Learning from Polar Representation: An Extreme-Adaptive Model for Long-Term Time Series Forecasting

Yanhong Li[1], **Jack Xu**[2], **David C. Anastasiu**[1]

[1] Santa Clara University, Santa Clara, CA, USA
[2] Santa Clara Valley Water District, San Jose, CA, USA
yli20@scu.edu, JXu@valleywater.org, danastasiu@scu.edu

Abstract

In the hydrology field, time series forecasting is crucial for efficient water resource management, improving flood and drought control and increasing the safety and quality of life for the general population. However, predicting long-term streamflow is a complex task due to the presence of extreme events. It requires the capture of long-range dependencies and the modeling of rare but important extreme values. Existing approaches often struggle to tackle these dual challenges simultaneously. In this paper, we specifically delve into these issues and propose Distance-weighted Auto-regularized Neural network (DAN), a novel extreme-adaptive model for long-range forecasting of stremflow enhanced by polar representation learning. DAN utilizes a distance-weighted multi-loss mechanism and stackable blocks to dynamically refine indicator sequences from exogenous data, while also being able to handle uni-variate time-series by employing Gaussian Mixture probability modeling to improve robustness to severe events. We also introduce Kruskal-Wallis sampling and gate control vectors to handle imbalanced extreme data. On four real-life hydrologic streamflow datasets, we demonstrate that DAN significantly outperforms both state-of-the-art hydrologic time series prediction methods and general methods designed for long-term time series prediction.

Introduction

Time series forecasting has a critical role in diverse domains, enabling effective resource management and informed policy decisions. However, certain time series data pose a unique problem because they contain sporadic but significant extreme events, such as unexpected flash floods or climate change-induced droughts in the problem of streamflow prediction. The ability to forecast time series that include these types of extreme occurrences is an important research direction which has seen much attention in recent years (Nguyen and Chan 2004; Ding et al. 2019; Qi and Majda 2020; Chen et al. 2020; Zhang et al. 2021; Li, Xu, and Anastasiu 2023a).

Traditionally, machine learning and statistics-based models were the basic foundation for time series prediction (Box and Pierce 1970; Nielsen 2019). However, techniques like Autoregressive Integrated Moving Average (ARIMA) (Box and Jenkins 1976) seem to perform badly when dealing with large variations in the streamflow values, while other methods (Shortridge, Guikema, and Zaitchik 2016; Papacharalampous and Tyralis 2022; Cheng et al. 2020) are generally designed for short future horizon forecasting.

A variety of neural network architectures have been investigated for hydrologic forecasting, including recurrent neural networks (Lai et al. 2018; Siami-Namini, Tavakoli, and Namin 2018), hybrid networks (Oreshkin et al. 2019) and graph neural networks (Wu et al. 2020; Cao et al. 2020). Some work employed Extreme Value Theory (EVT) to enhance the hydrologic time series performance (Li, Xu, and Anastasiu 2023a; Zhang et al. 2021). However, these studies primarily concentrate on short-term forecasting and their performance on longer time horizons is doubtful. While there has been a surge in transformer-based forecasting models asserting their high-performance capabilities for long-horizon time series tasks (Li et al. 2019; Qin et al. 2017; Zhou et al. 2021, 2022; Kitaev, Kaiser, and Levskaya 2020), recent research has raised questions about their efficacy, indicating that simpler linear models can outperform them (Zeng et al. 2022; Das et al. 2023). Moreover, imbalanced data or severe events might hurt all these state-of-art deep learning approaches when it comes to long-term predictions.

We focus on these challenges and innovatively address them through representation learning (Fortuin et al. 2018; Lei et al. 2017; Tonekaboni et al. 2022), a burgeoning field in unsupervised learning. Our aim is to extract latent states containing the extreme features in data for downstream tasks. To achieve this, we explore the potential of multi-loss functions (Ma et al. 2021, 2018) in shaping our training objective. The main contributions of this work are as follows:

- We propose a **D**istance-weighted **A**uto-regularized **N**eural network (DAN), which uses expandable blocks to dynamically facilitate long-term prediction.
- To improve the model's robustness to severe events, DAN innovatively uses a distance-weighted multi-loss method to extract the polar representations from time series simultaneously.
- We introduce a Kruskal-Wallis sampling policy to handle imbalanced extreme data and gate control vectors to boost the discriminatory capacity of indicators to accommodate imbalanced data.
- We evaluate DAN and competing methods on four separate datasets and find that DAN significantly outperforms

