77	689	692	45
	2	0.9164	0.2532	0.3365	9.98	689	694	75
	3	0.9341	0.1851	0.2987	8.13	689	694	75
S6	1	0.9979	0.002	0.0037	0.78	975	976	60
	2	0.9946	0.004	0.0059	1.81	975	976	60
	3	0.9975	0.002	0.0040	0.96	975	976	60

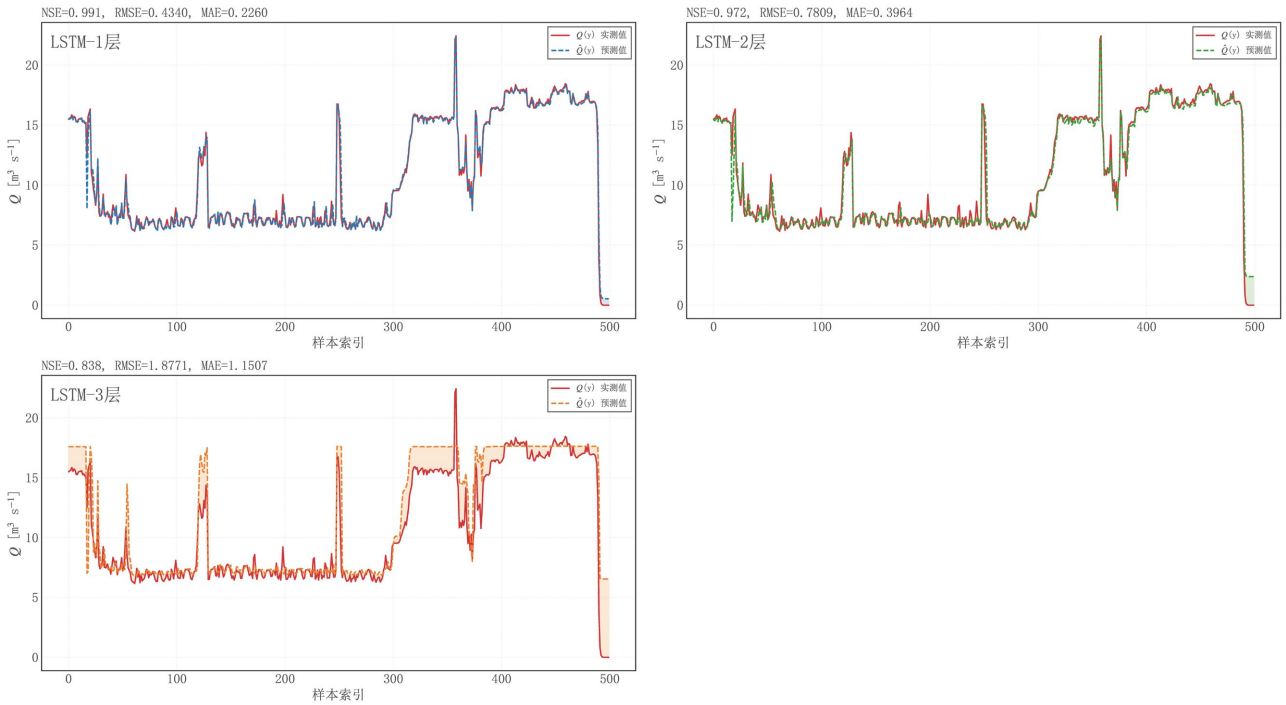


图 4. S4 站点不同 LSTM 层单步预报结果

表 3. 各站点不同层数 LSTM 模型单步预报统计指标分析结果

LSTM 层数	指标	NSE	MAE	RMSE	MAPE	峰现时间差/min
1	均值	0.9282	1.5069	3.0807	4.8267	467.5000
	方差	0.0141	8.1956	35.5891	14.4366	918481.2500
	极差	0.3329	7.9005	16.4039	8.99	2610
	变异系数	13%	190%	194%	79%	205%

续表

2	均值	0.9699	0.5982	1.2695	3.7700	47.5000
	方差	0.0006	0.5547	2.8648	8.4613	556.2500
	极差	0.0782	2.2327	4.9936	8.42	75
	变异系数	3%	125%	133%	77%	50%
3	均值	0.8920	1.8319	3.5239	5.3333	275.0000
	方差	0.0213	10.9273	42.9679	11.9122	258200.0000
	极差	0.4061	9.1749	18.1147	9.94	1410
	变异系数	16%	180%	186%	65%	185%

根据表 2 和表 3 的统计结果可知，LSTM 层数设定为 2 层时，模型在研究站点上的预报效果相较于 LSTM 层数为 1 和 3 的预报效果更加稳定，整体精度更高；因此，本研究设定 LSTM 层数为 2。模型流程图如图 5 所示。

