IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication December 09, 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3513256

Multivariate Segment Expandable Encoder-Decoder Model for Time Series Forecasting

YANHONG LI[1] and DAVID C. ANASTASIU[1], (Member, IEEE)
[1]Computer Science and Engineering, Santa Clara University, Santa Clara, CA 95053 USA
Corresponding author: David C. Anastasiu (e-mail: danastasiu@scu.edu).

ABSTRACT We present the Multivariate Segment-Expandable Encoder Decoder (MSEED), an advanced framework designed to solve the problem of extreme-adaptive multivariate time series forecasting. MSEED contains a hierarchical encoder-decoder architecture, a short-term-enhanced subnet, and a feature assembling layer that effectively integrates spatial and temporal information across the multivariate time series. The model's architecture is designed to capture quantile distributions across segmented subsequences layer by layer, enabling the detection of complex patterns at various scales, which enhances both the accuracy and robustness of forecasts. Moreover, MSEED incorporates a simple vanilla encoder-decoder model to strengthen practical short-term rolling predictions. The framework has been rigorously tested across four challenging datasets, focusing on two critical forecasting scenarios: long-term predictions for three days ahead and rolling predictions every four hours, simulating real-time decision-making in water resource management. In our experiments, MSEED consistently outperformed state-of-the-art models, showing improvements in forecasting accuracy from 18% to 74%.

INDEX TERMS Deep learning, representation learning, oversampling policy, streamflow prediction, hydrologic prediction, LSTM, time series.

I. INTRODUCTION

Time series forecasting is critical in a wide range of domains, including meteorology [1], energy management [2], and financial markets [3]. Yet, the task becomes hard when faced with datasets having pronounced skewness, complicating accurate long-term predictions. Taking hydrology as an example, streamflow predictions are convoluted due to multifaceted and unpredictable variables like weather patterns, geographical features, and human activities. Such intricacies make it harder to obtain accurate forecasts in this field. One of the main issues is capturing *long-range dependencies*, which can be understood as the longest signal path between any two positions in the time series. As the length of this path increases, the dependencies become more complex and difficult to model, so the models need a longer historical input to effectively learn these long-term patterns.

Forecasting from highly-skewed [4] and heavy-tailed datasets presents a myriad of challenges. Datasets with extreme events often suffer from an imbalance in training samples, with a majority of data points clustered at lower values and only a few at higher extremes. This skewed distribution can hinder traditional prediction algorithms from effectively capturing the underlying patterns of these anomalies. These rare events typically follow distributions that differ significantly from the bulk of the data, requiring specialized approaches to manage their non-Gaussian characteristics.

The challenge becomes even more significant when dealing with long-term multivariate time series forecasting that includes extreme values. The Transformer model, originally recognized for its achievements in language processing and computer vision [5], has recently extended its influence to time series forecasting, demonstrating robust capabilities in capturing intricate dependencies within sequences [6]–[8]. Nevertheless, a typical approach in these models is to compress many variables from the same timestamp into a single token, which can obscure important multivariate connections. Furthermore, the limited receptive field associated with single timestamp embeddings might fail to effectively capture useful information, particularly for events that are temporally misaligned. Current methods [9], [10] excel in forecasting normally distributed data; however, their accuracy decreases considerably with highly-skewed time series.

In our previous work, we introduced the Segment-Expandable Encoder-Decoder (SEED) model [11], tailored for univariate time series with high skewness and a heavy tail distribution. However, SEED is not suitable for multivariate time series as it cannot capture complex relationships among multiple time series inputs and has not been previously tested in a rolling prediction scenario, which holds significant practical value. Building on this foundation, we present the Multivariate Segment-Expandable Encoder-Decoder (MSEED) model in this study, tailored to fill this gap.

- **Segment Representation Integration**: MSEED integrates multivariate segment representation learning with a novel multi-tiered encoder-decoder framework.
- **Feature Assembling**: MSEED employs a method that merges spatial and temporal data, tailored to manage the intricate dynamics of multivariate time series, which enables the detection of complex patterns at various scales.
- **Short-Term-Enhanced SubNet**: MSEED contains a simplified component specifically designed to improve efficient short-term rolling predictions.
- **GMM-Based Oversampling Strategy**: MSEED integrates a Gaussian Mixture Model (GMM)-based sampling strategy to identify critical samples from imbalanced datasets, enhancing forecasting precision for heavily skewed time series.

Our comprehensive experiments highlight MSEED's significant potential for practical applications in multivariate, skewed, long-term time series predictions. MSEED consistently outperformed state-of-the-art models, improving forecasting accuracy by 18% to 74%.

II. RELATED WORK

A. TRADITIONAL METHODS

Time series prediction has been investigated for many years. Traditional methods for accurately predicting future values in time series include the univariate Autoregressive (AR), Moving Average (MA), Simple Exponential Smoothing (SES), and Extreme Learning Machine (ELM) algorithms, and most famously the Autoregressive Integrated Moving Average (ARIMA) [12] method and its several variants. Gaussian Process Regression (GPR) [13] and Quantile Regression (QR) [14] were used in some studies to not only predict but also quantify forecast uncertainty. Tree-based models, such as classification and regression trees (CARTs) and random forest (RF), have been employed due to their computational efficiency and ability to handle predictors without assuming any specific distribution. Additionally, Prophet [15] uses an additive model that captures nonlinear trends in the data, incorporating seasonal and holiday effects at various time scales, including annual, weekly, and daily patterns.

B. MULTIVARIATE TIME SERIES FORECASTING

In the realm of multivariate time series forecasting, studies have employed a variety of techniques ranging from traditional models like vector autoregression (VAR) and multivariate exponential smoothing to more contemporary deep learning approaches. VAR models, as described by Lütkepohl [16], statistically capture linear relationships across multiple dimensions and over time. Moreover, graph neural networks (GNNs) [17], [18] are employed to effectively address cross-dimensional dependencies by merging temporal and graph convolutional layers. Deep neural networks (DNNs) have demonstrated significant strengths across various domains. While traditional feed-forward deep learning models often struggle with time series data due to varying lengths and temporal dependencies, WaveNet [19] excels in generating high-quality audio and has proven effective for time series prediction tasks as well [20]. Similarly, DeepAR [21], a probabilistic forecasting model based on a Recurrent Neural Network (RNN) encoder-decoder architecture, leverages the autoregressive property of time series to generate probabilistic forecasts, allowing for uncertainty estimation.

C. TRANFORMER-BASED METHODS

Recent research has demonstrated the Transformer model's ability to boost prediction power [22], [23]. However, the Transformer model suffers from a number of serious drawbacks that prohibit it from being directly applicable to long time series forecasting, including quadratic temporal complexity, high memory utilization, and built-in limitations of the encoder-decoder design. To address these issues, alternative methods like Autoformer [24] and Reformer [25] have been proposed to improve the transformer's dependency discovery and representation ability. Informer [6] proposed a ProbSparse self-attention mechanism and a generative style decoder, while FEDFormer [7] represents time series by randomly selecting Fourier components in an attempt to improve efficiency compared to the standard Transformer. PatchTST [23] utilizes patching techniques to enhance time series modeling by extracting local semantics and ensuring channel independence. Crossformer [10] incorporates a cross-scale embedding layer along with Long Short Distance Attention (LSDA), enabling it to effectively capture dependencies that span time and multiple variables in multivariate time series. Meanwhile, iTransformer [9] refines the inputs for Transformer models to enhance time-series modeling, focusing on improving data interpretation and forecasting accuracy.