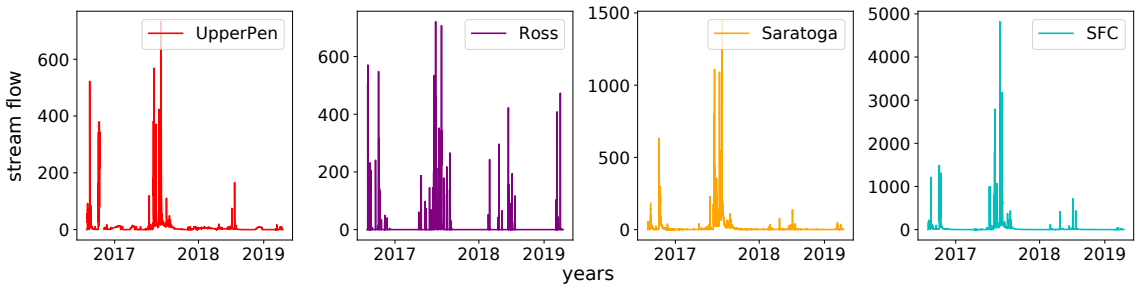

Figure 1: Streamflow over 3 years.

state-of-the-art baselines. Additionally, we carry out several ablation studies to comprehend the effects of specific design decisions.

Related Work

Streamflow forecasting holds a pivotal role in enhancing water resource allocation, management, flood warning, and mitigation of flood-related damages. Traditional methods for streamflow forecasting included 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) (Box and Jenkins 1976) method and its several variants. Wang et al. (Wang, Qiu, and Li 2018) developed a hybrid model combining Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD) and ARIMA for long-term streamflow forecasting, but they did not examine the effectiveness of their models on datasets with extreme values.

Recently, deep learning models have emerged as the preferred approach for forecasting rich time series data (Sen, Yu, and Dhillon 2019), outperforming classical statistical approaches such as ARIMA or GARCH (Box et al. 2015). Boris et al. (Oreshkin et al. 2019) proposed NBeats, which shows good performance on general time series prediction (Salinas et al. 2020). DeepAR (Salinas et al. 2020) learns a conditional distribution over the future values and uses the shared RNN to predict future values and their confidence. To tackle the long-term forecasting challenge, some recent transformer-based methods like Autoformer (Wu et al. 2021) and Reformer (Kitaev, Kaiser, and Levskaya 2020) have been proposed to empower the transformer with more sophisticated dependency discovery and representation ability. Informer (Zhou et al. 2021) proposed a ProbSparse self-attention mechanism and a generative style decoder which drastically improves the inference speed of long-sequence predictions. FEDFormer (Zhou et al. 2022) represents time series by randomly selecting a constant number of Fourier components to maintain the global property and statistics of time series as a whole.

On the other hand, many generic time series prediction models can perform poorly on data with high skewness and kurtosis scores. Conventional methods often falter when confronted with extreme events, which, although infrequent, hold considerable real-world implications—such as in specific instances of streamflow forecasting. Singh et al. (Singh, Ranjan, and Tiwari 2022) proved machine learning approaches suffer from the problem of imbalanced data distribution and noted that balancing the dataset is an imperative sub-task. An and Cho (An and Cho 2015) proposed an anomaly detection method using the reconstruction probability, which is a probabilistic measure that takes into account the variability of the data distribution. Ding et al. (Ding et al. 2019) explored the central theme of improving the ability of deep learning time series models to capture extreme events. Zhang et al. (Zhang et al. 2021) proposed a framework to integrate machine learning models with anomaly detection algorithms. In an earlier work, we proposed NEC+ (Li, Xu, and Anastasiu 2023a), a model specifically designed to provide good prediction performance on hydrological time series with extreme events. Additionally, concurrently with this work, we recently developed SEED (Li, Xu, and Anastasiu 2023b), a Segment-Expandable Encoder-Decoder model for univariate streamflow prediction.

Few of these prior works have concentrated on addressing both prolonged sequences and extreme occurrences. To bridge this gap, the proposed method, DAN, employs joint learning of two polar hidden spaces within a single series, guided by an associated distance-regularized loss function. This combined approach facilitates accurate end-to-end structure prediction. Remarkably, DAN's performance surpasses that of state-of-the-art methods across four real-life streamflow datasets.

Preliminaries

Problem Statement

Suppose we have a collection of m ($m >= 1$) related univariate time series, with each row corresponding to a different time series. We are going to predict the next h time steps for the first time series x_1, given historical data from multiple length-t observed series. 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 in our method.

