DeepGPD: A Deep Learning Approach for Modeling Geospatio-Temporal Extreme Events

Tyler Wilson,[1] **Pang-Ning Tan,**[1] **Lifeng Luo**[2]

[1] Dept of Computer Science and Engineering, Michigan State University
[2] Department of Geography, Michigan State University
{wils1270,ptan,lluo}@msu.edu

Abstract

Geospatio-temporal data are pervasive across numerous application domains. These rich datasets can be harnessed to predict extreme events such as disease outbreaks, flooding, crime spikes, etc. However, since the extreme events are rare, predicting them is a hard problem. Statistical methods based on extreme value theory provide a systematic way for modeling the distribution of extreme values. In particular, the generalized Pareto distribution (GPD) is useful for modeling the distribution of excess values above a certain threshold. However, applying such methods to large-scale geospatio-temporal data is a challenge due to the difficulty in capturing the complex spatial relationships between extreme events at multiple locations. This paper presents a deep learning framework for long-term prediction of the distribution of extreme values at different locations. We highlight its computational challenges and present a novel framework that combines convolutional neural networks with deep set and GPD. We demonstrate the effectiveness of our approach on a real-world dataset for modeling extreme climate events.

Introduction

Extreme geospatio-temporal events such as flooding, heat waves, and droughts are destructive natural forces with the potential to cause devastating losses in property and human lives. According to NOAA's National Center for Environmental Information, as of July 2021, there were 8 billion dollar weather/climate disaster events in 2021 alone, incurring close to $30 billion in total losses. Given the severity of their impact, accurate modeling of extreme events are therefore critical to provide timely information to the public threatened by such hazards and to minimize the risk for human casualties and property destruction.

Numerous methods have been developed in the past for modeling extremes. This includes outlier detection methods (Cheng et al. 2009, 2008b,a; Boriah et al. 2008), where the goal is not to predict future extreme events but to detect them retrospectively from observation data after they have occurred. Statistical approaches based on extreme value theory (Katz, Parlange, and Naveau 2002; Kharin and Zwiers 2005; López and Francés 2013) are also commonly used to infer the statistical distribution of the extreme values. Another approach is to cast the prediction of extreme events as a supervised learning problem (Nayak and Ghosh 2013; Laptev et al. 2017), which is the approach used in this paper. Specifically, we are interested in predicting the conditional distribution of excesses over a threshold (e.g., monthly precipitation or temperature that exceeds their 95th percentile) at various spatial locations. However, predicting the conditional distribution of such excesses is a challenging problem due to their rare frequency of occurrence. In addition, the predictive model must consider the complex spatial relationships between events at multiple locations. Identifying such complex and potentially nonlinear interactions among the predictors is a challenge that must be addressed.

In recent years, there have been growing interest in developing deep learning algorithms to address various spatio-temporal modeling problems (Wilson, Tan, and Luo 2018; Shi et al. 2015; Liu et al. 2019). For spatial data, one emerging technique that can effectively handle spatial relationships in the data is convolutional neural network (CNN). CNN initially found its application in computer vision tasks, but has since been successfully applied to a wider range of problems, including climate downscaling (Vandal et al. 2017), precipitation nowcasting (Shi et al. 2015) and hail prediction (Gagne et al. 2019). However, despite their growing body of literature, there has been scant research on spatio-temporal deep learning for modeling extreme events.

Non-parametric deep learning methods are generally ineffective at inferring the distribution of extreme events unless there are sufficiently long, historical observation data available. When trained for regression problems, deep learning models are generally trained to predict the conditional mean of a distribution using the mean squared error loss, and thus, fail to capture the tail of the distribution. Extreme events are governed by two parametric distributions (Coles 2001): the distribution of block maxima is governed by the generalized extreme value distribution (GEV) and the distribution of excesses over a threshold are governed by the generalized Pareto distribution (GPD).

