SimEXT: Self-supervised Representation Learning for Extreme Values in Time Series

Asadullah Hill Galib[1], Pang-Ning Tan[1], and Lifeng Luo[2]
[1]*Dept of Computer Science and Engineering, Michigan State University*
[2]*Dept of Geography, Environment, and Spatial Sciences, Michigan State University*
Emails: {galibasa, ptan, lluo}@msu.edu

***Abstract*—Forecasting extreme values in time series is an important but challenging problem as the extreme values are rarely observed even when a large amount of historical data is available. The modeling of extreme values requires a specific focus on estimating the tail distribution of the time series, whose statistical properties may differ from the distribution of its non-extreme values. To overcome this challenge, we present a novel self-supervised learning framework, `SimEXT`, to learn a robust representation of the time series that preserves the fidelity of its tail distribution. The framework employs a combination of contrastive learning and a reconstruction-based autoencoder architecture to facilitate robust representation learning of the temporal patterns associated with the extreme events. `SimEXT` also incorporates a wavelet-based data augmentation technique with a distribution-based loss function to prioritize the learning of extreme value distribution. We provide probabilistic guarantees on the wavelet-based augmentation that enables the wavelet coefficients to be perturbed during data augmentation without significantly altering the extreme values of the time series. Experimental results on real-world datasets show that `SimEXT` can effectively learn a robust representation of the time series to boost the performance of downstream tasks for forecasting block maxima values.**

***Index Terms*—forecasting; time series, extreme values**

I. Introduction

Deep time series forecasting models are widely used to predict the future outcomes of complex processes that evolve over time. The accuracy and effectiveness of these models often depend on their ability to discern the underlying patterns of the data and and using them to predict the time series' future evolution. A critical element in time series forecasting is predicting extreme values, which are values significantly outside the usual range. This is crucial in various domains, as extreme events can signify dire scenarios like natural disasters, financial crises, or public health risks.

Block maxima or minima [1] are commonly used to define extreme values in a time series. These definitions involve dividing a time series into non-overlapping blocks of a fixed period and identifying the maximum or minimum value within each block. Alternatively, extreme values can be defined as excess values over a user-specified threshold. In this paper, we focus on block maxima (or minima) as extreme values due to their critical significance for anticipating worst-case scenarios during forecast periods. For instance, predicting the maximum intensity of an upcoming hurricane or amplitude of seismic activity for a future time window can assist emergency planners in assessing its potential damages.

Accurate forecasting of extreme values in time series is challenging for several reasons. Firstly, it necessitates a focused approach on modeling the tail distribution [1], deviating significantly from conventional techniques that typically emphasize on modeling the conditional mean. Secondly, the rarity of extreme values compounds the difficulty of prediction, even with abundant historical data. Finally, the extreme values could be associated with certain peculiarities in the time series, such as abrupt changes, volatility clustering, persistent dependencies, etc [2], [3]. Advanced representation learning approach is therefore needed to learn the underlying patterns in the time series that can be utilized for extreme value forecasting.

Self-supervised learning (SSL) [4], [5] is an emerging machine learning technique that fosters robust feature representation learning despite data limitations. Self-supervised contrastive learning [4] employs data augmentation to address labeled data scarcity issues, comparing augmented versions of the same input to learn a robust representation that is invariant to changes introduced by the augmentation. For time series, it facilitates learning representations invariant to time shifts, scaling, or warping [6], thereby improving generalizability of the model. SSL also excels in capturing complex patterns and non-linear dependencies in time series [7].

Existing SSL approaches, although promising for extracting meaningful time series representations [4], [5], often prioritize common patterns over extreme values. Furthermore, current data augmentation methods can inadvertently distort extreme values, compromising the fidelity of the tail distribution in learned representations. To ensure robustness across scenarios and the ability to capture extreme event characteristics, it is essential to develop data augmentation techniques that account for extreme values when transforming time series data for SSL.

To address these challenges, we propose a SSL framework called `SimEXT` to learn a feature representation that captures the extreme values of a time series. `SimEXT` leverages contrastive learning with a reconstruction-based autoencoder architecture. A novel wavelet-based data augmentation technique is also introduced to ensure that the extreme values are not significantly altered during augmentation. Additionally, we investigate two distribution loss functions that emphasize on learning features that preserve the extreme values of a time series. These learned representations are subsequently applied to downstream tasks that focused on predicting future block maxima (or minima) of the time series.