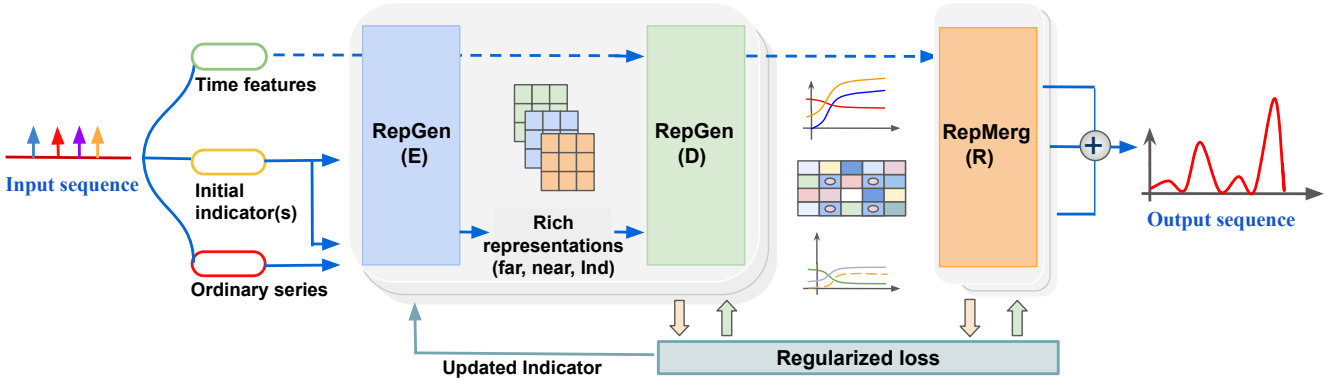

Figure 2: DAN's end-to-end extendable framework consists of two stages, named RepGen and RepMerg, respectively. RepGen contains three parallel encoder-decoder blocks, resulting in polar representations of ordinary series inputs and refined indicators. These elements are further merged in the RepMerg stack.

We first define this task by modeling the objective time series x_1 as the *ordinary series* and the group of related time series x_2 to x_m as *extraordinary indicators*. When an extraordinary indicator series is not available, our proposed model can generate a Gaussian Mixture Model (GMM) indicator based on x_1, which becomes x_2. In such cases, the problem can be reduced to that of univariate time series forecasting.

GMM Indicator

In our work, when there is no extraordinary indicator series provided, we use a Gaussian mixture model (GMM) (Day 1969) to learn a group of distributions from the input univariate time series. Then, we compute an indicator feature for each value in the time series as the weighted sum of all component probabilities, based on the weights learned during GMM model fitting. Due to lack of space, we detail this step in Section 4 of our technical appendix in (Li and Anastasiu 2023).

Kruskal-Wallis Test in Time Series

To balance the sparse distribution of extreme events, we employ the Kruskal-Wallis test (McKight and Najab 2010) as a non-parametric method to evaluate the normality of a training sample and guide our oversampling policy. The Kruskal-Wallis test examines two or more groups of time series based on their medians, in which the data are first ranked, and the sum of ranks is calculated for each group. The H value is then calculated from these rank sums, and compared to a critical value to determine if there are significant differences between the groups. Because the Kruskal-Wallis test does not assume a particular distribution, it is sometimes referred to as a distribution-free test (Ostertagova, Ostertag, and Kováč 2014; Breslow 1970). The H value is computed as

$$H = \frac{12}{n(n+1)} \sum_{j=1}^{k} \frac{R_j^2}{n_j} - 3(n+1). \quad (1)$$

In our work, we separate the sampled training sequence into k sub sequence groups of equal length. H is the Kruskal-Wallis test statistic, n is the total number of samples across all groups, R_j is the sum of ranks for the jth group, and n_j is the number of samples in the jth group. The oversampling policy will be described in the Kruskal-Wallis Sampling Section.

Methods

An overview of our architecture has been presented in Figure 2.

Polar Representation Learning

The key innovations in DAN are new mechanisms to generate and exchange information among polar (*far* and *near*) representations and the indicators (*ind*) for direct improvement of the predictions. Polar representation learning in RepGen allows for the separate encoding of extreme points, which are then preserved in RepMerg. This ensures that, during training, these representations are not compromised by the predominance of normal values, as they often adhere to different distributions. As described in Figure 3, RepGen consists of three encoder-decoder sub-networks. Please note that the CONV-LSTM layers are specifically designed for learning to predict far points. Because the hidden state for extreme events may be updated multiple times in repeated blocks, we use convolution operations to shorten the input sequence, which helps alleviate any potential exploding or vanishing gradient problem. The *far* and *near* point predictions are extended into the RepMerg stack and further refined as $\hat{y}_f$ and $\hat{y}_n$, respectively, after being added to the prior output of RepGen. To reflect the change of predicted values, $\hat{y}_i$ is expected to converge to the first order of y.

Architecture Items

Representation Merge In the RepMerg stack, to finish representation merging, the middle LSTM-FC block in the red circle takes the far point representation h_{far} as the initial hidden state and $\hat{y}_i$ as input, which is then combined with $\hat{y}_n$ to form a two-dimensional vector as input for another FC layer that outputs $\hat{y}$.

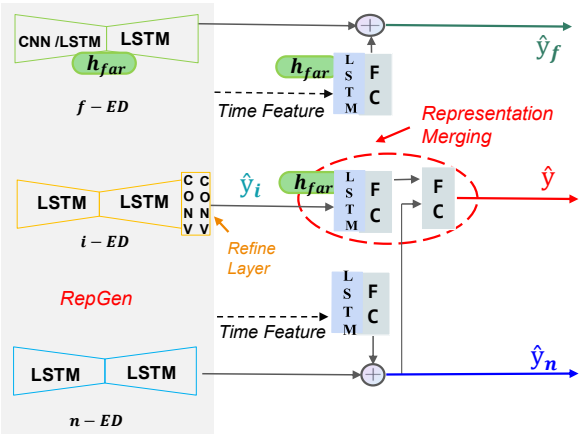

Figure 3: In the RepGen stack, "f-ED" is responsible for the representation learning of those points that are *far* away from the mean of the series ($\hat{y}_f$), including the extreme values sparsely distributed in the predicted zone. "n-ED" mainly focuses on learning hidden features of *near* points ($\hat{y}_n$), which include most of the normal values. "i-ED" is designed to learn the indicator representation ($\hat{y}_i$).