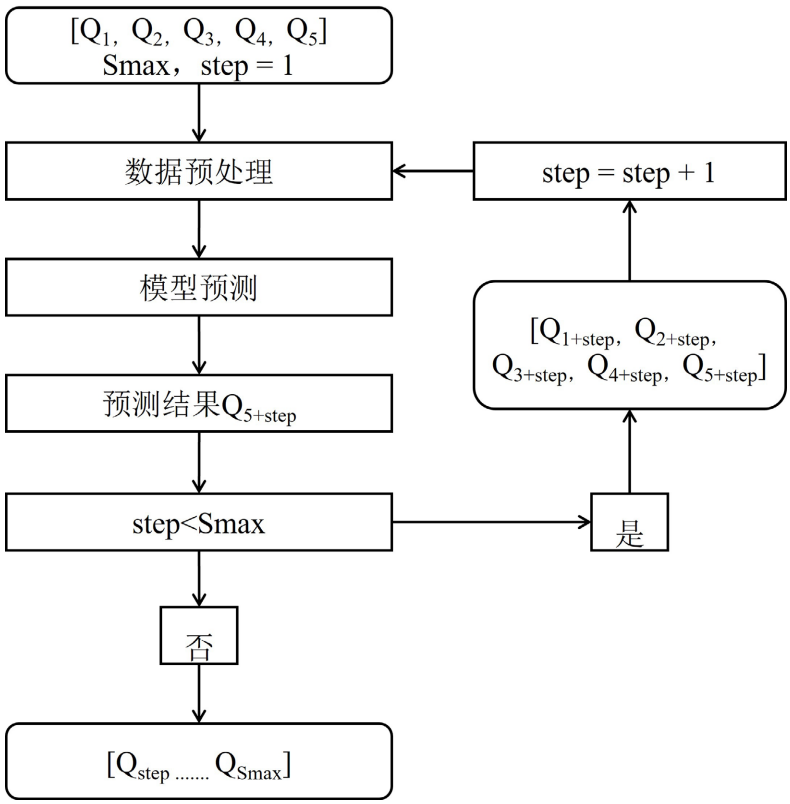


图 5. 基于 LSTM 的洪水预报模型

以 Q_{t-4}、Q_{t-3}、Q_{t-2}、Q_{t-1}、Q_t 五个时刻的流量数据作为模型最外层输入，最终输出为 $t+1$ 时刻的流量 Q_{t+1}，通过 LSTM 模型建立映射关系：

$$Q_{t+1} = f\left(Q_{t-4}, Q_{t-3}, Q_{t-2}, Q_{t-1}, Q_t\right) \tag{9}$$

在推理时，模型将进行如下滚动计算，根据历史监测数据 Q_{t-4}、Q_{t-3}、Q_{t-2}、Q_{t-1}、Q_t 计算得到 $t+1$ 时刻的流量 Q_{t+1}，再将 Q_{t-3}、Q_{t-2}、Q_{t-1}、Q_t、Q_{t+1} 作为输入计算得到 $t+2$ 时刻的流量 Q_{t+2}，一直计算到最大预报

步长 m 对应的 $t+m$ 时刻的流量 Q_{t+m}，计算过程如图 6 所示。

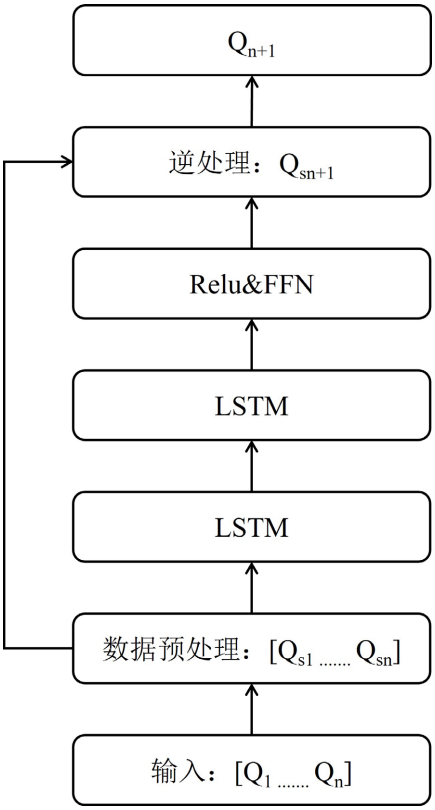


图 6. 洪水预报模型滚动预报计算流程

3.3. ONNX 模型轻量化处理与端侧部署

ONNX (Open Neural Network Exchange)是一种开源的神经网络模型格式标准，其核心目标是打破不同深度学习框架之间的模型格式壁垒，实现模型跨框架训练、跨平台部署的无缝衔接，包含计算图(Graph)、算子(Operator)和张量(Tensor)三大核心组件。ONNX 的模型轻量化处理主要通过 ONNX Runtime 推理引擎实现，主要包括计算图优化、常量折叠、量化优化、硬件适配优化四个部分，具体内容如表 4 所示。与原始 PyTorch 模型相比，经 ONNX 优化后的模型不再依赖庞大的 PyTorch 训练架构，只需要轻量化的 ONNX Runtime 推理引擎即可进行快速推理；优化计算图、常量折叠等优化可以进一步降低模型的推理计算耗时与资源使用情况；由于 ONNX Runtime 设计的目标之一就是实现模型的跨平台部署，因此，ONNX 优化后的模型将支持跨平台部署，不再需要针对不同端侧平台进行模型重构。

表 4. ONNX 优化内容

优化内容	优化方法	作用
计算图优化	消除冗余节点、算子融合	减少计算图的复杂度，降低推理时的算子调用开销
常量折叠	对计算图中由常量组成的表达式进行预计算	将结果存储为常量张量，避免推理时重复计算
量化优化	将模型的浮点精度从 FP32 转换 FP16	在不显著损失精度的前提下，将模型体积压缩 50%，同时降低内存带宽占用和计算开销
硬件适配优化	ONNX Runtime 针对端侧 CPU 优化指令集调用	提升计算效率