While transformer-based models excel at detecting long-range dependencies using self-attention mechanisms, their application to long-term forecasting often leads to decreased accuracy or higher computational demands, limiting their practicality [8]. Moreover, current approaches to long-term time series forecasting have typically overlooked the challenges posed by heavily skewed datasets.

D. EXTREME ADAPTIVE METHODS

Handling datasets with infrequent or extreme events presents significant challenges in time series prediction, requiring the creation of specialized algorithms for exact forecasting under

these circumstances. An and Cho [26] designed a method for anomaly detection that uses reconstruction probability to reflect the intrinsic variability of data distributions. Ding et al. [27] adopted a different tactic by enhancing the capability of deep learning models to identify and predict exceptional events. Their method adjusts predictions based on how closely current data resembles past extreme events, though it may suffer from high memory demands and potential generalizability issues. This approach was later augmented by the Generalized Extreme Value Loss (GEVL) [28], replacing the Gaussian kernel with heavy-tailed distribution kernels like Gumbel and Frechet for loss estimation. Instead of modifying predictions, this method adjusts the loss estimator to a heavy-tailed distribution. The variational disentangled extremal (VIE) classifier [29] utilizes representation learning with a combination of Gaussian and Generalized Pareto distribution priors to efficiently classify extreme event data. Finally, we previously introduced the NEC+ model [30], which trains three predictors simultaneously to maintain excellent forecasting performance for reservoir water level prediction, and DAN [31], which learns and merges rich representations to adaptively predict streamflow.

Our proposed MSEED model incorporates several innovative methods, enabling effective handling of multivariate time series forecasting, including in rolling prediction contexts. The model captures both broad and detailed trends within skewed datasets, offering a more efficient and precise method for forecasting skewed time series. We validate its performance through year-long rolling predictions on four hydrologic datasets, demonstrating its superior forecasting capabilities.

III. PRELIMINARY

A. PROBLEM STATEMENT

In this work, we are tacking the problem of single-target multivariate time series forecasting. Given historical data from multiple length-t observed series, from x_1 to x_m, we are aiming to predict the next h time steps for the first time series x_1. The problem can be described as,

$$\begin{bmatrix} x_{\mathbf{1},1} & \cdots & x_{\mathbf{1},t} \\ x_{2,1} & \cdots & x_{2,t} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,t} \end{bmatrix} \in \mathbb{R}^{m \times t} \rightarrow [x_{\mathbf{1},t+1}, ..., x_{\mathbf{1},t+h}] \in \mathbb{R}^h,$$

where $x_{i,j}$ denotes the value of time series i at time j. The matrix on the left are the inputs, and $x_{1,t+1}$ to $x_{1,t+h}$ are the outputs of our method. We define the group of related time series x_2 to x_m as *auxiliary series*. Root mean square error (RMSE) and mean absolute percentage error (MAPE), as standard scale-free metrics, are used to evaluate forecasting performance.

B. DATA DESCRIPTIONS

As shown in FIGURE 1, our research leverages a hydrologic dataset first documented in [31], which includes streamflow measurements from four Californian streams—Ross, Saratoga, UpperPen, and SFC—alongside data from four corresponding rain sensors. Rainfall serves as auxiliary data within our problem framework, aiding in the prediction of streamflow. Reflecting California's dry summer season, our analysis specifically targets the wetter months from September to May, omitting the summer months to align with the original study's design. These streams are pivotal for the health of California's freshwater ecosystems, with regulated streamflow and water retention being crucial for supporting native species and habitat sustainability. Typically, these streams exhibit stable low flows during the dry summer months and experience pronounced surges during the wetter winter months, reflecting the extreme values characteristic of this long-term forecasting dataset.

TABLE 1: Input Data Statistics

Statistic / **Streamflow**	Ross	Saratoga	UpperPen	SFC
min	0.00	0.00	0.00	0.00
max	1440.00	2210.00	830.00	7200.00
mean	2.91	5.77	6.66	20.25
std. deviation	24.43	26.66	21.28	110.03
skewness	19.84	19.50	13.42	18.05
kurtosis	523.16	697.78	262.18	555.18
Statistic / **Rainfall**	Ross	Saratoga	UpperPen	SFC
min	0.00	0.00	0.00	0.00
max	0.43	4.01	7.68	8.34
skewness	18.11	68.31	413.02	406.11
kurtosis	524.93	17037.06	254167.52	312091.60

Table 1 presents various statistics for our input time series, offering insights into their distribution characteristics such as minimum, maximum, skewness, and kurtosis. The high skewness and kurtosis suggest a significant departure from the normal distribution, indicating a prevalence of extreme values or outliers in our dataset.

C. PIECEWISE LINEAR REPRESENTATION OF TIME SERIES

Time series databases are becoming increasingly popular, and several high level representations have been proposed, such as Fourier Transforms [32], Wavelets [33], Symbolic Mappings [34] and Piecewise Linear Representation (PLR) [35].

FIGURE 2 shows an example PLR representation of a curve using 8 segments. The piece-wise linear function can be described as:

$$f(x) = \begin{cases} m_1 \cdot x + b_1, & \text{if } x \in [a_1, b_1] \\ m_2 \cdot x + b_2, & \text{if } x \in (a_2, b_2] \\ \dots & \\ m_n \cdot x + b_n, & \text{if } x \in (a_n, b_n] \end{cases}$$

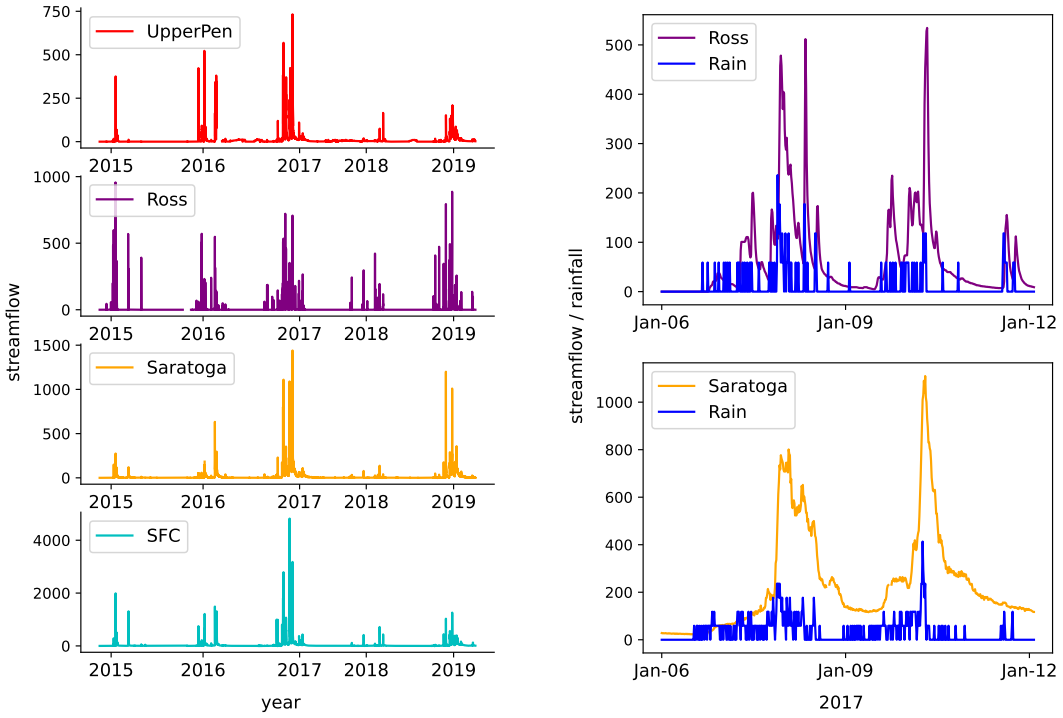


FIGURE 1: This figure on the left illustrates that all four streams experienced significant peaks from September to May over the years 2015-2019. Each streamflow exhibited unique fluctuations during winter due to geographical and meteorological variations. The right side of the figure presents the relationship between streamflow and rainfall for Ross and Saratoga in January 2017. To enhance visibility, the rainfall data has been magnified by 1500 times. While there is a general correlation between the two variables, local changes remain nonlinear. Additionally, the range of streamflow values varies without a consistent yearly pattern, and on a finer scale of every 15 minutes, the fluctuations are even more unpredictable, posing a considerable challenge for this study.