In this paper, we propose a novel framework that combines extreme value theory (EVT) with deep learning. Specifically, our framework leverages the strengths of deep learning in modeling complex relationships in geospatio-temporal data as well as the ability of GPD to capture

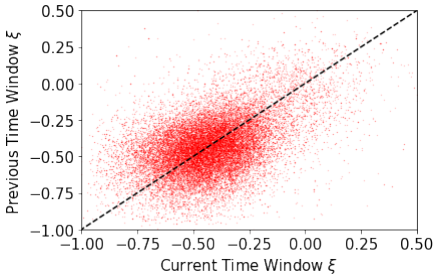


Figure 1: Relationship between shape parameter ξ of generalized Pareto distribution for modeling precipitation excesses in two successive time windows.

the distribution of excess values with limited observations. However, integrating a deep neural network (DNN) with EVT is a challenge as the loss function minimized by the DNN must be modified to maximize the likelihood function of the GPD. Another computational challenge is that the sufficient statistics of GPD must satisfy certain positivity constraints unlike the output of DNN, which are typically unconstrained. Furthermore, the distribution of extreme values are often temporally correlated. For example, Figure 1 shows the relationship between the shape parameter ξ of the GPD distribution for precipitation excesses from one year to the next based on 45-year data from more than 1000 stations considered in our study. This poses a challenge from a modeling perspective as the number of excesses above a threshold tends to vary from one time step to the next. Developing a deep learning approach that can incorporate such variable number of excess values as predictors, in addition to other fixed length vectors, is another challenge to be addressed.

The major contributions of this paper are as follows:

1. We propose a deep learning framework to model the distribution of extreme events. The framework combines CNN with deep sets (Zaheer et al. 2017) for modeling geo-spatial relationships among predictors that include fixed-length vectors and variable-sized sets.
2. We propose a re-parameterization method for constraining the outputs of the DNN so that they satisfy the requisite constraints and present an algorithm that learns the GPD parameters in an end-to-end fashion.
3. We evaluate our proposed framework on a real world climate dataset and demonstrate its effectiveness in comparison to conventional EVT and deep learning approaches.

Related Work

Convolutional neural networks (Krizhevsky, Sutskever, and Hinton 2012) have gained considerable attention over the last several years due to its success in artificial intelligence applications. The strength of CNN lies in its ability to model spatial relationships. For example, CNN has been successfully used to model spatial relationships in geographic applications (Máttyus, Luo, and Urtasun 2017; Zhang, Liu, and Wang 2018; Sun et al. 2018; Lebedev et al. 2019; Vandal et al. 2017; Kaiser et al. 2017). A similar intuition motivates the application of CNN to model temporal autocorrelations in time series data. For example, Bai et al. (Bai, Kolter, and Koltun 2018) provided empirical evidence that a properly designed convolutional architecture can outperform recurrent neural networks. There have also been concerted efforts to model both spatial and temporal relationships jointly using CNN (Tran et al. 2015, 2018; Ji et al. 2010). Instead of applying convolution to model temporal relationships, some research uses recurrent layers to model temporal relationships (Donahue et al. 2015). For example, (Shi et al. 2015) replaces every instance of matrix multiplication in an LSTM with 2d convolution and then feeds the data to the LSTM at each time step. However, none of these approaches are designed for modeling extreme values.

Statistical approaches based on extreme value theory (EVT) (Katz, Parlange, and Naveau 2002; Kharin and Zwiers 2005; López and Francés 2013) are commonly used to infer the distribution of extreme values. Several recent papers have combined deep learning with EVT but often only as a post-processing step. For example,(Wu et al. 2021) fit a GPD to the residuals of a neural network to help detect cyber risks. Similarly, Yu et al. (Yu et al. 2021) identify samples with unknown classes at test time using the Weibull distribution. Weng et al. (Weng et al. 2018) utilize EVT to derive a neural network robustness metric called CLEVER. In none of these cases are deep learning and EVT integrated together within a single end-to-end learning framework. Instead, EVT is used as a post-processing step to identify unusual samples or as a robustness score of the network. In contrast we integrate EVT directly into our deep learning formulation to predict the GPD parameters and training it in an end-to-end fashion. Ding et al. (Ding et al. 2019) do integrate EVT into the loss function but in an ad-hoc way, where the CDF of the GPD is used to assign weights on extreme samples whose prediction is framed as a binary classification problem. Rather than learning the parameters of GPD, they instead treated them as user-provided hyper-parameters.