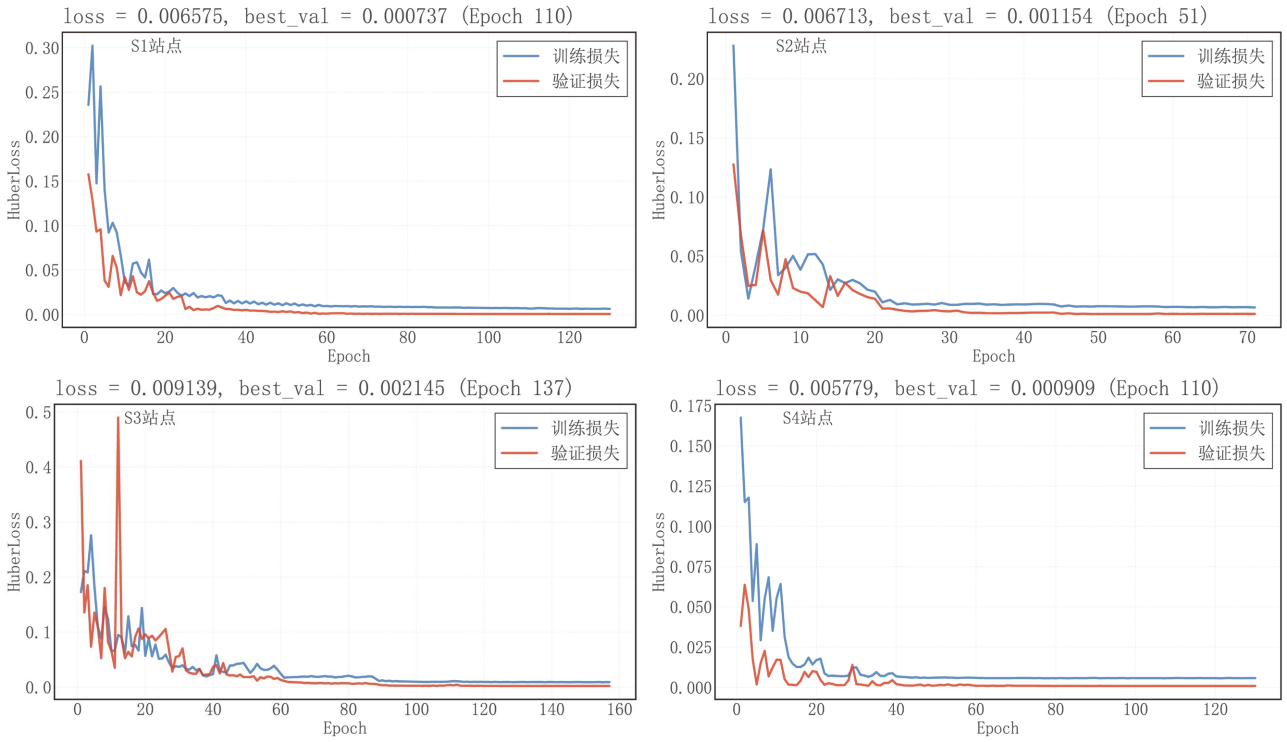
4. 结果分析与讨论

4.1. 模型预报精度验证

对本次研究选择的 6 个站点分别构建对应的 LSTM 洪水预报模型，训练数据将遵循“分层抽样”原则，沿着监测时间按照 8:1:1 的比例将数据集拆分为训练集、验证集、测试集；以 HuberLoss 作为损失函数进行模型训练，初始学习率设定为 1e−4，采用余弦退火策略动态调整模型学习率，模型具体参数如表 5 所示，各站点模型训练结果如表 6 所示。其中，各站点训练损失变化过程如图 7 所示。

表 5. 模型参数

参数名称	值	含义
激活函数	Relu	进行非线性变换
学习率	1e−4	模型参数更新步长，余弦退火策略动态调整
损失函数	HuberLoss	平均绝对误差
早停耐心值	15 步	模型验证损失连续 15 步没有优化就停止训练
最大训练步数	500 步	最大训练步数
LSTM 隐藏层神经元	256	LSTM 内部神经元个数
前馈网络隐藏层神经元个数	512	全连接神经元个数
Batch_size	256	一个训练批次的数据数量
Seq_len	5	输入序列长度
Optimizer	AdamW	优化器
Dropout	0.1	随机丢弃率，防止模型过拟合



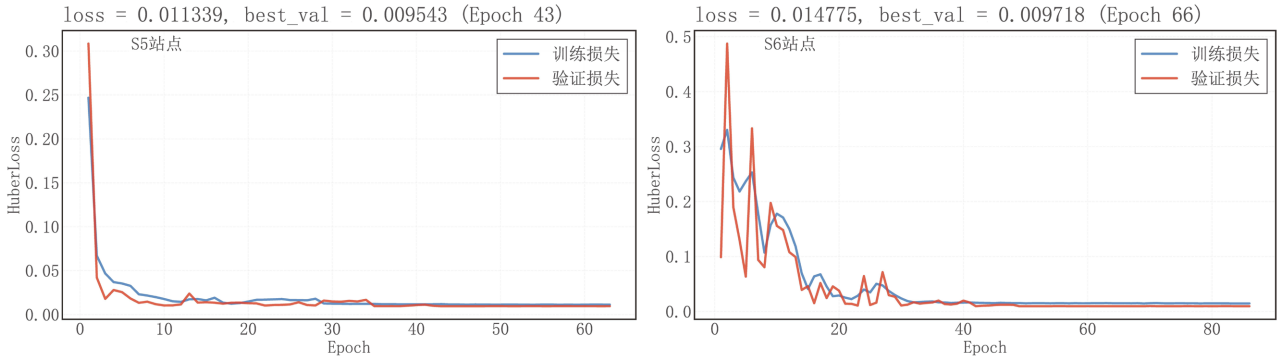


图 7. 各站点模型训练损失收敛过程

表 6. 各站点模型训练情况

序号	站点名称	实际训练步数	最终训练损失	最佳验证损失
1	S1	125	0.006575	0.000737
2	S2	125	0.005779	0.000909
3	S3	78	0.013846	0.013266
4	S4	137	0.006498	0.001230
5	S5	124	0.008324	0.000972
6	S6	74	0.008136	0.000768

为全面分析模型预报精度，将设置 1、3、6、12 四种滚动预报步长，使用纳什系数(NSE)、平均绝对误差(MAE)、均方根误差(RMSE)、平均绝对百分比误差(MAPE)四个指标评价模型在测试集上的表现，部分站点预报过程与实测过程对比结果如图 8，图 9 所示，各站点模型在测试集上的预报精度统计结果如图 10 所示。

$$\mathrm{NSE}=1-\frac{\sum_{i=1}^{n}\left(Q_i^{ture}-Q_i^{pre}\right)^2}{\sum_{i=1}^{n}\left(Q_i^{ture}-Q_{mean}^{ture}\right)^2} \tag{10}$$