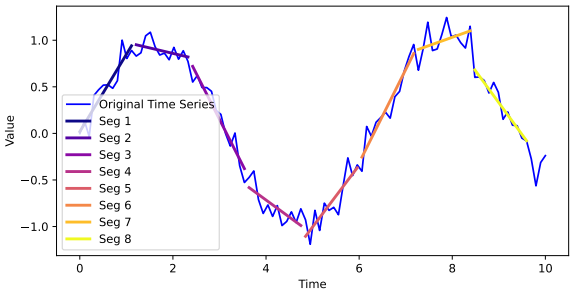


FIGURE 2: Segment representation example. By dividing the time series into multiple segments and fitting a linear regression model to each segment, PLR captures the changing patterns and trends in the data more effectively compared to a single linear regression model.

In this representation, the function $f(x)$ is defined by linear segments between each pair of points (a_i, b_i). The slopes $(m_1, m_2, \ldots, m_n)$ and intercepts $(b_1, b_2, \ldots, b_n)$ determine the behavior of the function over the corresponding intervals. PLR has been used in data mining applications for fast similarity search [36], novel distance measures [37], concurrent analysis of text and time series, vehicle speed estimation [38], and change point detection [39]. PLR simplifies the representation of time series, making their analysis more efficient while preserving key characteristics. In essence, PLR splits a series into several segments such that the maximum error of each segment does not exceed a threshold [40]. However, the PLR algorithm mainly describes the linear relationship of the multi-segment representation and is often used as a preprocessing step to reduce both the space and computational cost of storing and transmitting time series.

In our research, inspired by PLR, we proposed a segment-expandable encoder-decoder architecture which aims to predict segment mean values layer by layer in an expanding way, with a goal of accurate future predictions for heavily skewed long-term time series.

D. GAUSSIAN MIXTURE MODELS

Gaussian Mixture Models (GMMs) are probabilistic models that hypothesize data generation from a mixture of several Gaussian distributions, each characterized by its own mean and covariance. These models are adept for scenarios where data emerge from distinct subpopulations represented through Gaussian statistics. Formally, a GMM is defined as a weighted sum of M component Gaussian densities,

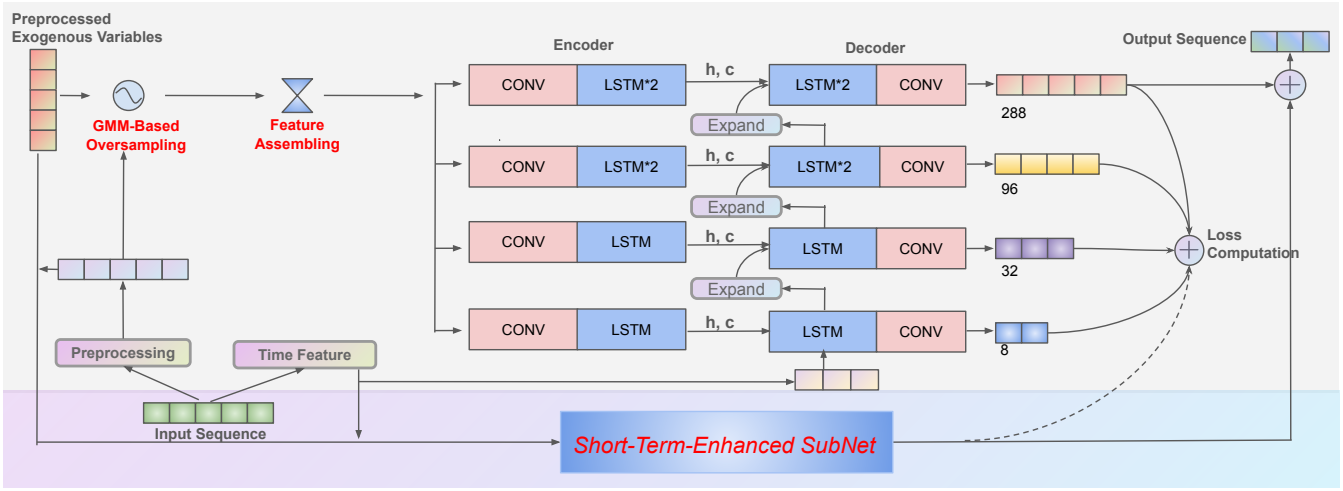


FIGURE 3: The MSEED architecture comprises three core components: embedding, encoder, and decoder. Initially, the input sequence undergoes preprocessing and sampling, with time features generated using sine and cosine transformations. Specifically, each month-day date is encoded into a feature pair using trigonometric or cyclical encoding, capturing the 365-day periodicity within a range of -1 to 1. Each layer's output contributes to the loss as a regularization factor. The output sequence length escalates from lower to upper layers, allowing varied scale information capture. The top layer's output is then refined to produce the final predicted sequence.

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \mathcal{N}(x|\mu_i, \Sigma_i).$$

Here, x represents a data point, w_i are the mixture weights, and $\mathcal{N}(x|\mu_i, \Sigma_i)$ is the Gaussian distribution for component i with mean μ_i and covariance Σ_i. The mixture weights are constrained such that $\sum_{i=1}^{M} w_i = 1$. GMMs are typically optimized through the Expectation-Maximization (EM) algorithm, which iteratively adjusts the parameters of all composite functions to maximize the data likelihood.

In this study, we train a three-component Gaussian Mixture Model (GMM) using data from rain sensors. We then use the Gaussian cluster with the highest mean value as a threshold to identify the occurrence of extreme events.

E. CONTINUOUS QUANTILE VALUE

In statistics and probability, a quantile is a value that divides a probability distribution or a set of data into equal parts. Quantiles are cut points dividing the range of a distribution into continuous intervals with equal probabilities. They can also be applied to continuous distributions, generalizing rank statistics to continuous variables and facilitating extreme value predicitons [41]. For example, if the first quantile equals 5, it indicates that $\frac{1}{5} = 20\%$ of all observations are less than or equal to 5.

In our model, we leverage the concept of a continuous quantile distribution where the 0.85 quantile value signifies that 85% of the data points in a series do not exceed this value. This approach helps us understand the behavior of the majority while also pinpointing upper thresholds crucial for analyzing extreme scenarios.

F. ROLLING PREDICTION

During the inference phase, we implemented a rolling prediction strategy, updating streamflow forecasts every four hours. Each cycle produced 288 data points, forecasting the next three days in 15-minute intervals, based on the prior 1440 time steps or 15 days of data. While our system is designed to provide reliable three-day forecasts, it places a greater emphasis on the accuracy of the first four hours within these predictions, as forecasts are updated every four hours.

This rolling approach is particularly practical for real-time decision-making, which is why our study focuses on enhancing the performance of these short-term predictions within the broader context of maintaining three-day forecast accuracy over an annual cycle.