II. Preliminaries

A. Problem Statement

Consider a time series of T discrete time steps, $y_1, y_2, \cdots, y_T$, where $y_i \in \mathbb{R}$. Assume the time series is divided into a set of non-overlapping time windows, where each window w_t corresponds to the interval $[t-\alpha, t+\beta]$ and contains a segment $(y_{t-\alpha}, \cdots, y_t, \cdots, y_{t+\beta})$ of the input time series. For each window w_t, we define its *predictor window* as the interval $[t-\alpha, t]$ and its *forecast window* as the interval $[t+1, t+\beta]$. The block maxima of the time series for the forecast window of w_t is given by:

$$m_t = \max_{\tau \in [t+1, t+\beta]} y_t$$

Our objective here is two-fold: (1) to learn a robust feature representation of the time series in each predictor window, i.e., $\boldsymbol{z}_t = h_\theta(\boldsymbol{x}_t) \in \mathbb{R}^d$, where $\boldsymbol{x}_t = (y_{t-\alpha}, \cdots, y_t) \in \mathbb{R}^{\alpha+1}$ is the time series segment associated with the predictor window, $h_\theta(\cdot)$ is an encoder model that maps the input time series segment into its latent representation, and θ is the learnable parameters; and (2) to predict the block maxima, $\hat{m}_t$, of the forecast window by learning a mapping function $f_\phi(\cdot)$ such that $\hat{m}_t = f_\phi(\boldsymbol{x}_t, \boldsymbol{z}_t)$, where ϕ is the learnable parameters.

B. Contrastive Learning

Contrastive learning aims to learn feature representations in such a way that similar instances are close to each other in the latent representation space while dissimilar instances are far apart by minimizing the following NT-Xent loss [8]:

$$\ell(i,j) = -\log \frac{\exp\left[sim\left(h(\boldsymbol{x}^i), h(\boldsymbol{x}^j)\right)/\tau\right]}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp\left[sim\left(h(\boldsymbol{x}^i), h(\boldsymbol{x}^k)\right)/\tau\right]} \quad (1)$$

where $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ are two augmented views of a data instance, $\mathbb{1}_{k \neq i}$ is an indicator function, $h(\cdot)$ denotes the representation learning function, sim$(\cdot,\cdot)$ is the cosine similarity measure, and τ is the temperature hyperparameter. The final loss is computed for all positive pairs in a minibatch of size N as follows [8]:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{2N} \sum_{k-1}^{N} \left[\ell(2k-1, 2k) + \ell(2k, 2k-1)\right] \quad (2)$$

C. Discrete Wavelet Transform

The discrete wavelet transform (DWT) decomposes a time series into its approximation (scaling) and detail (wavelet) coefficients, facilitating multi-resolution analysis. Approximation coefficients convey information about the signal's overall trend (low-frequency components), while detail coefficients capture the noisy temporal variations (high-frequency components). An input signal $\boldsymbol{x}_t$ can be expressed as linear combination of the scaling functions $\phi(t)$ and wavelet functions $\psi(t)$ using the detail and approximation coefficients as follows:

$$\boldsymbol{x}_t = \sum_k c_j(k)\phi_{j,k}(t) + \sum_k \sum_j d_j(k)\psi_{j,k}(t) \quad (3)$$

where j and k are the scale and dilation parameters respectively, $c_j(k)$ denotes the approximation (scaling) coefficient, and $d_j(k)$ denotes the detail (wavelet) coefficient. For orthogonal scaling and wavelet functions, $c_j(k)$ and $d_j(k)$ can be calculated by taking their inner product with the original signal $\boldsymbol{x}_t$.

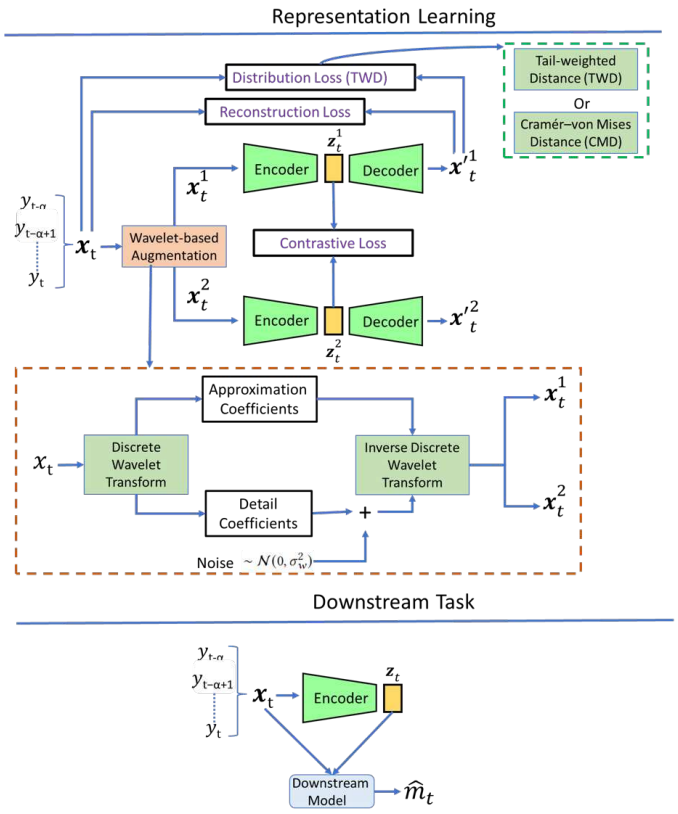

Fig. 1. An overview of the proposed SimEXT framework for time series forecasting of extreme values.