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}\left(Q_i^{ture}-Q_i^{pre}\right)^2}{n}} \tag{11}$$

$$\mathrm{MAE}=\frac{\sum_{i=1}^{n}\left|Q_i^{ture}-Q_i^{pre}\right|}{n} \tag{12}$$

$$\mathrm{MAPE}=\frac{\sum_{i=1}^{n}\frac{\left|Q_i^{ture}-Q_i^{pre}\right|}{Q_i^{ture}}}{100\times n}\times 100\% \tag{13}$$

式中，Q_i^{ture} 表示第 i 时刻的实测流量，Q_i^{pre} 表示第 i 时刻的预报流量，Q_{mean}^{ture} 表示实测流量序列的平均值，n 为序列总长度。

根据图 10 的统计结果，LSTM 洪水预报模型的预报精度随着滚动预报步数的增加而逐步下降，表现出正常的误差累计统计规律。具体而言，在滚动预报步数为 1 步和 3 步的情况下，各站点预报结果 NSE 均高于 0.91，所有站点的 MAPE 结果均小于 10%，预报结果具有较高的精度和可信度；滚动预报步数增加到 6 步，各站点预报结果 NSE 均高于 0.86，所有站点的 MAPE 均小于 12%，其中 S2、S3、S4、S6 站点的预报结果 NSE 均高于