Indicator Refine Layer It should be highlighted that precise $\hat{y}_i$ prediction is important for performance improvement as it can help predict extreme events. Therefore, an additional refine layer made of double CNN components is intended to assist in producing the expected indicator by first refining the value within a local horizon. The indicator input can then be cyclically updated from the output of the refine layer of the "i-ED" as the RepGen is designed to be an expandable stack that can be repeated multiple times.

Gate Control Vector Given the significance of $\hat{y}_i$, we provide another way to hone the predicted indicator. As shown in Figure 3, we designed a gate control vector m_{far}, whose values reflect precisely in which places $\hat{y}$ is closer to $\hat{y}_f$. Similarly, m_{near} is the complement of m_{far}. We then compute $\hat{y}_w$ as $m_{far} \odot \hat{y}_f + m_{near} \odot \hat{y}_n$, where $\odot$ is the component-wise multiplication. We therefore use $\hat{y}_w$ to increase the discriminatory ability of $\hat{y}_i$ when it is expected to reflect better extreme values without sacrificing the overall normal values.

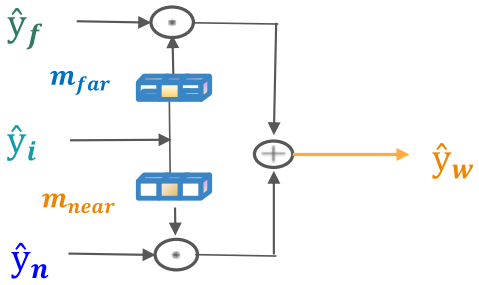

Figure 4: Gate control vector. m_{far} is equal to $sigmoid(\alpha\hat{y}_i)$, where α is an amplifier and equals 4 in our experiments; m_{near} is computed as $1 - m_{far}$.

.

Auto-Regularized Loss Function

Different from the conventional usage of regularization loss as a penalty term for preventing overfitting of the model to the training data, our approach employs multiple distance-weighted loss functions when training the DAN model, with the objective of compelling the model to learn more informative representations. Moreover, it should be noted that our method can also serve as an effective regularizer for preventing overfitting of the model to the base normal values in the long-term time series prediction task.

We define $w_f = (\tanh(y))^2$ to be a weight that emphasizes the accuracy of points further away from the mean value of 0 (since the series were standardized to have 0 mean). In contrast, $w_n = (1 - |\tanh(y)|)^2$ focuses more on the points closer to zero. The square root of w_f, denoted by w_p, is a more moderate way to maintain discriminatory output for indication-related tasks. These weight definitions contribute to the accuracy of the model in predicting extreme events in a long-term time series. Based on these weights, we build our multi regulation loss as follows,

$$\mathcal{L}_1 = RMSE((\hat{y}_f \odot w_f), (y \odot w_f)),$$
$$\mathcal{L}_2 = RMSE((\hat{y}_n \odot w_n), (y \odot w_n)),$$
$$\mathcal{L}_3 = RMSE(\hat{y}_w \odot w_p, y \odot w_p),$$
$$\mathcal{L}_4 = RMSE(\hat{y}_i \odot w_p, y_i \odot w_p),$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are used to regulate the bipolar representation learning and $\mathcal{L}_3$ and $\mathcal{L}_4$ force the predicted indicator to reflect the change of predicted values by setting y_i equal to the first order of y. Then, the overall loss is composed as,

$$\mathcal{L} = RMSE(\hat{y}, y) + \lambda \times (\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4),$$

where λ is a multiplier ($\lambda = max(-1 \cdot e^{\frac{epoch}{45}} + 2, 0.2)$ in our experiments) applied on those regulation items, decreasing with each epoch.

Kruskal-Wallis Sampling

Given that extreme events are rare within our data compared with normal ones, we utilize Kruskal-Wallis sampling to oversample regions with extreme events in our training set that our model can learn appropriate patterns from. Namely, for each random sample x of size $t + h$ we draw from the input sequence, we first split the sequence into k consecutive subsequences of equal size and compute the Kruskal-Wallis test statistic H between the k sub-sequences, using Equation 1. To avoid the H statistic being affected by minor differences in the sub-sequences, we round values in x to the nearest integer before computing H. We then include the sample in the training set if $H > \epsilon$, where ϵ is a threshold, or otherwise include the sample with probability $p < 1$. The threshold ϵ allows us to set the relative change in the sample that makes it more likely to contain an extreme event, while the probability p allows us to choose how many normal samples should be included in the training set.