In practice, the approximation and detail coefficients can be calculated without using the scaling and wavelet functions. Instead, a cascading filter banks algorithm is employed, allowing the coefficients to be recursively computed as follows:

$$\begin{aligned} c_j(k) &= \sum_m h_{\text{lp}}(m-2k)\ c_{j+1}(m), \\ d_j(k) &= \sum_m h_{\text{hp}}(m-2k)\ d_{j+1}(m) \end{aligned} \quad (4)$$

where $m = 2k + n$ while $h_{\text{lp}}(\cdot)$ and $h_{\text{hp}}(\cdot)$ denote the low-pass and high-pass filters, respectively. Choosing the right wavelet function depends on the signal's characteristics and application goals. For instance, the Haar wavelet is preferred when rapid computation is vital. It reduces coefficients by half at each decomposition level, yielding a more compact signal representation, particularly advantageous for large datasets or real-time applications requiring computational efficiency.

III. Proposed SimEXT Framework

SimEXT combines contrastive learning with a reconstruction-based autoencoder to generate a robust latent representation of time series data. An overview of the proposed framework is shown in Fig. 1.

A. Wavelet-based Data Augmentation

Contrastive learning requires performing data augmentation to generate conceptually similar samples by perturbing the original data. For time series, the perturbation methods include jittering, flipping, shuffling, time warping, etc. Choosing the right data augmentation approach is crucial for block maxima forecasting to ensure that the learned representation preserves the overall pattern of the time series without significantly altering its block maxima values. To achieve this, the framework introduces a wavelet-based data augmentation technique. Before describing the approach, we first examine the effect of jittering on the block maxima of a time series.

Theorem 1. *Let $y_1, y_2, \cdots, y_n$ be a sequence, where $y_i \in \mathbb{R}$ and $M_n = \max_i y_i$. Assuming $\hat{M}_n = \max_i \hat{y}_i$, where $\hat{y}_i = y_i + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, it can be shown that:*

$$\mathbb{E}_\epsilon \left[\hat{M}_n - M_n\right] \leq \log n + \frac{\sigma^2}{2} \tag{5}$$

Proof. First, observe that $\mathbb{E}_\epsilon \left[\hat{M}_n - M_n\right] = \mathbb{E}_\epsilon \left[\hat{M}_n\right] - M_n$. Since the exponent function is non-negative, the expected value of $\hat{M}_n$ can be expressed as follows:

$$\begin{aligned} \mathbb{E}_\epsilon \left[\hat{M}_n\right] = \mathbb{E}_\epsilon \left[\max_i \left(\epsilon_i + y_i\right)\right] &= \mathbb{E}_\epsilon \left[\max_i \left(\log e^{\epsilon_i + y_i}\right)\right] \\ &\leq \mathbb{E}_\epsilon \left[\log \sum_{i=1}^{n} e^{\epsilon_i + y_i}\right] \end{aligned}$$

The inequality above can be further simplified using Jensen inequality as follows:

$$\mathbb{E}_\epsilon \left[\hat{M}_n\right] \leq \log \mathbb{E}_\epsilon \left[\sum_{i=1}^{n} e^{\epsilon_i} e^{y_i}\right] = \log \left(\sum_{i=1}^{n} \mathbb{E}_\epsilon \left[e^{\epsilon_i}\right] e^{y_i}\right)$$

Assuming $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, it can be shown that

$$\mathbb{E}\left[e^{\epsilon}\right] = \int_{-\infty}^{\infty} e^{\epsilon} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\epsilon^2}{2\sigma^2}} d\epsilon = e^{\frac{\sigma^2}{2}}$$

Replacing the expected value into the inequality given in (6), we obtain the following:

$$\begin{aligned} \mathbf{E}_\epsilon \left[\hat{M}_n\right] \leq \log \left(\sum_{i=1}^{n} e^{\frac{\sigma^2}{2} + y_i}\right) &\leq \log \left(\sum_{i=1}^{n} e^{\frac{\sigma^2}{2} + M_n}\right) \\ &= \log n + \frac{\sigma^2}{2} + M_n \end{aligned}$$

The proof follows by subtracting M_n from the expected value given above. □

Theorem 1 provides an upper bound on the expected value of the difference between the perturbed block maxima and the original block maxima of a time series of length n. Note that the bound is proportional to the variance of the noise as well as number of perturbed data points. Thus, by using a smaller variance during the jittering process, fewer number of perturbed data points, or both, this can help avoid altering the block maxima value significantly.

Algorithm 1 Wavelet-based Data Augmentation Algorithm

Input: Time series predictor $\boldsymbol{x_t}$ and noise variance, σ^2.
Output: Augmented time series pair, $\boldsymbol{x}_t^1$ and $\boldsymbol{x}_t^2$.