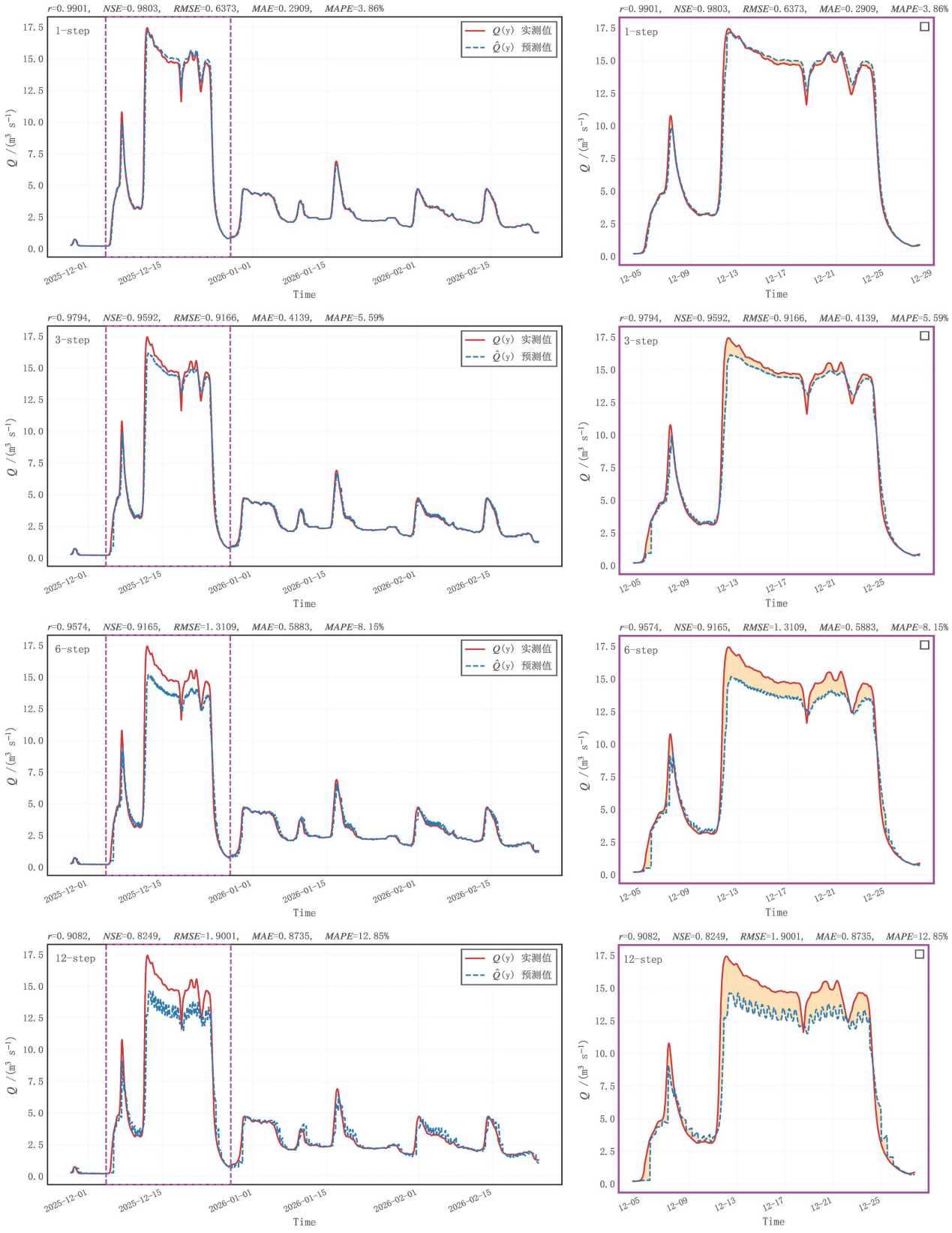


图 8. S1 站点实测结果与预报结果对比图

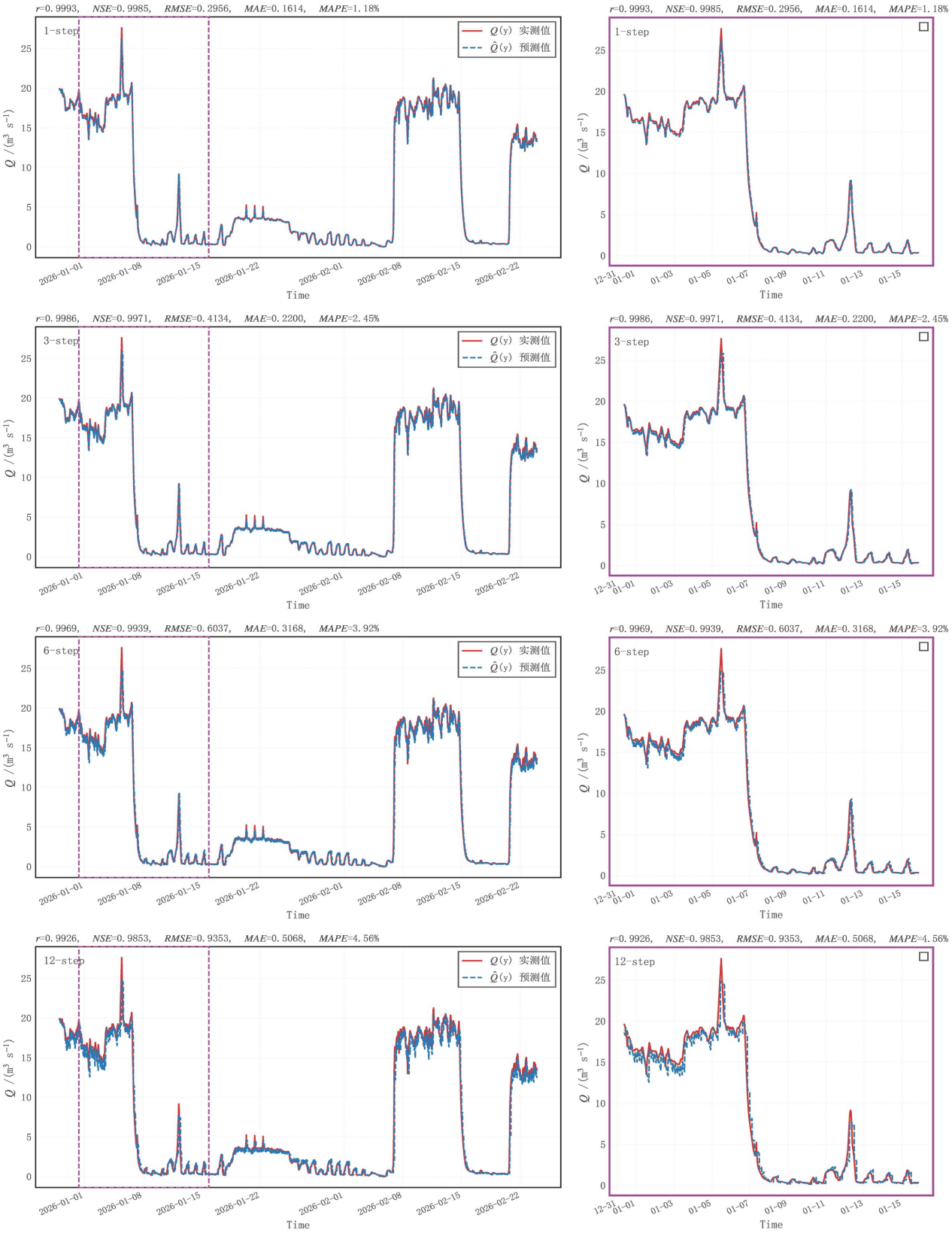


图 9. S2 站点实测结果与预报结果对比图

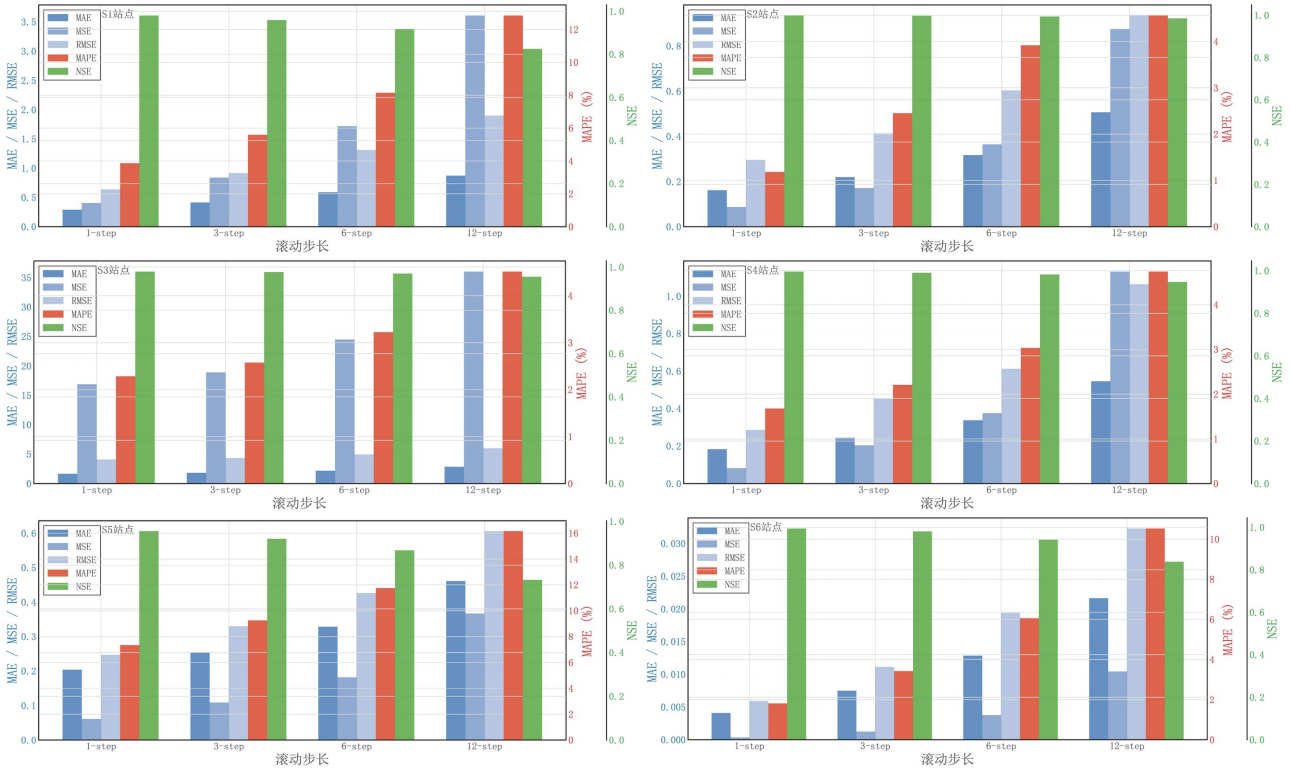


图 10. 各站点预报结果统计指标

0.94，MAPE 均小于 7%，模型在各站点的预报结果均可以提供有价值的参考，S2、S3、S4、S6 站点的预报结果依旧具有较高的精度和可信度；随着滚动预报步数的进一步增加到 12 步，S1、S5、S6 站点的预报精度开始呈现快速下降趋势，预报误差显著上升，S2、S3、S4 站点的预报精度依旧保持在较高水平，NSE 均在 0.94 以上，MAPE 均小于 5%。综上而言，本研究构建的 LSTM 洪水预报模型预报精度满足短期洪水预报需求，可以作为基于数据驱动方法的洪水预报模型进行“测算一体”体系构建验证。

4.2. 端侧设备部署性能验证

4.2.1. 体积压缩效果对比

基于 ONNX 对 PyTorch 模型进行转换时，采用 onnxoptimizer 工具进行算子融合优化，优化策略包括消除死节点、消除 Identity 节点、融合连续 concat/transpose/squeeze 操作、提取常量到初始化器等。量化方面，采用 onnxconverter-common 工具将模型权重从 FP32 转换为 FP16 精度，并采用 onnxruntime 的动态量化功能进一步将权重从 FP32 量化到 INT8。在转换过程中，同时将 scaler 标准化参数和模型配置信息打包到 ONNX 模型的 metadata 中，实现模型的自包含部署。

为便于模型在端侧快速部署、离网独立运行以及便捷接入其它功能模块，模型推理功能被写成了标准的 API 接口并使用 PyInstaller 将 API 打包为可执行文件，打包的文件仅包含模型推理所必须的依赖文件，并使用 UPX 进行了文件压缩。各站点 PyTorch 原始模型和经 ONNX 转换处理后的模型文件体积以及打包好的推理模型可执行文件体积，具体情况如表 7 所示。

从表 7 的对比数据可以看出，ONNX 优化的模型文件体积相较于 PyTorch 原生模型文件体积减少了 67%，可执行文件体积减少了 69%。整体来看，PyTorch 原生模型文件的体积对现在的设备而言也在可承担范围内，ONNX 的主要优势体现在可执行文件的体积压缩上，由于 PyTorch 的原生模型文件依赖庞大的 PyTorch 框架进

行推理，这极大地增加了可执行文件的体积，对端侧设备上宝贵的存储空间造成了巨大的负担；而 ONNX 优化后的模型文件仅依赖轻量的 ONNX Runtime 引擎进行模型推理，打包后的可执行文件剔除了体积庞大的 PyTorch 框架，仅保留轻量的 numpy、pandas、ONNX Runtime 等必要依赖，大大降低了可执行文件体积，使得其能在端侧设备上快速部署。

表 7. 文件体积对比

站点名称	PyTorch		ONNX	