IV. METHODS

A. OVERVIEW OF THE MSEED FRAMEWORK

As illustrated in FIGURE3, the target time series and auxiliary time series are sampled after undergoing oversampling, forming a set of sequences that pass through a feature assembling layer. At each time point, this layer generates a high-dimensional value, which is then processed by a CNN embedding layer before being fed into four LSTM encoders. The first layer of the decoder is tasked with predicting the quantile values of several segments, achieved by processing its output through a CNN for dimension reduction and involving it in the Kullback-Leibler (KL) divergence loss calculation. The high-dimensional output of this stage then serves as input for the subsequent LSTM decoder layer. For instance, as depicted in FIGURE 3, the initial layer learns the quantile values for eight segments, leading to an output of eight units, whereas the final layer handles the entire sequence, yielding an output

of 288 units. This scaling of responsibilities allows lower layers to operate with simpler computational demands and fewer trainable weights—e.g., 512 in the initial two layers and doubling in the subsequent layers—to adequately power the more complex tasks of the upper layers, which address longer sequence predictions with increased variability.

Notably, we have also incorporated a short-term-enhanced subnet that specifically aims to enhance the accuracy of the forecast in the initial four hours of the prediction sequence. The detailed procedure will be elaborated in subsequent sections.

B. GMM BASED OVERSAMPLING

Given the substantial volume of data, approximately 1.4 million points, and its uneven distribution with a high presence of outliers, a nuanced oversampling strategy is critical. Our approach needs to effectively manage the prevalence of extreme values without compromising the integrity of normal data. In the SEED framework, a simple classification method is used to highlight crucial data points. However, in MSEED, due to the multivariate nature of time series forecasting and the indicative role of auxiliary data, particularly in our datasets, we opt for a more tailored strategy. The auxiliary data, notably rainfall, exhibits an even more skewed and imbalanced distribution. Consequently, we employ rainfall data-driven oversampling to capture rare but significant variations in the dataset, which is described in Algorithm1

Require: Rain data, Number of clusters M, Extreme value threshold factor 1.2, Step size s, Scope ν, Oversampling ratio os
Ensure: Augmented dataset aligned with targeted oversampling
1: Fit a Gaussian Mixture Model (GMM) to the rain data and identify M clusters with mean values $\mu_1, \mu_2, \ldots, \mu_M$
2: Compute z as the maximum mean value from the clusters
3: **for** each random sampled data in the dataset **do**
4: **if** any rain value in the past and future 1.5 days exceeds $1.2 \times z$ **then**
5: Flag this sample for oversampling
6: Identify the peak value within the next 3 days
7: Calculate start point for sampling as $\lfloor \frac{s \times \nu}{2} \rfloor$ positions left of the peak
8: **for** i from 0 to ν with step size s **do**
9: Collect sample at position *start point* + i
10: **end for**
11: **end if**
12: **end for**
13: Cap the oversampled data to $os\%$ of the total training set volume
Algorithm 1: Rainfall-Driven Oversampling Procedure

In this algorithm, we initially use GMM unsupervised training to derive three Gaussian distributions from the rainfall data, considering the distribution with the highest mean as indicative of extreme values. For training data extraction, we randomly select a subsequence comprising 1440 historical values and 288 values to be predicted. If a rainfall value within 1.5 days of the prediction start point exceeds $1.2\times$ the highest mean, we perform oversampling. Specifically, centered around the peak value during the prediction period, we shift $(s \times \nu)/2$ positions to the left and sample ν times with a step size of s. This method aims to address the issue of imbalance by oversampling near extremes, enhancing the model's robustness.

C. FEATURE ASSEMBLING

As shown in FIGURE 4, we introduce a sophisticated feature assembling method that significantly enhances the handling of spatio-temporal dynamics in multivariate time series forecasting. This method adeptly manages data by isolating critical time points across multiple sequences.

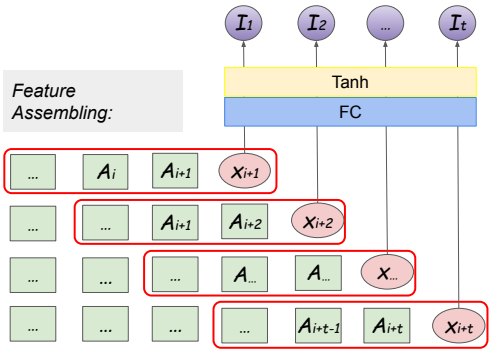


FIGURE 4: Feature Assembling Method

Consider an input sequence of length t. At each time point, say the starting one for simplicity, this approach combines the goal value at that point, relevant values from auxiliary sequences, and their fa preceding values, where fa is the assembling length. These components are combined to form a multidimensional subsequence that is distinct to that time instance.

This multi-dimensional subsequence undergoes transformation through a dense layer integrated with a nonlinear activation function. By assembling the self-contained representations, complex relationships across variables are extracted, promoting the data into an enriched high-dimensional space.

This carefully processed subsequence, which serves as the encoder's input at each unique time point, encodes complex inter-variable connections within its structure by the purposeful aggregation of related sequence data from past events. This capacity enables the MSEED encoder to operate efficiently without being constrained by the time alignment restrictions of conventional models, allowing it to focus more intensively on delicate temporal and spatial correlations within the data.

D. CONVOLUTIONAL EMBEDDING LAYERS

To preprocess the input before it is fed into the LSTM encoder, we use convolutional embedding layers with various kernel sizes. These diverse kernel sizes enable the extraction of features at various spatial resolutions. Smaller kernels excel in detecting local, granular patterns, but larger kernels understand broader, global contexts. This tiered technique allows each layer of the LSTM encoder to focus on processing specific regions of the input sequence.

Furthermore, we use larger kernels with no padding for the initial layers of our convolutional embeddings. This method not only reduces the dimensionality of inputs for the lower-level LSTM layers, making their computational duties easier, but it also addresses the frequent LSTM difficulties of exploding and vanishing gradients by minimizing the sequence length over which gradients must be transmitted.

E. DECODER ARCHITECTURE

In SEED, the mean value is used to progressively learn characteristics of smaller segments, where each segment is based on the outputs of the previous one, refining the neural network's learning through the learning of feature differences. However, in MSEED, considering the potential strong indicative nature of the auxiliary series—due to high skewness and sudden value changes in the series—relying solely on learning the mean might cause the model to consistently orbit around the majority of normal values. To address this, we shifted to using quantile values for approximation. By adjusting this value, we can adapt to the distribution of different datasets without making any prior assumptions about data distribution. This adjustment offers greater flexibility, as the mean is merely one specific instance of various possible quantile values.

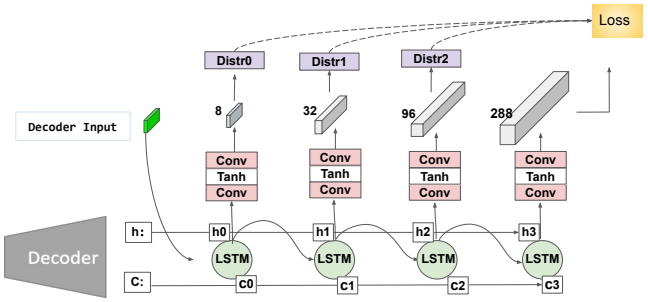


FIGURE 5: Decoder architecture of MSEED.

For example, as illustrated in FIGURE 3, to produce a sequence of 288 predictions, the decoder utilizes four layers. Each layer targets a different segment length, specifically 8, 32, 96, and 288, which divide the output sequence into increasingly granular segments. At the initial tier, the output spans 8 segments, predicting for segments containing 36 data points each, derived from $288/8$. These predictions are then expanded into 32 segments for the next layer, which focuses on segments with 8 points each, achieved by replicating each quantile value four times to form a new sequence as the next layer input. For instance, a vector of quantiles $\langle a, b, \ldots, g, h\rangle$ is expanded to $\langle a, a, a, a, b, b, b, b, \ldots, g, g, g, g, h, h, h, h\rangle$. This method of expansion continues until the final layer, where the output directly corresponds to the full prediction sequence, with each segment reduced to a single data point, effectively mapping each quantile directly to its respective value in the sequence.