$(c_{j_0,k}, d_{j,k}) \leftarrow \mathbf{DWT}(\boldsymbol{x_t})$
for i = 1 to 2 **do**
  $\delta_{j,k}^i \leftarrow d_{j,k} + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
  $\boldsymbol{x}_t^i \leftarrow \mathbf{IDWT}(c_{j_0,k}, \delta_{j,k}^i)$
end for
return $\boldsymbol{x}_t^1$ and $\boldsymbol{x}_t^2$

Algorithm 1 summarizes the pseudocode of our proposed wavelet-based data augmentation approach. Given an input segment, $\boldsymbol{x}_t$, we first employ DWT to decompose the time series into its approximation coefficients ($c_{j_0,k}$) and wavelet (detail) coefficients ($d_{j,k}$). We apply the Daubechies 1 wavelet, also known as the Haar wavelet, in this study but the methodology is applicable to other wavelet functions. As mentioned in Section II-C, by employing the Haar wavelet, the number of detail and approximation coefficients will be halved of the length of the original time series. Since the approximation coefficient captures the overall trend while the detail coefficient captures the noisy, high-frequency components, we perform augmentation by applying jittering to the detail coefficients only. Specifically, each detail coefficient is perturbed by adding a Gaussian noise with variance σ^2. As the number of detail coefficients is half of the length of $\boldsymbol{x}_t$ and using a low variance σ^2, following Theorem 1, this ensures that the block maxima of the augmented time series is close to that of its input time series. Finally, we employ the inverse discrete wavelet transform (IDWT) on the approximation and perturbed detail coefficients to construct the augmented time series.

B. Self-supervised Representation Learning

The wavelet-based data augmentation technique described in the previous section is used to create augmented pairs of similar samples for each predictor window $\boldsymbol{x}_t$. Each augmented pair $(\boldsymbol{x}_t^1, \boldsymbol{x}_t^2)$ is passed to an autoencoder to generate its corresponding feature embedding $(\boldsymbol{z}_t^1, \boldsymbol{z}_t^2)$, capable of reconstructing the original sample: $\boldsymbol{x}_t'^i = Decoder(\boldsymbol{z}_t^i)$, where $\boldsymbol{z}_t^i = Encoder(\boldsymbol{x}_t^i)$. The framework employs a reconstruction loss based on the squared Euclidean norm between the original and augmented sample to preserve important features. Additionally, the contrastive loss (Eq. (2)) is calculated for all positive samples to ensure that the representations of similar pairs remain close. By minimizing both losses, the framework learns a robust representation of the time series.

C. Enforcing Fidelity of Tail Distribution

The reconstruction and contrastive losses alone cannot guarantee that the learned representation would preserve the fidelity of the block maxima distribution. To address this, we introduce a distribution loss in the objective function to emphasize feature learning that considers extreme values in the time series. Let $\boldsymbol{x}_t$ be the original input time series and $\boldsymbol{x}_t'$

be the reconstructed time series. Their corresponding empirical cumulation distribution functions (ECDFs) are defined as

$$F_x(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[x_{(i)} \leq z],\ F_{x'}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[x'_{(i)} \leq z] \quad (6)$$

where n is the sample size, $x_{(i)}$ is the i-th largest observation in $\boldsymbol{x}$, and $\mathbb{1}[x_{(i)} \leq z]$ is an indicator function, whose value is 1 if $x_{(i)} \leq z$ and 0 otherwise. We consider two approaches for measuring the distribution loss.

1) Tail-Weighted Distance (TWD) between Empirical Cumulative Distribution Functions (ECDFs): This loss is motivated by the Kolmogorov–Smirnov (KS) distance [9] for determining whether a sample is drawn from a particular distribution: $D_{KS}(F_{x'}, F_x) = \sup_z |F_{x'}(z) - F_x(z)|$. However, as the distance does not consider whether the extreme values of the distribution are well-preserved. we introduce the following tail-weighted distance as our distribution loss function:

$$\text{TWD} = \frac{1}{n}\sum_{j=1}^{n} \gamma(p_j) \cdot\ |F_{x'}(x_j) - F_x(x_j)| \quad (7)$$

where F_x and $F_{x'}$ are the ECDFs of the original and reconstructed time series respectively, $p_j = F_x[x_j]$ is the percentile of the j-th observation, and $\gamma(p_j) = p_j^2$ is a tail-weighted function, whose value grows quadratically with increasing p_j.

2) Cramér–von Mises Distance (CMD) between the ECDFs of GEV Distribution for Block Maxima: In this approach, we employ the Cramér-von Mises (CM) distance [10] to measure the deviation between the GEV distribution of the block maxima values in the original and reconstructed time series. Let $m = \max_{x_i \in \boldsymbol{x}} x_i$ be the block maxima of the original series and $m' = \max_{x_i \in \boldsymbol{x'}} x_i$ be the block maxima of the reconstructed series. Given a set of block maxima values generated from the predictor windows of the training data, we use the maximum likelihood approach to estimate the GEV parameters of the block maxima values. Let G_x and $G_{x'}$ be the ECDF of the fitted GEV distributions of block maxima values associated with the original and reconstructed time series, respectively. We then compute the CMD as follows:

$$\text{CMD} = \sqrt{\sum_{i=1}^{n} |G_{x'}(z_i) - G_x(z_i)|} \quad (8)$$

where $\{z_1, z_2, \cdots, z_n\}$ are the samples drawn from the fitted GEV distribution and n is the number of samples drawn.