	模型文件体积/KB	可执行文件体积/KB	模型文件体积/KB	可执行文件体积/KB
S1	2450	220,893	816	68,812
S2	2450	220,893	816	68,812
S3	2448	220,893	816	68,812
S4	2450	220,893	816	68,812
S5	2450	220,893	816	68,812
S6	2448	220,893	816	68,812

4.2.2. 端侧设备推理速度以及资源使用情况

本研究选择的 6 个监测站点中，S1、S2、S3、S4 站点工作站处理器为 Intel (R) Core (TM) i7-10810U，主频 1.10 GHz，内存 16 GB，系统类型为基于 x64 处理器的 64 位操作系统，操作系统为 Windows Server 2019 Datacenter；S4、S5 站点工作站处理器为 Intel (R) N100，主频 806 MHz，内存 16 GB，系统类型为基于 x64 处理器的 64 位操作系统，操作系统为 Windows Server 2019 Datacenter。在各站点工作站上使用相同数据对比 PyTorch 原始模型、经 ONN 轻量化处理后的模型的推理速度、冷启动耗时、模型加载内存增量 3 项指标，各项均取 30 次实验结果均值。

从表 8 和表 9 的结果来看，ONNX 优化的模型相较于 PyTorch 原生模型在推理速度上呈现明显提升：其中 S1~S4 站点的平均推理耗时降幅约 7%~13%，S5~S6 站点的平均推理耗时降幅约 25%~31%；不过两框架的单次推理用时均处于毫秒级别，单轮推理速度提升的实际感知有限。在资源占用方面，PyTorch 模型加载耗时 21433.84~21641.44 ms，而 ONNX 模型加载耗时仅 145.49~146.32 ms，加载耗时降幅超 99%；模型加载内存增量方面，PyTorch 模型加载内存增量为 111.97 MB，ONNX 模型仅 0.542~0.544 MB，内存占用大幅降低。在推理精度上，ONNX 优化的模型相较于 PyTorch 原生模型预报结果的 NSE 没有变化。这证明使用 ONNX 优化后，不仅能有效提升模型推理速度，还能大幅降低模型加载耗时和内存占用，且完全不会损失模型预报精度。

表 8. PyTorch 模型推理速度与资源使用结果

站点名称	滚动预报步长	平均推理耗时/ms	最小推理耗时/ms	最大推理耗时/ms	模型加载耗时/ms	模型加载内存增量/MB	NSE
S1	1	6.21	5.83	10.54	21433.841	111.97	0.98
	3	18.31	15.94	22.67			0.96
	6	37.00	32.83	43.20			0.92
	12	73.06	66.36	81.06			0.83
S2	1	6.21	5.83	10.54	21433.841	111.97	0.99
	3	18.31	15.94	22.67			0.99
	6	37.00	32.83	43.20			0.99
	12	73.06	66.36	81.06			0.98