This hierarchical technique effectively controls extreme values by anticipating quantile values for sub-segments with variable lengths. Extreme values are scattered across multiple layers of the hierarchy, strengthening the segments that include them. As shown in FIGURE 5, to optimize each layer's effectiveness, we introduce a two-layer CNN following the LSTM stack. This adaptation ensures a one-dimensional output of consistent length and aids in learning the distribution of segment quantile values within the predicted sequence.

F. SHORT-TERM ENHANCED SUBNET

While trends and distributions within the dataset shape long-term forecasts, short-term fluctuations are significantly influenced by external variables. To address this, we introduced an auxiliary Encoder-Decoder that effectively integrates learning both short-term and long-term features. The output of the

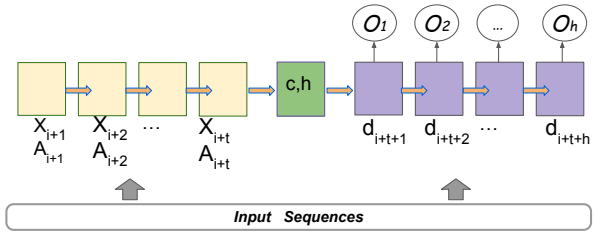


FIGURE 6: Short-Term-Enhanced SubNet of MSEED.

sub-network fulfills two key roles. First, it is integrated with the outputs from the main MSEED network, enhancing the overall prediction accuracy. Second, its capacity for short-term prediction is utilized as a component of the loss function. This inclusion acts as a penalty term, underscoring the critical role of auxiliary variables in refining short-term forecast accuracy. Experiments show that the short-term-enhanced network is important in capturing immediate changes influenced by external factors.

G. MULTIPLE-OBJECTIVE LOSS FUNCTIONS

Our model utilizes the Kullback-Leibler divergence loss as a way to ensure that the predicted segment distributions align with the ground truth segment distributions. By minimizing the Kullback-Leibler divergence between the two distributions, the model is encouraged to iteratively improve predictions of the segment distributions during the training process. For example, in the 4-level setting described in FIGURE 3, the regularization loss term for the ith layer can be described as,

$$\mathcal{L}_i = \mathrm{KL}\left(\mathrm{softmax}(p\_q_i), \mathrm{softmax}(g\_q_i)\right),$$

TABLE 2: Effectiveness comparisons with state-of-the-art methods. Over 1600 test points in the test set were inferenced on all datasets. The best results are in bold and the second best results are underlined.

Methods	Metric	Ross		Saratoga		UpperPen		SFC	
		Three days	Four hours	Three days	Four hours	Three days	Four hours	Three days	Four hours
FEDformer	RMSE	6.01	3.95	6.01	4.82	3.05	2.55	23.54	17.11
	MAPE	2.10	2.05	1.55	1.54	1.87	1.75	2.35	2.16
Informer	RMSE	7.84	6.76	5.04	3.78	5.88	5.00	39.89	23.21
	MAPE	4.05	4.71	1.43	1.54	4.10	3.99	8.64	3.61
Nlinear	RMSE	6.10	2.76	5.23	4.13	1.57	0.51	18.47	5.08
	MAPE	1.99	0.52	0.83	0.82	0.45	0.16	0.92	0.52
Dlinear	RMSE	7.16	3.31	4.33	1.79	3.53	1.35	21.62	8.75
	MAPE	3.10	1.15	1.40	0.65	2.35	0.69	2.74	1.45
LSTM-Atten	RMSE	7.35	6.84	6.49	5.59	6.35	4.75	34.17	23.09
	MAPE	3.74	4.10	1.80	1.79	4.76	3.67	9.90	6.25
NEC+	RMSE	9.44	2.07	1.88	**0.26**	2.22	0.33	17.00	**2.36**
	MAPE	4.80	0.45	0.17	0.07	0.95	0.06	1.07	**0.07**
iTransformer	RMSE	4.56	2.14	2.37	0.94	1.12	0.58	17.04	11.00
	MAPE	0.57	0.43	0.27	0.18	0.11	0.06	0.47	0.54
DAN	RMSE	4.25	2.61	1.80	0.62	1.10	0.43	15.23	3.73
	MAPE	**0.07**	0.46	0.14	0.22	0.15	0.07	0.26	0.22
MSEED	RMSE	**4.21**	**1.57**	**1.70**	0.27	**1.03**	**0.28**	**14.81**	2.99
	MAPE	**0.07**	**0.07**	**0.10**	**0.05**	**0.06**	**0.01**	**0.14**	**0.07**

where p_q_i is the output of the CNN layer in FIGURE 3, which represent the predicted segment quantile values in the ith layer, while g_q_i is the vector of computed ground truth quantile values for the segments in the ith layer. Applying the *softmax* function turns both vectors of quantile values into distributions, which then allows us to compute the *KL* divergence between the two distributions.

$$\mathcal{L}_5 = RMSE(\hat{y}_{aux}[:st], y[:st]),$$
$$\mathcal{L}_6 = RMSE(\hat{y}, y),$$

Similarly, to further emphasize the importance of auxiliary variables in short-term predictions, we use L_5 to regularize the output of short-term-enhanced sub-network $\hat{y}_{aux}$. Here, st represents the length of the short-term interval, which is set to 16 (4 hours) in our experiments. Then, the overall loss is composed as,

$$\mathcal{L} = \lambda \times \left(\sum_{i=1}^{4} \mathcal{L}_i + \mathcal{L}_5 \right) + \mathcal{L}_6$$

where y_p is the output of the top layer of hierarchy encoder-decoder, $\hat{y} = y_p + y_{aux}$, and λ is a multiplier ($\lambda = max(-1 \cdot e^{\frac{epoch}{45}} + 2, 0.1) \times 200$ in our experiments) applied on the regulation items, decreasing with each epoch. Initially, λ is set high to guide the network towards learning polar representations for more accurate predictions. This 'teacher mode' diminishes over time; λ starts at 200 and decreases to 20 as training progresses.

V. EXPERIMENTS

To provide a comprehensive evaluation of our proposed MSEED model, we utilized four distinct streamflow paired with rainfall datasets and compared its performance against eight baseline alternatives. We further assessed the effectiveness of the feature assembling method, the impact of the GMM-based oversampling policy, the quantile approximation, and the short-term-enhanced subnet via a series of ablation studies. Our key findings are:

- Across the four datasets and two scenarios, MSEED consistently outperforms the three second-best models by an average of 18% to 74% in RMSE and MAPE.
- The feature assembling method effectively mines the spatial and temporal relationships within multivariate sequences, enhancing prediction accuracy.
- The GMM-based oversampling policy effectively captures extreme events and improves MSEED's forecasting performance.
- The inclusion of the short-term-enhanced subnet significantly enhances short-term forecasting accuracy without compromising long-term prediction capabilities, aligning the model more closely with practical operational demands.

A. EXPERIMENTAL SETTINGS

Data for training and validation was drawn from January 1988 to August 2021 and we aim to accurately project the streamflow for the subsequent year (September 2021 to May 2022), with predictions made every four hours. Each prediction estimates the upcoming 3 days based on the preceding 15 days of data. Since the sensors measure the streamflow and precipitation every 15 minutes, we are attempting a lengthy forecasting horizon ($h = 288$).