Evaluation

Code and data for DAN can be found at https://github.com/davidanastasiu/dan. In this section, we present empirical results from evaluating our proposed framework. We are interested in answering the following research questions with

regards to prediction effectiveness: (1) How does DAN compare against state-of-the-art baselines? (2) What is the effect of DAN's extensible framework? (3) What is the effect of the Kruskal-Wallis oversampling policy? (4) How do the critical design elements of the framework affect performance?

Experimental Settings

Dataset We used four groups of hydrologic datasets from Santa Clara County, CA, namely Ross, Saratoga, UpperPen, and SFC, named after their respective locations. Each group included a streamflow dataset and an associated rainfall dataset. Statistics of our primary series are shown in Table 1. Our task was to forecast the streamflow for wet seasons in a hydrologic year, excluding the summer months, namely September 2021 to May 2022, in a rolling manner. The training and validation sets were randomly sampled from series spanning from January 1988 to August 2021. Inference involved predicting the streamflow every 4 hours for the next 3 days. Since the sensors measure the streamflow and precipitation every 15 minutes, we are attempting a lengthy forecasting horizon ($h = 288$), which is unquestionably an LSTF task based on the most recent research (Zhou et al. 2021, 2022; Zeng et al. 2022; Das et al. 2023). Before training a model, all time series were pre-processed by log transform ($x_i = \log(1 + x_i) \forall i$) and standardization (subtract mean and divide by standard deviation). Inference predictions were post-processed by inverting the standardization and log transform operations.

	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
skewness	19.84	19.50	13.42	18.05
kurtosis	523.16	697.78	262.18	555.18

High skewness and kurtosis scores indicate that there is a significant deviation from a normal distribution in our data.

Table 1: Streamflow Datasets Statistics

Model Parameters For all models, after testing different LSTM layer widths, we found that 512 node layers for ROSS and 384 layers for the other three sensors were the most effective. We set $h = 288$ (3 days) based on the problem definition and tested different values of $t \in \{288, 672, 1440\}$ (3, 7, 15 days), with $t = 1440$ producing the best results for all data streams. In the RepGen stage, the three CNN layers produce 256 channels each. The kernel sizes used in these layers are 11, 7, and 3, respectively. The stride, padding, and activation function remain the same across all three layers, with a stride equal to the kernel size, no padding, and a subsequent $tanh$ activation function. We use two stacked CNN1d layers for indicator refinement, with the kernel size and padding set to 7 for the first layer and 3 for the second.

Experimental Results

Baselines We include seven state-of-the-art models for comparison, of which FEDFormer (Zhou et al. 2022), Informer (Zhou et al. 2021), NLinear (Zeng et al. 2022), and DLinear (Zeng et al. 2022) focus on long-term time series forecasting, while NEC+ (Li, Xu, and Anastasiu 2023a) holds the best performance for hydrologic time series prediction in the presence of extreme events. These five models were used as baselines for both multivariate and univariate prediction. In addition, Attention-LSTM (Le et al. 2021) was used as a state-of-the-art hydrologic multivariate model used to predict stream-flow using rainfall data. Finally, N-BEATS (Oreshkin et al. 2019), a state-of-the-art univariate baseline method that outperformed all competitors on the standard M3 (Makridakis and Hibon 2000), M4 (Makridakis, Spiliotis, and Assimakopoulos 2018), and TOURISM (Athanasopoulos et al. 2011) time series datasets, was also used in the comparison.

Multivariate and Univariate Results Table 2 shows the test root mean squared error (RMSE) and mean absolute percentage error (MAPE) performance for the models that achieved the best performance on our validation dataset. For these experiments, all DAN results were achieved using the same random seed. Due to lack of space, we include the definition of our performance metrics and multiple seed run performance statistics in Section 5 of our technical appendix in (Li and Anastasiu 2023). In the multivariate forecasting task, our proposed model DAN outperformed all baselines on all four benchmark datasets. Compared to the second-best results, DAN achieved an overall 19% relative RMSE reduction. Notably, the improvement was most significant for the UpperPen dataset, where DAN achieved a 30% improvement. For the univariate forecasting task, DAN outperformed other methods on three out of four benchmark datasets. Although NBeats achieved a 6% relative RMSE reduction compared to DAN for the UpperPen sensor, DAN surpassed NBeats with a relative RMSE reduction of 18%, 49%, and 52% on the Ross, Saratoga, and SFC datasets, respectively.

Inference Overall Analysis Figure 5, in which we present rolling prediction results for the whole test set (1600 time points) for the Ross sensor, helps explain DAN's good performance. To make it easier to visualize, the test set was sampled every 5 points (320 time points sampled) and a specific period including extreme events is denoted by the red box and focused on in the right figure. We observed that DAN performed better than other models on areas with extreme events. While Informer and NLinear could follow the extreme events better than other baselines, they predicted values much higher than the actual peaks. On the other hand, DLinear and Attention-LSTM performed better than NEC+ and FEDFormer, but they predicted values much lower than the ground truth.