D. Optimization

The SimEXT framework is trained to minimize the following loss function:

$$\mathcal{L} = \lambda_1\, \mathcal{L}_{contrastive} + \lambda_2\, \mathcal{L}_{recon} + \lambda_3\, \mathcal{L}_{dist} \quad (9)$$

where $\mathcal{L}_{contrastive}$ is the contrastive loss given in Eq. (2), $\mathcal{L}_{recon} = \sum_t \|\boldsymbol{x}_t - \boldsymbol{x}'_t\|_2^2$ is the reconstruction loss of the autoencoder, and $\mathcal{L}_{dist}$ is either the TWD or CMD distribution losses described in Section III-C. Note that λ_1, λ_2, and λ_3 are hyperparameters that manage the trade-off between the different components of the loss function.

IV. Experimental Evaluation

We have performed extensive experiments to evaluate the performance of our SimEXT framework.

We consider the following three datasets for our experiments. (1) **Hurricane** [11]. This dataset corresponds to wind speed values at 6-hourly intervals for 3,111 hurricanes between the years 1851 to 2019. We segmented each hurricane into non-overlapping 24-time step (6-day) time windows. The first 16 time steps (4 days) constitute the predictor window, while the last 8 time steps (2 days) form the forecast window. (2) **Climate**[1]. This dataset comprises of daily maximum temperature for 3 weather stations (Maple City, Hart, and Eau Claire) from 1978 to 1998. We segment the data into 14-day non-overlapping time windows, with the initial 7 days as the predictor window and the subsequent 7 days as the target window. (3) **ECL** [2]. This dataset comprises the hourly electricity consumption of 321 clients. We partition the time series into non-overlapping time windows of 14-day duration. The initial 7-day period is used as predictor time window, while the remaining 7-day interval defines the forecast window.

To demonstrate the efficacy of SimEXT, we conducted a comparative analysis against the following state-of-the-art time series representation learning methods: (1) **CoST** [12], (2) **TS2Vec**, (3) **TNC**, and (4)**TimeCLR** [13]. The experiments were conducted using the following downstream models: (1) **LSTM**, (2) **Transformer**, (3) **Informer** [14], (4) **EVL** [15], and (5) **DeepExtrema** [16]. Furthermore, to understand the efficacy of the features generated by our proposed SimEXT framework, we also compare its performance under the following experimental settings: (1) **Original Features**: The original features of the time series are directly fed into the downstream forecasting models without any transformation. (2) **AE**: This setting uses the autoencoder module only to learn the representation. (3) **AE + CL**: This setting incorporates both the autoencoder and contrastive learning modules. (4) **AE + CL + TWD**: This extends the previous setting by incorporating the TWD distribution loss. (5) **AE + CL + CMD**: This is similar to previous setting except is uses the CMD distribution loss.

A. Experiment Settings

We partitioned each dataset into training, validation, and testing, according to an 8:1:1 ratio. We repeated the experiments 5 times using different partitioning of the data. Data is standardized to have zero mean and unit variance. Our framework employs a 4-layer bidirectional LSTM architecture for the encoder and decoder components. The training was facilitated using the Adam optimizer. We conducted hyperparameter tuning for all methods using the Ray Tune framework with an Asynchronous Successive Halving Algorithm (ASHA) scheduler for early stopping. We evaluated the framework's performance using: (1) Root Mean Squared Error (RMSE) between predicted and actual block maxima within the forecast window, (2) Correlation between the predicted and actual

[1]https://www.narccap.ucar.edu/data/index.html

[2]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

TABLE I

PERFORMANCE COMPARISON AGAINST STATE-OF-THE-ART TIME SERIES REPRESENTATION LEARNING MODELS FOR BLOCK MAXIMA FORECASTING AND CLASSIFICATION. RMSE AND CORRELATION ARE CALCULATED FOR BLOCK MAXIMA PREDICTION. F1 SCORE IS MEASURED FOR CLASSIFICATION EXTREME EVENTS (CATEGORY 4 AND ABOVE FOR HURRICANE AND ABOVE 90TH PERCENTILE FOR CLIMATE AND ECL DATA)