During the inference phase, we predicted streamflow using rolling predictions at intervals of 4 hours. Each prediction, however, inferred 288 data points, i.e., the predicted streamflow over the next 3 days at 15 minute intervals. To make

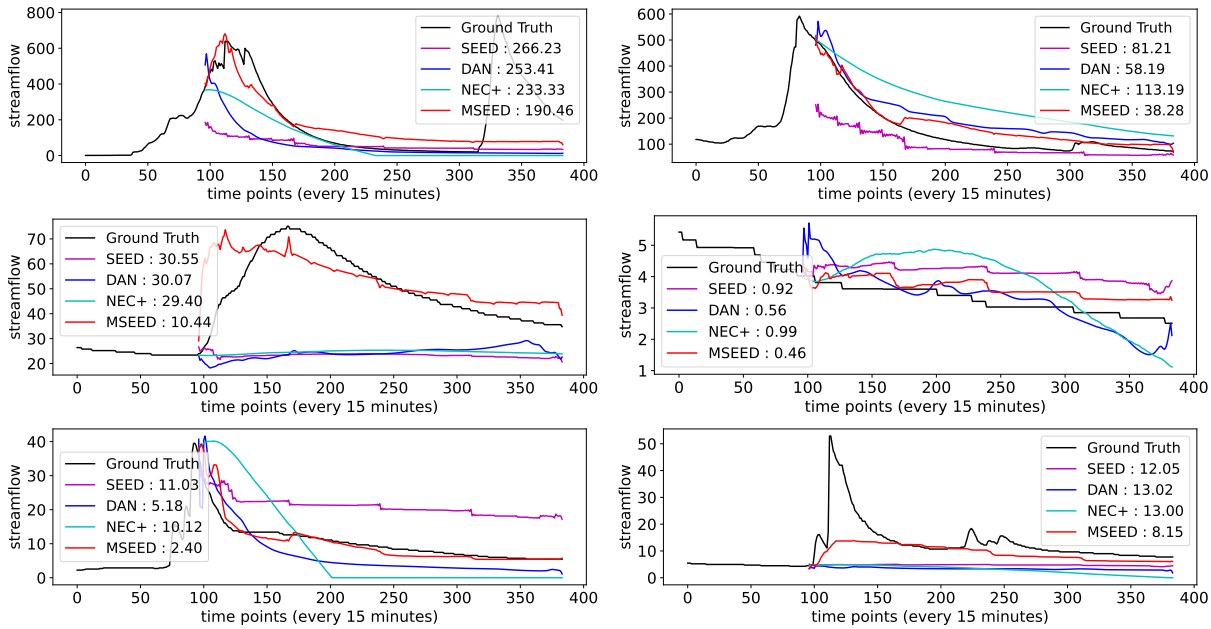


FIGURE 7: Comparative examples with the best baselines reveal that, while the DAN and NEC+ models generally capture the overall trend of the ground truth, MSEED excels in predicting streamflow values with greater accuracy, both in the short and long terms.

these predictions, we utilized the previous 15 days of data, equivalent to 1440 time steps. Prior to model training, all time series underwent pre-processing steps including a logarithmic transformation, $x_i = \log(1 + x_i)\ \forall i$, and standardization (subtracting the mean and dividing by the standard deviation). In order to obtain the final inference predictions, we performed post-processing by reversing the standardization and logarithmic transformations.

In our models, we utilized a 4-layer MSEED architecture, where the first two layer consisted of a 1-layer LSTM and the top two layer consisted of a 2-layer LSTM. After experimenting with different layer widths in [256, 300, 320, 360, 384, 400, 512], we found the best results at (CNN_width, FC_width, LSTM_width) of (256, 512, 384), (400, 512, 300), (384, 384, 360), and (384, 512, 320) nodes per layer for the Ross, Saratoga, UpperPen, and SFC datasets, respectively. In the embedding stage, we employed five CNN layers, each generating 384 channels. The kernel sizes for these layers were set to 4, 3, 2, and 2, respectively, from the bottom to the top layer. The quantile value is 0.9 for UpperPen and 0.85 for the other three datasets. The best feature assembling length *fa* for all sensors was 60.

All models were trained using PyTorch 1.11.0+cu10.2 on a Linux server running Ubuntu 20.04.6. The server was equipped with a 12-core Intel(R) Core(TM) i9-7920X CPU, 128 GB RAM, and 4 NVIDIA RTX 2080 Ti GPUs. However, the algorithm only used one GPU when training the model or performing inference. The code for our method will be made freely available upon publication of the article.

B. BASELINE METHODS

We compared our proposed method, MSEED, against a wide array of state-of-the-art time series and hydrologic prediction methods, introduced earlier in the related work section:

- FEDFormer [7], which combines Transformer with the seasonal-trend decomposition methods, has been shown to be more efficient than the standard Transformer, yielding improved results for long-term series forecasting;
- Informer [6], a transformer-style model for long-term time series prediction with a prob-sparse self-attention mechanism;
- NEC+ [42], a group of LSTM-based models that holds the best performance for *hydrologic* time series prediction in the presence of extreme events;
- NLinear [8], an effective linear model with one order difference preprocessing for long-term time series;
- DLinear [8], a trend decomposed linear model for long-term time series prediction;
- Attention-LSTM [43] serves as a state-of-the-art multivariate model in hydrology;
- iTransformer [9] achieves state-of-the-art performance on a variety of challenging multivariate time series prediction;
- DAN [31] learns and merges rich representations to adaptively predict streamflow.

C. MAIN RESULTS

The experimental results, which are presented in Table 2, show MSEED's superior performance in multivariate time series forecasting, particularly under high value scenarios. Par-

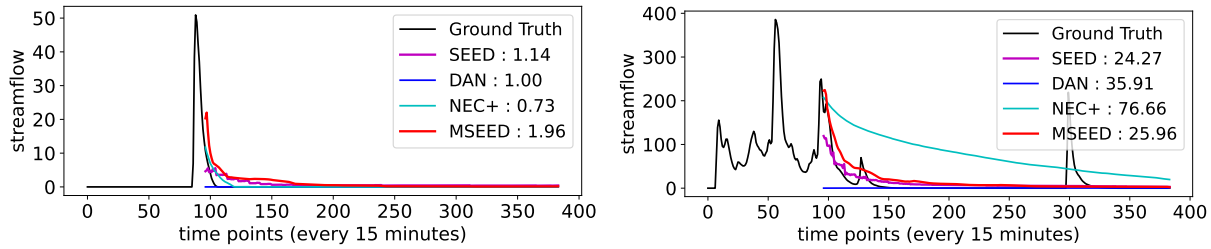


FIGURE 8: Ablation comparative examples with the SEED.

ticularly, MSEED regularly outperforms comparable models overall on all metrics, including some significant improvements on some specific datasets. For example, in 3-day prediction, when compared to DAN, MSEED obtains an impressive 60% improvement in MAPE for the UpperPen dataset, and when compared to NEC+, it improves RMSE by more than 90% and MAPE by 98% for the Ross dataset. Furthermore, MSEED outperforms iTransformer by more than 49% in RMSE and 67% in MAPE for the Saratoga dataset.

Overall, across the four datasets and two scenarios, MSEED improves 18%, 35%, 21% on RMSE and 65%, 74% and 56% on MAPE on average compared to NEC+, iTransformer, and DAN, respectively.

Transformer-based methods such as FEDFormer and Informer, while effective in some scenarios, struggle significantly with datasets exhibiting large variances and fail to adapt effectively to extreme values. In contrast, models tailored specifically for hydrologic data forecasting, such as NEC+ and DAN, though better than more generic models, still fall short of MSEED's performance. Fully connected networks like NLinear and DLinear, which decompose data into main trends and residuals, perform better than some Transformer-based methods in rolling prediction scenarios but do not achieve the high accuracy that MSEED does.