Ablation Studies

The superior performance of our method comes from its extendable framework, creative Kruskal-Wallis sampling policy, and three key architecture designs: 1) the use of a refine layer in RepGen that refines the indicator information using double CNNs with moving kernel convolution operations, 2) the

Methods	Metric	Ross		Saratoga		UpperPen		SFC	
		Multi	Single	Multi	Single	Multi	Single	Multi	Single
FEDformer	RMSE	6.01	6.49	6.01	6.85	3.05	2.38	23.54	24.10
	MAPE	2.10	2.49	1.55	2.26	1.87	1.02	2.35	2.817
Informer	RMSE	7.84	9.14	5.04	4.89	5.88	5.33	39.89	19.00
	MAPE	4.05	5.45	1.43	0.73	4.10	4.21	8.64	0.54
Nlinear	RMSE	6.10	5.84	5.23	4.98	1.57	1.74	18.47	18.43
	MAPE	1.99	1.62	0.83	0.75	0.45	0.57	0.92	0.87
Dlinear	RMSE	7.16	6.90	4.33	4.06	3.53	3.25	21.62	23.64
	MAPE	3.10	2.79	1.40	1.31	2.35	2.04	2.74	4.02
NEC+	RMSE	9.44	9.33	1.88	1.95	2.22	1.94	17.00	16.39
	MAPE	4.80	4.53	0.17	0.21	0.95	0.80	1.07	0.55
LSTM-Atten / NBeats	RMSE	7.35	5.16	6.49	3.60	6.35	**1.23**	34.17	31.47
	MAPE	3.74	1.25	1.80	0.70	4.76	**0.25**	9.90	3.24
DAN	RMSE	**4.25**	**4.24**	**1.80**	**1.84**	**1.10**	1.31	**15.23**	**15.20**
	MAPE	**0.07**	**0.09**	**0.14**	**0.16**	**0.15**	0.32	**0.26**	**0.21**

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.

Table 2: Multivariate/Univariate Long-Term ($h = 288$) Series Forecasting Results on Four Datasets

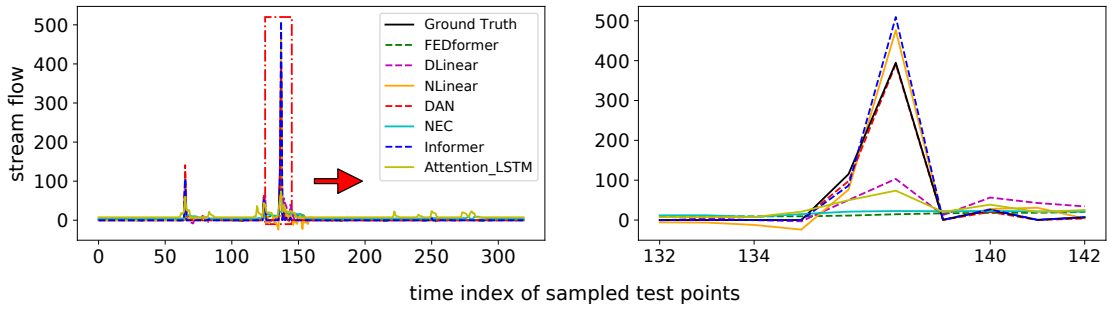

Figure 5: Sampled multivariate inference for the Ross sensor. The right figure emphasizes extreme events occurring during the example time period in the red box of the left figure.

RepMerg layer which combines the rich polar representations with the indicator learned from RepGen, and 3) regularization loss components. We will examine their effects respectively in this section.

Effects of DAN's Extensible Framework Our network comprises two main stages. The repeatable stacks are named "E", "D" and "R" in Figure 2, endowing DAN with the capability of updating the indicator in an evolutional way by repeating "E"+"D" and including multiple refinements by repeating "R". The best stack configuration may vary for different datasets, depending on the intrinsic relationships in the multivariate series. We experimented with various combinations and identified the best results as "EDEDRR", "EDR", "EDEDRR", and "EDEDR" for Ross, Saratoga, UpperPen, and SFC, respectively.

Effects of the Oversampling Policy Figure 6 displays the distribution of the H values before and after applying the Kruskal-Wallis sampling algorithm. These observations highlight the impact of adjusting p and ϵ on the distribution of H values in the training set. If we maintain the p value and increase the ϵ value, the training set will contain more samples with H values exceeding ϵ, as illustrated in the rightmost three figures in the first row of Figure 6.

Policy($\epsilon = 10$)	Ross	SFC
$20\% <= p <= 40\%$	**23.5**	**118.1**
$60\% <= p <= 80\%$	25.0	122.0
$p = 100\%$	28.7	129.0

We used ϵ = 10 and the p values used were grouped into 3 cases. $p = 100\%$ means no Kruskal-Wallis oversampling was applied.