续表

S3	1	6.21	5.83	10.54	21433.841	111.97	0.98
	3	18.31	15.94	22.67			0.97
	6	37.00	32.83	43.20			0.97
	12	73.06	66.36	81.06			0.95
S4	1	6.21	5.83	10.54	21433.841	111.97	0.99
	3	18.31	15.94	22.67			0.99
	6	37.00	32.83	43.20			0.98
	12	73.06	66.36	81.06			0.95
S5	1	6.95	5.74	11.93	21641.438	111.97	0.95
	3	19.63	17.06	23.92			0.92
	6	39.88	36.24	47.94			0.87
	12	76.19	67.32	95.31			0.73
S6	1	6.95	5.74	11.93	21641.438	111.97	0.99
	3	19.63	17.06	23.92			0.98
	6	39.88	36.24	47.94			0.94
	12	76.19	67.32	95.31			0.84

表 9. ONNX 模型推理速度与资源使用结果

站点名称	滚动预报步长	平均推理耗时/ms	最小推理耗时/ms	最大推理耗时/ms	模型加载耗时/ms	模型加载内存增量/MB	NSE
S1	1	5.80	5.14	8.82	146.32	0.544	0.98
	3	15.89	13.89	19.54			0.96
	6	32.50	29.28	35.94			0.92
	12	66.70	62.51	72.43			0.83
S2	1	5.80	5.14	8.82	146.32	0.544	0.99
	3	15.89	13.89	19.54			0.99
	6	32.50	29.28	35.94			0.99
	12	66.70	62.51	72.43			0.98
S3	1	5.80	5.14	8.82	146.32	0.544	0.98
	3	15.89	13.89	19.54			0.97
	6	32.50	29.28	35.94			0.97
	12	66.70	62.51	72.43			0.95
S4	1	5.80	5.14	8.82	146.32	0.544	0.99
	3	15.89	13.89	19.54			0.99
	6	32.50	29.28	35.94			0.98
	12	66.70	62.51	72.43			0.95

续表

S5	1	4.92	4.04	8.69	145.49	0.542	0.95
	3	13.75	12.06	21.16			0.92
	6	27.53	24.78	32.65			0.87
	12	57.23	52.12	66.85			0.73
S6	1	4.92	4.04	8.69	145.49	0.542	0.99
	3	13.75	12.06	21.16			0.98
	6	27.53	24.78	32.65			0.94
	12	57.23	52.12	66.85			0.84

4.3. 讨论

本研究以解决传统洪水预报云端部署中存在的问题为核心，将 ONNX 轻量化技术与 LSTM 时序预报模型结合，实现了洪水预报模型在端侧 AiFlow 监测设备的测算一体化部署，从技术实现和实际应用层面验证了端侧轻量化洪水预报的可行性，为水文预报的端侧化、实时化发展提供了新的技术路径。

从应用价值来看，相较于传统的云端集中部署模式，本研究提出的端侧测算一体方案从根源上规避了数据传输延迟、网络信号依赖、运维成本高的问题。模型在端侧设备完成本地数据处理与预报推理，实现了“即测即报”，大幅提升了洪水预报的时效性，对于汇流速度快的中小流域而言，能够为应急调度、避洪转移等争取关键时间；同时，端侧部署减少了云端服务器集群建设、网络链路维护以及水文数据传输存储的成本，且模型可在偏远无网地区离网运行，填补了传统云端预报的场景盲区，提升了洪水预报体系全天候、全区域适配能力。

从技术实现来看，ONNX 框架的轻量化处理在保证预报精度基本无损的前提下，实现了模型体积、资源占用的大幅优化，解决了深度学习模型参数量大、计算开销高而难以在端侧部署的问题。通过将标准化参数、模型配置信息打包至 ONNX 模型 metadata，以及将推理功能封装为标准 API 并压缩为可执行文件，实现了模型的自包含部署与端侧快速接入，提升了技术的工程化应用价值。此外，本研究设计的轻量化数据预处理流程、增量更新的滑动窗口滚动预报策略，进一步降低了端侧推理的计算开销，适配了端侧设备的算力与存储约束。

本研究仍存在一定的局限性。其一，模型架构方面，本研究采用基础的双层 LSTM 模型完成测算一体的可行性验证，未充分挖掘更先进时序模型的预报潜力，Transformer、Mamba、Informer 等架构在长时序水文数据预报中表现出更优的性能，后续可将此类模型纳入研究体系，结合 ONNX 完成轻量化改造，进一步提升端侧模型的预报精度与长步长预报能力；其二，硬件适配方面，本研究仅在 x64 架构的端侧工作站完成实验验证，针对 ARM 等主流嵌入式硬件平台的适配性研究不足，后续需开展多硬件平台的部署测试，优化算子调用与硬件指令集匹配，提升技术的普适性；其三，数据融合方面，本研究仅基于流量单要素时序数据构建预报模型，未融合降雨、水位、流域下垫面等多源水文气象数据，后续可引入多源数据构建融合模型，提升模型对洪水过程的表征能力与预报鲁棒性；其四，应用场景方面，本研究针对河道流量开展预报，后续可将端侧轻量化测算一体方案拓展至山洪、城市内涝等其他洪涝灾害类型的预报中，丰富技术的应用场景。

5. 结论

本研究针对传统洪水预报模型云端部署存在的延迟高、成本高、端侧场景适应性差等问题，构建了基于 ONNX 的端侧轻量化 LSTM 测算一体洪水预报体系，以山东省内 6 个 AiFlow 视觉测流监测站点的流量数据为基础，从预报精度、轻量化效果、端侧部署性能三个维度开展验证，得出以下结论：