D. VISUAL ANALYSIS

FIGURE 7 displays forecasts for MSEED and the following two top-performing models. As illustrated in the image, MSEED's predictions closely match the ground truth, especially in datasets with significant oscillations. Notably, MSEED better catches short-term oscillations and severe values than DAN, as indicated by lower RMSE values. This improved performance is primarily due to the embeddings' broad expressive capabilities, along with the attention method.

The feature assembling method and the Short-Term Enhanced SubNet in MSEED successfully maintain critical inter-variable relationships and intra-variable temporal sequences, which are essential for good prediction.

VI. ABLATION STUDIES

To examine the effects of our architecture components, we conducted a group of experiments.

TABLE 3: Comparison With Different *os*

Metrics/	3-day		4-hour	
os	RMSE	MAPE	RMSE	MAPE
0	4.44	0.22	2.33	0.15
10%	4.32	0.18	1.63	**0.07**
20%	**4.21**	**0.07**	**1.59**	**0.07**
30%	4.28	0.15	1.73	0.16

A. EFFECT OF THE OVERSAMPLING POLICY

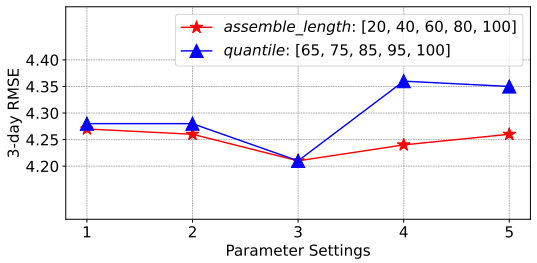
To evaluate the impact of the oversampling policy, we conducted a grid search on the combination values of s and v, aiming to capture more information from the more important sampled sequences. This approach allowed the models to effectively adapt to the unique characteristics of the data and extract valuable insights. We obtained the best results at (s, v) of (1,2), (1,3), (2,4), and (3,4) for the Ross, Saratoga, UpperPen, and SFC datasets, respectively.

Further, based on the optimal (s, v), we initially trained the models without any oversampling, i.e., $os = 0$. TABLE 3 shows the RMSE of the inference on the Ross test sets. We observed that increasing the *os* had a positive impact initially. However, there was a point of diminishing returns, where further increasing the *os* did not lead to a significant decrease in RMSE. This suggests that there is an optimal threshold *os* value beyond which the policy's effectiveness plateaus.

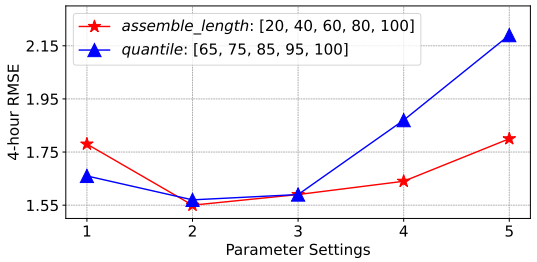
It is worth noting that the effect of this policy was variable on different datasets, likely due to the difference in variance.

B. EFFECT OF THE FEATURE ASSEMBLING METHOD

By holding the optimal settings for Ross constant and varying one parameter at a time, we assessed the impact of two critical factors. Initially, we altered the assembling length from 20 to 100. As shown in FIGURE 9, it is evident that the assembling length plays a role in unveiling the spatial relationships among multivariate data, with a length of 60 yielding the best results for the 3-day forecast. However, too short an assembling length fails to provide enough hints to uncover spatial relationships among different time series, while too long a length leads to information redundancy, negatively impacting prediction accuracy.



(a) 3-day RMSE



(b) 4-hour RMSE

FIGURE 9: RMSE on Ross datasets given assembling length $s \in \{20, 40, 60, 80, 100\}$, regularization quantile $100 \times q \in \{65, 75, 85, 95, 100\}$.

C. EFFECT OF THE APPROXIMATION QUANTILE VALUES

Subsequently, while still keeping other values fixed, we changed the quantile values used for progressive approximation in the loss regularization item. As shown in FIGURE 9, the optimal result was achieved at 0.85. Higher quantile settings guide the model to learn more about extreme events, particularly when the quantile is set to 1, pushing the segment mean towards maximum values. Although this offers a stronger alert for predicting extreme events, it compromises overall forecasting accuracy. On the other hand, a lower quantile, while representing the majority of cases, struggles to aid in predicting extreme values.

D. EFFECT OF THE MULTIVARIATE INPUT

To assess the effectiveness of introducing multivariate capabilities and the specifically designed MSEED, we compared SEED and MSEED across all dimensions in all datasets. As seen in the TABLE 4, while both models perform similarly in 3-day forecasts, MSEED clearly outperforms SEED in 4-hour scenarios. This improvement is attributed to the inclusion of auxiliary variables and the model's successful exploitation of their nonlinear relationships. For instance, in our datasets, rainfall is a stronger indicator of streamflow changes within a few hours rather than over three days. The model leverages this fact to excel in rolling predictions, showcasing its advantage in practical applications. This characteristic is also evident in FIGURE 8; although SEED's 3-day RMSE is slightly better, MSEED's forecast curve is noticeably superior in the initial hours, aligning more closely with real-world application needs.

TABLE 4: Comparison With SEED

	3-day		4-hour	
Metrics	RMSE	MAPE	RMSE	MAPE
		Ross		
SEED	4.23	0.11	1.64	0.07
MSEED	**4.21**	**0.07**	**1.57**	0.07
		Saratoga		
SEED	**1.67**	**0.09**	0.28	0.06
MSEED	1.70	0.10	**0.27**	**0.05**
		UpperPen		
SEED	1.07	0.10	0.32	0.05
MSEED	**1.03**	**0.06**	**0.28**	**0.01**
		SFC		
SEED	**14.44**	0.20	6.26	0.07
MSEED	14.81	**0.14**	**2.99**	0.07

E. EFFECT OF SHORT-TERM-ENHANCED SUBNET

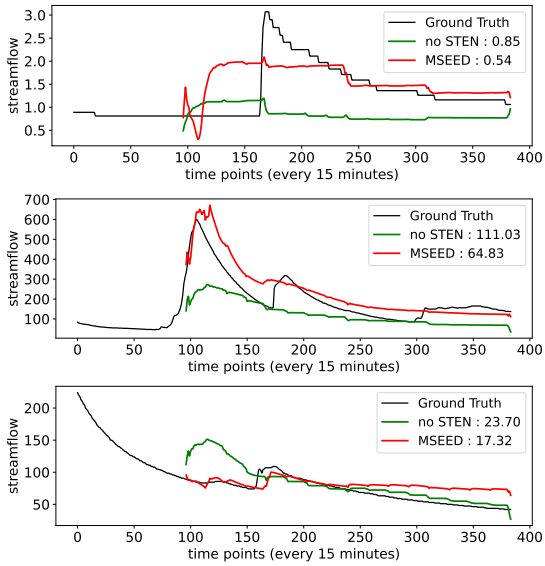


FIGURE 10: Examplesof the Short-Term-Enhanced SubNet effect.

To validate the effectiveness of the Short-Term Enhanced Subnet, we conducted experiments by removing this subnet along with its associated loss regularization item, relying solely on the output from the hierarchical encoder-decoder. In these tests, the RMSE for a three-day forecast on the SFC dataset increased from 14.81 to 15.69, while the RMSE for rolling predictions jumped significantly from 2.99 to 7.39. FIGURE 10 clearly shows that both short-term and long-term forecasting accuracies are adversely affected without the SubNet, especially the short-term predictions.