Evaluation on Representation Learning Models									
Methods	Climate			Hurricane			ECL		
	RMSE	Corr	F1	RMSE	Corr	F1	RMSE	Corr	F1
CoST [12]	3.08 ± 0.27	0.80 ± 0.03	0.85 ± 0.02	15.12 ± 0.25	0.87 ± 0.04	0.84 ± 0.04	0.98 ± 0.05	0.79 ± 0.03	0.82 ± 0.03
TS2Vec [17]	2.99 ± 0.28	0.81 ± 0.03	0.86 ± 0.01	15.03 ± 0.26	0.88 ± 0.03	0.86 ± 0.03	0.95 ± 0.03	0.78 ± 0.03	0.80 ± 0.04
TNC [18]	3.05 ± 0.23	0.80 ± 0.02	0.84 ± 0.03	15.06 ± 0.28	0.89 ± 0.02	0.85 ± 0.03	0.94 ± 0.04	0.78 ± 0.04	0.82 ± 0.04
TimeCLR [13]	3.18 ± 0.28	0.79 ± 0.03	0.82 ± 0.03	15.22 ± 0.30	0.85± 0.03	0.83 ± 0.04	1.04 ± 0.07	0.76 ±0.03	0.81 ± 0.03
SimEXT (TWD)	**2.82 ± 0.25**	**0.82 ± 0.02**	0.87 ± 0.02	14.86 ± 0.22	**0.90 ± 0.03**	0.88 ± 0.04	**0.88 ± 0.05**	**0.81 ± 0.03**	**0.85 ± 0.03**
SimEXT (CMD)	2.84 ± 0.28	0.80 ± 0.03	**0.88 ± 0.03**	**14.82 ± 0.24**	0.89 ± 0.03	**0.90 ± 0.02**	0.90 ± 0.04	0.80 ± 0.04	0.84± 0.04

TABLE II

PERFORMANCE COMPARISON OF DOWNSTREAM MODELS UNDER DIFFERENT EXPERIMENT SETTINGS FOR BLOCK MAXIMA FORECASTING AND CLASSIFICATION. RMSE AND CORRELATION ARE CALCULATED FOR BLOCK MAXIMA PREDICTION. F1 SCORE IS MEASURED FOR CLASSIFICATION EXTREME EVENTS (CATEGORY 4 AND ABOVE FOR HURRICANE AND ABOVE 90TH PERCENTILE FOR CLIMATE AND ECL DATA)