1) 构建的双层堆叠 LSTM 洪水预报模型具备可靠的短期预报精度，完全满足中小河流洪水预报的实际需

求。模型在 6 个监测站点的 1、3 步滚动预报中 NSE 均高于 0.91，MAPE 均小于 10%；6 步滚动预报 NSE 均高于 0.86，MAPE 均小于 12%；仅 12 步滚动预报中部分站点精度出现明显下降，整体呈现符合误差累积规律的精度变化特征，为端侧测算一体体系构建提供了可靠的模型基础。

2) ONNX 对模型的轻量化优化效果显著，实现了精度与轻量化的平衡。经 ONNX 计算图优化、算子融合与量化处理后，模型文件体积减少 70%~71%，可执行文件体积减少 81%；端侧推理冷启动耗时降低 53%~61%，预报结果的 NSE 基本没有变化，在大幅降低模型存储与启动开销的同时，保证了预报精度基本无损。

3) 轻量化后的模型在端侧设备上表现出优异的运行性能，完全适配端侧监测设备的资源约束。模型在端侧运行时，模型推理加载内存降低 86%~87%，单次推理耗时处于毫秒级别，且加载耗时较原始 PyTorch 模型降幅超 99%，实现了“数据采集、本地推理、预报输出”的全流程闭环。

4) 本研究提出的端侧轻量化测算一体洪水预报方法，突破了传统云端部署的场景与成本限制，显著降低了洪水预报的运维成本和预报延迟，成功实现了河道流量的端侧测算一体化。该方法为中小河流、偏远地区的洪水预报提供了一种低成本、高适配、易部署的技术方案，也为深度学习模型在水文水资源领域的端侧应用提供了技术参考。

参考文献

[1] 方宏远, 郑飞飞, 狄丹阳, 等. 考虑河道分洪措施与洪涝灾害风险的应急避难选址研究[J]. 人民黄河, 2025, 47(9): 9-14+36.

[2] 童超, 詹晗煜, 崔罡, 刘康, 欧阳磊, 肖宏宇. 融合 ResNet-18 与水动力模型的洪水演进快速预测[J]. 水资源保护, 2026, 42(1): 129-136.

[3] 罗秋实, 沈洁, 李舒. 郑州市洪涝模拟模型研究[J]. 人民黄河, 2024, 46(7): 42-47+78.

[4] 张倬豪. 基于 HEC-RAS 二维建模的渗漏屏障有效性分析与洪水预报[J]. 武汉理工大学学报, 2024, 46(1): 97-104.

[5] 杨宇, 王喆, 刘贞鹏, 等. 基于 TUFLOW 模型的洪水实时预测预报技术研究与实践[J]. 广西水利水电, 2022(5): 94-99.

[6] 程桂芳, 周芸. 黄河三门峡水文站非汛期流量预报研究[J]. 人民黄河, 2025, 47(4): 38-43+57.

[7] 刘浩, 张玲. 基于 Transformer 模型的洪水预报研究与分析[C]//河海大学, 浙江省水利学会, 上海市水利学会, 江苏省水利学会, 安徽省水利学会. 2025 (第十三届)中国水利信息化技术交流会论文集. 2025: 1329-1339.

[8] 蔡金华. 基于 DWT 与 Informer 融合算法的洪水预测模型研究[D]: [硕士学位论文]. 武汉: 武汉纺织大学, 2024.

[9] 段生月, 王长坤, 张柳艳. 基于正则化 GRU 模型的洪水预报[J]. 计算机系统应用, 2019, 28(5): 196-201.

[10] VASWANI, A., SHAZEER, N., PARMAR, N., et al. Attention is all you need. In: GUYON I., VON LUXBURG U., BENGIO S., et al., Eds., Advances in neural information processing systems 30 (NIPS 2017). Long Beach: Curran Associates, Inc., 2017: 6000-6010.

[11] ZHOU, H., ZHANG, S., PENG, J., et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. 2020.

[12] GRAVES, A. Long short-term memory. In Studies in computational intelligence. Berlin: Springer, 2012: 37-45. https://doi.org/10.1007/978-3-642-24797-2_4

[13] 钱峰, 成建国, 夏润亮, 等. 数字孪生水利“天空地水工”一体化监测感知体系构建与应用初探[J]. 中国水利, 2024(24): 39-47.

[14] 李小静, 肖文. 数字孪生水利工程建设费用测算[J]. 水利技术监督, 2026(2): 35-38+76.

[15] LEE, Y. J., HWANG, J. Y., PARK, J., JUNG, H. G. and SUHR, J. K. Deep neural network-based flood monitoring system fusing rgb and lwir cameras for embedded iot edge devices. Remote Sensing, 2024, 16(13): 2358. https://doi.org/10.3390/rs16132358

[16] THIRUGNANASAMMANDAMOORTHI, P., GHOSH, D., DEWANGAN, R. K., et al. FloodNet-Lite: A lightweight deep learning for flood mapping using remote sensing data with optimized UNet and edge deployment approach in 6G. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2025, 18: 20294-20314. https://doi.org/10.1109/jstars.2025.3591406

[17] SAMIKWA, E., VOIGT, T. and ERIKSSON, J. Flood prediction using IoT and artificial neural networks with edge computing. In 2020 International conferences on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData) and IEEE congress on cybermatics

(Cybermatics). Piscataway: IEEE, 2020: 234-240. https://doi.org/10.1109/ithings-greencom-cpscom-smartdata-cybermatics50389.2020.00053

[18] 赋设备“智慧”一视见“水情”——武汉大学智慧水业研究所所长刘炳义教授谈 AiFlow 视频测流产品[J]. 中国水利, 2021(12): 114-118.

[19] LE, T. D., BERCEA, G. T., CHEN, T., et al. Compiling ONNX neural network models using MLIR. 2020.

[20] 刘婷利. 视觉测流技术在郴州水文站的应用研究[J]. 人民珠江, 2025, 46(S1): 97-99.

[21] YEO, I. K., JOHNSON, R. A. A new family of power transformations to improve normality or symmetry. Biometrika, 2000, 87(4): 954-959. https://doi.org/10.1093/biomet/87.4.954