The inference examples of the model without SubNet are represented by the green line in FIGURE 10. According to these findings, the MSEED model's prediction accuracy may suffer if the SubNet is removed.

VII. CONCLUSION

The Multivariate Segment-Expandable Encoder-Decoder (MSEED) model introduced in this study significantly advances the forecasting of complex, skewed time series. By integrating segment representation learning with a multi-tiered encoder-decoder framework, MSEED effectively captures intricate dynamics across various scales, enhancing both accuracy and granularity of predictions. Its novel components, including the feature assembling layer, a Short-Term-Enhanced Subnet, and an auxiliary-focused Gaussian Mixture Model (GMM)-based sampling strategy, enable it to adeptly handle both long and short-term fluctuations and extreme events, areas where traditional models often falter. Tested across diverse datasets, MSEED consistently outperforms existing multivariate models, demonstrating substantial improvements in multivariate time series forecasting accuracy.

REFERENCES

[1] A. F. Faisal, A. Rahman, M. T. M. Habib, A. H. Siddique, M. Hasan, and M. M. Khan, "Neural networks based multivariate time series forecasting of solar radiation using meteorological data of different cities of bangladesh," *Results in Engineering*, vol. 13, p. 100365, 2022.

[2] Y.-H. Lin, H.-S. Tang, T.-Y. Shen, and C.-H. Hsia, "A smart home energy management system utilizing neurocomputing-based time-series load modeling and forecasting facilitated by energy decomposition for smart home automation," *IEEE Access*, vol. 10, pp. 116 747–116 765, 2022.

[3] K. Sako, B. N. Mpinda, and P. C. Rodrigues, "Neural networks for financial time series forecasting," *Entropy*, vol. 24, no. 5, p. 657, 2022.

[4] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7*. Springer, 2003, pp. 107–119.

[5] 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.

[6] 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, 2021, pp. 11 106–11 115.

[7] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27 268–27 286.

[8] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *arXiv preprint arXiv:2205.13504*, 2022.

[9] 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.

[10] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[11] Y. Li, J. Xu, and D. C. Anastasiu, "Seed: An effective model for highly-skewed streamflow time series data forecasting," in *2023 IEEE International Conference on Big Data (Big Data)*, ser. IEEE BigData 2023. Los Alamitos, CA, USA: IEEE Computer Society, Dec 2023.

[12] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. : Holden-Day, 1976.

[13] J. Han, X.-P. Zhang, and F. Wang, "Gaussian process regression stochastic volatility model for financial time series," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 6, pp. 1015–1028, 2016.

[14] Q. Huang, H. Zhang, J. Chen, and M. He, "Quantile regression models and their applications: A review," *Journal of Biometrics & Biostatistics*, vol. 8, no. 3, pp. 1–6, 2017.

[15] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018. [Online]. Available: https://doi.org/10.1080/00031305.2017.1380080

[16] H. Lütkepohl, "Vector autoregressive models," in *Handbook of research methods and applications in empirical macroeconomics*. Edward Elgar Publishing, 2013, pp. 139–164.

[17] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 766–17 778, 2020.

[18] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[19] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[20] F. Dorado Rueda, J. Durán Suárez, and A. del Real Torres, "Short-term load forecasting using encoder-decoder wavenet: Application to the french grid," *Energies*, vol. 14, no. 9, p. 2524, 2021.

[21] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[22] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[23] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[24] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[25] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[26] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[27] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1114–1122.

[28] M. Zhang, D. Ding, X. Pan, and M. Yang, "Enhancing time series predictors with generalized extreme value loss," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[29] Z. Xiu, C. Tao, M. Gao, C. Davis, B. A. Goldstein, and R. Henao, "Variational disentanglement for rare event modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 469–10 477.

[30] Y. Li, J. Xu, and D. C. Anastasiu, "An extreme-adaptive time series prediction model based on probability-enhanced lstm neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8684–8691.

[31] Y. Li, J. Xu, and D. Anastasiu, "Learning from polar representation: An extreme-adaptive model for long-term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 171–179.

[32] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and information Systems*, vol. 3, pp. 263–286, 2001.

[33] K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*. IEEE, 1999, pp. 126–133.

[34] R. A. K.-l. Lin and H. S. S. K. Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," in *Proceeding of the 21th International Conference on Very Large Data Bases*. Citeseer, 1995, pp. 490–501.

[35] L. Chua and R. Ying, "Canonical piecewise-linear analysis," *IEEE Transactions on Circuits and Systems*, vol. 30, no. 3, pp. 125–140, 1983.

[36] H. Wu, B. Salzberg, and D. Zhang, "Online event-driven subsequence matching over financial data streams," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004, pp. 23–34.

[37] F. Yajing and G. Yujian, "A novel approach based on neural networks and support vector machine for stock price pattern discovery," in *2017 IEEE*

International Conference on Big Knowledge (ICBK). IEEE, 2017, pp. 259–263.

[38] S. Hua, M. Kapoor, and D. C. Anastasiu, "Vehicle tracking and speed estimation from traffic videos," in *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW'18, vol. 1, July 2018, pp. 153–1537.

[39] Q. Tao, L. Li, X. Huang, X. Xi, S. Wang, and J. A. Suykens, "Piecewise linear neural networks and deep learning," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 42, 2022.

[40] Y. Hu, P. Guan, P. Zhan, Y. Ding, and X. Li, "A novel segmentation and representation approach for streaming time series," *IEEE Access*, vol. 7, pp. 184 423–184 437, 2018.

[41] J. Beirlant, G. Dierckx, and A. Guillou, "Estimation of the extreme-value index and generalized quantile plots," *Bernoulli*, vol. 11, no. 6, pp. 949 – 970, 2005. [Online]. Available: https://doi.org/10.3150/bj/1137421635

[42] Y. Li, J. Xu, and D. C. Anastasiu, "An extreme-adaptive time series prediction model based on probability-enhanced lstm neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, pp. 8684–8691, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26045

[43] Y. Le, C. Chen, T. Hang, and Y. Hu, "A stream prediction model based on attention-lstm," *Earth Science Informatics*, vol. 14, pp. 1–11, 06 2021.

YANHONG LI is pursuing her Ph.D. in Computer Science and Engineering at Santa Clara University's School of Engineering. Her research interests are centered on the development of advanced computational models and algorithms, with a keen focus on machine learning, pattern recognition, and deep learning technologies. She has a particular interest in time series representation learning, as well as object detection, and tracking. Additionally, her work extends to the realm of environmental science, where she applies her computational expertise to develop predictive models for hydrologic flow and water level management, aiming to enhance forecasting accuracy and reliability. Li's multidisciplinary approach combines rigorous engineering methodologies with cutting-edge artificial intelligence to address complex real-world challenges.

DAVID C. ANASTASIU is an Assistant Professor in the Department of Computer Science and Engineering at Santa Clara University. His research interests fall broadly at the intersection of artificial intelligence/machine learning, data mining, computational genomics, and high performance computing. Much of his work has been focused on scalable and efficient methods for analyzing sparse data. He has developed serial and parallel methods for identifying near neighbors, characterizing how user behavior changes over time, analyzing traffic based on video sensors, and methods for personalized and collaborative presentation of Web search results, among others. In the biomedical domain, he has worked on methods for sensory-based prediction of Autism in children, searching related biochemical compounds, and designating the severity of kidney disease. Prof. Anastasiu serves on the program committees and senior program committees of the most prominent IEEE and ACM data science-related conferences and his work, which is funded by the National Science Foundation and several industrial partners, has been published in many top-tier conferences and journals.