Evaluation on Downstream Models										
Methods	Configuration	Climate			Hurricane			ECL		
		RMSE	Corr	F1	RMSE	Corr	F1	RMSE	Corr	F1
LSTM	Original Features	3.22 ± 0.25	0.73 ± 0.04	0.74 ± 0.04	15.51 ± 0.35	0.76 ± 0.35	0.78 ± 0.03	1.22 ± 0.09	0.74 ± 0.05	0.76 ± 0.03
	AE	3.18 ± 0.26	0.72 ± 0.05	0.75 ± 0.05	15.55 ± 0.37	0.78 ± 0.05	0.79 ± 0.04	1.12 ± 0.08	0.76 ± 0.03	0.77 ± 0.03
	AE + CL	3.11 ± 0.24	0.75 ± 0.04	0.78 ± 0.03	15.34 ± 0.32	0.81 ± 0.03	0.82 ± 0.03	1.06 ± 0.05	0.76 ± 0.02	0.79 ± 0.02
	AE + CL + TWD	3.02 ± 0.25	0.77 ± 0.02	0.78 ± 0.03	15.12 ± 0.26	0.82 ± 0.04	0.82 ± 0.03	1.00 ± 0.06	0.78 ± 0.03	0.80 ± 0.02
	AE + CL + CMD	2.97 ± 0.24	0.76 ± 0.03	0.79 ± 0.04	15.09 ± 0.24	0.82 ± 0.03	0.84 ± 0.02	0.99 ± 0.07	0.79± 0.03	0.79 ± 0.03
Informer	Original Features	3.19 ± 0.23	0.73 ± 0.04	0.78 ± 0.05	15.33 ± 0.32	0.79 ± 0.04	0.82± 0.03	1.11 ± 0.08	0.76 ± 0.03	0.77 ± 0.03
	AE	3.15 ± 0.21	0.74 ± 0.05	0.79 ± 0.05	15.30 ± 0.29	0.81 ± 0.03	0.81 ± 0.03	1.05 ± 0.05	0.75 ± 0.03	0.77 ± 0.04
	AE + CL	3.08 ± 0.23	0.77 ± 0.03	0.82 ± 0.04	15.08 ± 0.24	0.86 ± 0.04	0.84 ± 0.04	0.99 ± 0.07	0.77 ± 0.04	0.80 ± 0.03
	AE + CL + TWD	2.97 ± 0.25	0.81 ± 0.02	0.83 ± 0.03	14.94 ± 0.23	0.86 ± 0.04	0.86 ± 0.02	0.96 ± 0.05	0.79 ± 0.03	0.81 ± 0.03
	AE + CL + CMD	2.99 ± 0.26	0.78 ± 0.03	0.85 ± 0.04	14.97 ± 0.25	0.87 ± 0.04	0.89 ± 0.02	0.98 ± 0.06	0.80 ± 0.03	0.82 ± 0.03
Transformer	Original Features	3.21 ± 0.26	0.72 ± 0.05	0.78 ± 0.05	15.41 ± 0.36	0.78 ± 0.03	0.81 ± 0.04	1.14 ± 0.07	0.75 ± 0.04	0.75 ± 0.04
	AE	3.12 ± 0.23	0.73 ± 0.06	0.77 ± 0.06	15.36 ± 0.35	0.80 ± 0.04	0.82 ± 0.03	1.04 ± 0.06	0.77 ± 0.02	0.78 ± 0.03
	AE + CL	3.03 ± 0.25	0.78 ± 0.04	0.83 ± 0.05	15.13 ± 0.29	0.87 ± 0.05	0.85 ± 0.04	1.01 ± 0.07	0.77 ± 0.03	0.79 ± 0.02
	AE + CL + TWD	2.94 ± 0.29	0.80 ± 0.03	0.86 ± 0.03	14.98 ± 0.25	0.88 ± 0.03	0.87 ± 0.03	0.99 ± 0.06	0.80 ± 0.02	0.81 ± 0.04
	AE + CL + CMD	2.91 ± 0.26	0.79 ± 0.04	0.85 ± 0.04	14.95 ± 0.28	0.86 ± 0.04	0.89 ± 0.02	0.95 ± 0.04	0.80 ± 0.01	0.83 ± 0.02
EVL	Original Features	3.28 ± 0.27	0.72 ± 0.03	0.76 ± 0.05	15.44 ± 0.30	0.78 ± 0.04	0.78 ± 0.03	1.09 ± 0.07	0.77 ± 0.03	0.74 ± 0.05
	AE	3.21 ± 0.25	0.74 ± 0.04	0.78 ± 0.04	15.40 ± 0.32	0.80 ± 0.05	0.78 ± 0.03	1.04 ± 0.05	0.77 ± 0.02	0.77 ± 0.02
	AE + CL	3.18 ± 0.27	0.76 ± 0.03	0.82 ± 0.03	15.21 ± 0.28	0.80 ± 0.03	0.81 ± 0.03	1.02 ± 0.05	0.78 ± 0.03	0.78 ± 0.03
	AE + CL + TWD	3.07 ± 0.24	0.79 ± 0.02	0.84 ± 0.03	15.10 ± 0.21	0.84 ± 0.04	0.84 ± 0.04	0.98 ± 0.07	0.80 ± 0.02	0.82 ± 0.03
	AE + CL + CMD	3.05 ± 0.26	0.79 ± 0.03	0.83 ± 0.04	15.07 ± 0.24	0.83 ± 0.03	0.84 ± 0.02	0.99 ± 0.05	0.79± 0.02	0.81 ± 0.04
DeepExtrema	Original Features	3.10 ± 0.17	0.73 ± 0.05	0.78 ± 0.04	15.18 ± 0.26	0.83 ± 0.04	0.84 ± 0.04	1.02 ± 0.07	0.77 ± 0.04	0.80 ± 0.03
	AE	3.02 ± 0.19	0.75 ± 0.04	0.79 ± 0.05	15.21 ± 0.28	0.82 ± 0.03	0.84 ± 0.03	0.98 ± 0.06	0.79 ± 0.02	0.82 ± 0.02
	AE + CL	2.92 ± 0.21	0.78 ± 0.05	0.84 ± 0.04	15.04 ± 0.25	0.88 ± 0.04	0.88 ± 0.03	0.95 ± 0.06	0.78 ± 0.03	0.83 ± 0.03
	AE + CL + TWD	**2.79 ± 0.24**	**0.83 ± 0.02**	0.88 ± 0.02	14.81 ± 0.23	**0.90 ± 0.02**	0.88 ± 0.03	**0.86 ± 0.06**	**0.83 ± 0.02**	0.85 ± 0.03
	AE + CL + CMD	2.81 ± 0.26	0.81 ± 0.04	**0.89 ± 0.02**	**14.78 ± 0.25**	0.89 ± 0.02	**0.89 ± 0.02**	0.89 ± 0.05	0.82 ± 0.03	**0.86 ± 0.03**

block maxima of the forecast window, (3) F1 score of the extreme event detection, where an extreme event is defined as a hurricane intensity value surpassing 130 mph (i.e., category 4 and above hurricanes) or a temperature and electricity consumption value exceeding the 90th percentile at a specific location (for the climate and ECL datasets).

B. Experimental Results

Table I provides a comprehensive comparison of the performance of `SimEXT` compared to other state-of-the-art representation learning methods. The results suggest that the proposed `SimEXT` framework, which incorporates distribution losses (TWD and CMD), outperforms all the baseline methods, namely CoST [12], TS2Vec [17], TNC [18], and TimeCLR [13], in terms of RMSE, Correlation, and F1 Score. Furthermore, Figure 2 provides a visual representation of the probability distribution comparison between the ground truth, `SimEXT`, and TS2Vec for block maxima forecasting using the hurricane dataset. The figure demonstrates that the proposed `SimEXT` significantly outperforms TS2Vec in matching the ground truth distribution for Category 1-5 hurricanes. These findings serve as compelling evidence supporting the superior performance of `SimEXT` in capturing the tail distribution of a time series while simultaneously learning its feature representation.