Table 3: $RMSE_{far}$ when Oversampling

Our oversampling policy can help shift the focus of our model towards improving "far" point prediction performance. To test this, we conducted multiple runs of each model, aver-

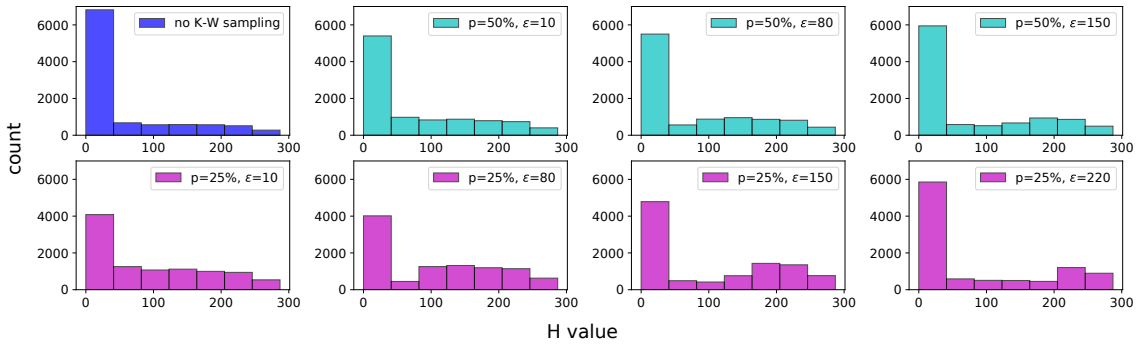

Figure 6: Oversampling policy. Initially, a majority of the samples have H values below 10. However, upon setting $p = 50\%$ and $\epsilon = 10$, the number of samples with H values below 10 decreases, while the count for higher H values increases. The second row showcases the scenario for $p = 25\%$.

aging the RMSE values for points with values greater than 1.5 standard deviations above the mean of the series, which we call $RMSE_{far}$ in Table 3. The results show that the $RMSE_{far}$ can be steadily decreased by decreasing p.

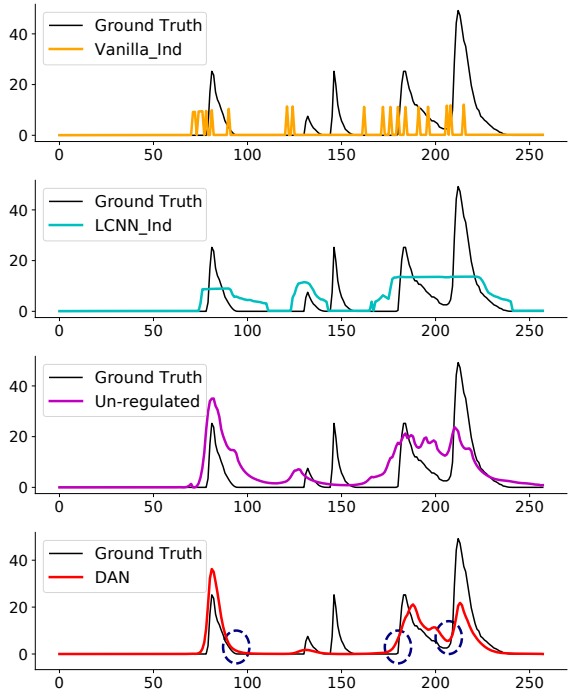

Figure 7: Inference examples to show the effects of different architecture elements on the Ross dataset.

Effects of DAN Architecture Elements In this section, we investigate the impact of different design elements on the performance of our network. To isolate the effects of these design elements and obtain a more comprehensive understanding of each one, we replace the input indicator in our network with ground truth rain data and only use a simple EDR architecture stack, which will remove the effect of varying indicator performance from the comparison. By doing so, we can observe how the network performs when certain layers are removed or added back.

To create a baseline structure, we first removed the refine and the RepMerge layers, using only the encoder-decoder block to generate the indicator, and set $\hat{y}$ equals to $\hat{y}_w$ to bypass the RepMerge structure. This produced the first predicted sequence (named **Vanilla_Ind** in Figure 7), which shows an example of inference on Ross sensor data. We then added the refine layer back, which improved the results as shown in the second sub-figure (named **LCNN_Ind**). Next, we added back the RepMerg layer. Concurrently, we also removed the regularization loss items except $\mathcal{L}_3$, obtaining the third figure (named **Un-regulated**). Finally, we added all regularization loss items back, which gave the best result, as shown in the fourth figure.

These experiments demonstrate that the CNN-FC with moving kernel convolutional operations refines the indicator information, and RepMerg produces better results as the Gate control vector mechanism increases the discrimination of predicted values. Adding polar representation of the basic series assists in identifying data distributions beyond the indicator information and enhances the accuracy of data at corners of each fluctuation, as denoted in the blue circles in Figure 7. Therefore, including the refine layer, representation learning, and gate control vector resulted in the best performance.

Conclusion

In this work, we presented a novel end-to-end framework, DAN, designed to better account for rare yet important extreme events in long single- and multi-variate streamflow time series. Our framework learns polar representations for predicting extreme and normal values, along with a representation merging model that makes prediction in an expandable way. In addition, to improve training performance, our framework uses Kruskal-Wallis sampling policies to accommodate imbalanced extreme data and a distance-weighted multi-loss regularization penalty. Extensive experiments using more than 33 years of streamflow data from Santa Clara County, CA, showed that DAN provides significantly better predictions than state-of-the-art baselines.

References

An, J.; and Cho, S. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1): 1–18.

Athanasopoulos, G.; Hyndman, R. J.; Song, H.; and Wu, D. C. 2011. The tourism forecasting competition. *International Journal of Forecasting*, 27(3): 822–844.

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

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Box, G. E.; and Pierce, D. A. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332): 1509–1526.

Breslow, N. 1970. A generalized Kruskal-Wallis test for comparing K samples subject to unequal patterns of censorship. *Biometrika*, 57(3): 579–594.

Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33: 17766–17778.

Chen, Z.; Yu, H.; Geng, Y.-a.; Li, Q.; and Zhang, Y. 2020. EvaNet: An extreme value attention network for long-term air quality prediction. In *2020 IEEE International Conference on Big Data (Big Data)*, 4545–4552. IEEE.

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

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.

Day, N. E. 1969. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3): 463–474.

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

Fortuin, V.; Hüser, M.; Locatello, F.; Strathmann, H.; and Rätsch, G. 2018. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199*.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Le, Y.; Chen, C.; Hang, T.; and Hu, Y. 2021. A stream prediction model based on attention-LSTM. *Earth Science Informatics*, 14: 1–11.

Lei, Q.; Yi, J.; Vaculin, R.; Wu, L.; and Dhillon, I. S. 2017. Similarity preserving representation learning for time series clustering. *arXiv preprint arXiv:1702.03584*.

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

Li, Y.; and Anastasiu, D. C. 2023. Learning from Polar Representation: An Extreme-Adaptive Model for Long-Term Time Series Forecasting. arXiv:2312.08763.

Li, Y.; Xu, J.; and Anastasiu, D. C. 2023a. An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Li, Y.; Xu, J.; and Anastasiu, D. C. 2023b. SEED: An Effective Model for Highly-Skewed Streamflow Time Series Data Forecasting. In *2023 IEEE International Conference on Big Data (Big Data)*, IEEE BigData 2023. Los Alamitos, CA, USA: IEEE Computer Society.

Ma, H.; Zhang, Z.; Li, W.; and Lu, S. 2021. Unsupervised human activity representation learning with multi-task deep clustering. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1): 1–25.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

Makridakis, S.; and Hibon, M. 2000. The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4): 451–476. The M3- Competition.

Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2018. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4): 802–808.

McKight, P. E.; and Najab, J. 2010. Kruskal-wallis test. *The corsini encyclopedia of psychology*, 1–1.

Nguyen, H. H.; and Chan, C. W. 2004. Multiple neural networks for a long term time series forecast. *Neural Computing & Applications*, 13: 90–98.

Nielsen, A. 2019. *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly Media.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Ostertagova, E.; Ostertag, O.; and Kováč, J. 2014. Methodology and application of the Kruskal-Wallis test. In *Applied mechanics and materials*, volume 611, 115–120. Trans Tech Publ.

Papacharalampous, G.; and Tyralis, H. 2022. A review of machine learning concepts and methods for addressing challenges in probabilistic hydrological post-processing and forecasting. *Frontiers in Water*, 4: 961954.

Qi, D.; and Majda, A. J. 2020. Using machine learning to predict extreme events in complex systems. *Proceedings of the National Academy of Sciences*, 117(1): 52–59.

Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; and Cottrell, G. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191.

Sen, R.; Yu, H.-F.; and Dhillon, I. S. 2019. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32.

Shortridge, J. E.; Guikema, S. D.; and Zaitchik, B. F. 2016. Machine learning methods for empirical streamflow simulation: a comparison of model accuracy, interpretability, and uncertainty in seasonal watersheds. *Hydrology and Earth System Sciences*, 20(7): 2611–2628.

Siami-Namini, S.; Tavakoli, N.; and Namin, A. S. 2018. A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 1394–1401. IEEE.

Singh, A.; Ranjan, R. K.; and Tiwari, A. 2022. Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, 34(4): 571–598.

Tonekaboni, S.; Li, C.-L.; Arik, S. O.; Goldenberg, A.; and Pfister, T. 2022. Decoupling local and global representations of time series. In *International Conference on Artificial Intelligence and Statistics*, 8700–8714. PMLR.

Wang, Z.-Y.; Qiu, J.; and Li, F.-F. 2018. Hybrid Models Combining EMD/EEMD and ARIMA for Long-Term Streamflow Forecasting. *Water*, 10(7).

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. 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*, 753–763.

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

Zhang, Y.; Li, J.; Carlo, A.; Manda, A. K.; Hamshaw, S.; Dascalu, S. M.; Harris, F. C.; and Wu, R. 2021. Data Regression Framework for Time Series Data with Extreme Events. In *2021 IEEE International Conference on Big Data (Big Data)*, 5327–5336.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.