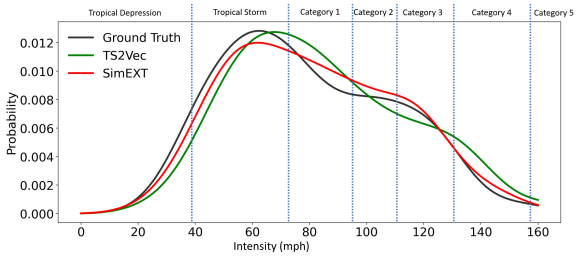

Fig. 2. Probability distribution comparison of ground truth, `SimEXT`, and TS2Vec for block maxima forecasting with hurricane data

Table 2 presents the results of applying `SimEXT` to five downstream models (LSTM, Informer, Transformer, EVL,

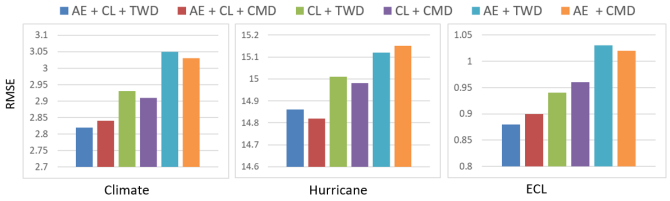

Fig. 3. RMSE comparison of the different modules of SimEXT under different experimental setting for block maxima forecasting

and DeepExtrema) for block maxima forecasting. The table showcases the performance of SimEXT under five different settings: Original Features, AE, AE + CL, AE + CL + TWD, and AE + CL + CMD. Consistently, the results suggest that the incorporation of distribution losses (AE + CL + TWD and AE + CL + CMD) significantly enhances the performance of block maxima prediction for all downstream models. This finding underscores the effectiveness of the proposed distribution losses in capturing extreme values and improving forecasting accuracy. Moreover, the results reveal that the DeepExtrema model for downstream tasks, when using it with the SimEXT representation learning approach, achieves the best performance, exhibiting the lowest RMSE and the highest correlation and F1 score. The combined evidence from Tables I and II supports the superiority of SimEXT in capturing extreme values and enhancing representation learning for time series data. By incorporating distribution losses, SimEXT outperforms state-of-the-art methods. Additionally, when applied to downstream models for block maxima forecasting, SimEXT consistently improves performance across various settings.

C. Ablation study

In this study, we investigate the effect of gradually incorporating the AE, CL, and TWD/CMD modules into the proposed SimEXT framework. Figure 3 shows the RMSE values obtained when using different modules of the SimEXT framework. The results suggest that incorporating CL (contrastive learning) alone yields more substantial benefits compared to incorporating only AE (Autoencoder) for all datasets used. This finding aligns with current understanding that contrastive learning is generally more beneficial in learning meaningful representations compared to autoencoder-based methods. Nevertheless, our results also highlight the positive impact of incorporating the distribution loss, ultimately leading to the best performance across all three datasets. This suggests that while the autoencoder module alone may not deliver optimal results on its own, it does exert a positive impact when integrated alongside the CL and distribution loss (TMD or CMD).

V. Conclusion

This paper introduces SimEXT, a novel self-supervised learning framework for modeling time series extreme values. It combines a novel wavelet-based data augmentation with contrastive learning and auto-encoders to learn time series representations. To preserve extreme values, SimEXT includes a distribution loss function focused on capturing block maxima. Experimental results on real-world data show that SimEXT enhances the performance of existing representation learning and downstream approaches for forecasting block maxima.

VI. Acknowledgment

This research is supported by the U.S. National Science Foundation under grant IIS-2006633. Any use of trade, firm, or product names are for descriptive purposes only and do not imply endorsement by the U.S. Government.

References

[1] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An introduction to statistical modeling of extreme values*, vol. 208. Springer, 2001.

[2] G. J. Ross, "Modelling financial volatility in the presence of abrupt changes," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 2, pp. 350–360, 2013.

[3] D. Kumar, "Sudden changes in extreme value volatility estimator: Modeling and forecasting with economic significance analysis," *Economic Modelling*, vol. 49, pp. 354–371, 2015.

[4] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[5] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances, and challenges," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 42–62, 2022.

[6] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[9] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[10] H. CRAMAR, "On the composition of elementary errors," *Skand. Aktuarietids*, vol. 11, pp. 13–74, 1928.

[11] C. W. Landsea and J. L. Franklin, "Atlantic hurricane database uncertainty and presentation of a new database format," *Monthly Weather Review*, vol. 141, no. 10, pp. 3576–3592, 2013.

[12] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," *arXiv preprint arXiv:2202.01575*, 2022.

[13] X. Yang, Z. Zhang, and R. Cui, "Timeclr: A self-supervised contrastive learning framework for univariate time series representation," *Knowledge-Based Systems*, vol. 245, p. 108606, 2022.

[14] 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, pp. 11106–11115, 2021.

[15] 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*, pp. 1114–1122, 2019.

[16] A. H. Galib, A. McDonald, T. Wilson, L. Luo, and P.-N. Tan, "DeepExtrema: A Deep Learning Approach for Forecasting Block Maxima in Time Series Data," *arXiv preprint arXiv:2205.02441*, 2022.

[17] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8980–8987, 2022.

[18] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," *arXiv preprint arXiv:2106.00750*, 2021.