Preliminaries

Let $\mathcal{D} = \left\{ (X_{il}, Y_{il}) \;\middle|\; i \in \{1, \cdots, n\};\ l \in \{1, \cdots, L\} \right\}$ be a geospatio-temporal dataset, where X_{il} denote the predictor attribute values for the time window $(t_{i-1}, t_i]$ in location l and Y_{il} denote the corresponding target (response) values for the time window $(t_i, t_{i+1}]$. Since we are interested in predicting the excesses above a threshold in the next time window, the target variable corresponds to the set of excess values at location l during the period $(t_i, t_{i+1}]$, i.e., $Y_{il} = \{y_{tl} \mid y_{tl} \geq u, t \in (t_i, t_{i+1}]\}$. In addition, the predictors can be divided into two groups, $X_{il} \equiv (X^v_{il}, X^s_{il})$, where $X^v_{il} \in \mathbb{R}^d$ is a fixed length vector and $X^s_{il} \in \mathbb{R}^{p_i}$ is a variable length vector corresponding to the set of excess values in the previous window, i.e., $X^s_{il} = \{y_{tl} \mid y_{tl} \geq u, t \in (t_{i-1}, t_i]\}$. Note that the number of excess values can vary, e.g., one window may have 10 excess values while the previous window has only 5 excess values. The collections of excess values associated with the current and next time windows form the sets X^s_{il} and Y_{il}, respectively. Our goal is to estimate the

conditional distribution $P(Y_{il,j}|X_{il})$ for all the locations l_i conditioned on the predictors observed in the current window, where $Y_{il,j}$ is an element of the set Y_{il}.

Extreme Value Theory

This paper focuses primarily on the use of generalized Pareto distribution (GPD) for modeling the distribution of excesses above a given threshold. For example, in precipitation prediction, one may be interested in modeling the distribution of high precipitation values above a certain threshold.

Let $Y_1, Y_2, \cdots$ be a sequence of independent and identically distributed random variables. Given an excess value $Y = u + y$, where u is some pre-defined threshold, the conditional probability of observing the excess event is:

$$P(Y - u \leq y \mid Y > u) = \begin{cases} 1 - \left[1 + \frac{\xi y}{\sigma}\right]^{-1/\xi}, & \xi \neq 0 \\ 1 - e^{-y}, & \xi = 0 \end{cases}$$

Furthermore, its density function is given by:

$$P(y) = \begin{cases} \frac{1}{\sigma}\left[1 + \frac{\xi y}{\sigma}\right]^{-\frac{1}{\xi}-1}, & \xi \neq 0 \\ \frac{1}{\sigma} e^{-\frac{y}{\sigma}} & \xi = 0 \end{cases} \quad (1)$$

subject to the constraint $\forall y : 1 + \frac{\xi y}{\sigma} > 0$. The GPD has two parameters, shape, ξ, and scale, σ The shape parameter has a significant impact on the overall structure of the probability density. When ξ is negative, the support of the distribution is finite such that $0 < y < -\frac{\sigma}{\xi}$ due to the constraint. When ξ is zero or positive, its support ranges from 0 to positive infinity.

The advantage of using the GPD to model extreme values is its generality as one does not have to know the underlying distribution of the random variable prior to thresholding since the distribution of excesses will be governed by the GPD in relatively general conditions. In many cases, the values of ξ and σ may depend on some contextual features as predictors x. Assuming a linear relationship between ξ and x and between $\log(\sigma)$ and x (the log linear relationship is used to guarantee that the estimate of σ is non-negative):

$$\xi = f_\xi(x) = w_1^T x, \quad \log(\sigma) = f_\sigma(x) = w_2^T x \quad (2)$$

where w_1 and w_2 are the model parameters, which can be learned by minimizing the negative log-likelihood of GPD.

One important consideration when modeling data using a GPD is the choice of threshold u since the threshold must be set high enough for the GPD to be applicable. A common way to evaluate the suitability of a given threshold is by examining the mean residual life plot. If a collection of samples were drawn from a GPD then the empirical distribution of the excesses should have a linear relationship with selected threshold. Specifically, we have:

$$E(Y - u|Y > u) = \frac{\sigma_0 + \xi u}{1 - \xi} \quad (3)$$

for threshold u, and $Y \sim GPD(\xi, \sigma_0)$. In the experiment section, we will verify our choice of threshold by examining the mean residual life plot for our precipitation data.

Deep Set

To accommodate the variable size set of excess values as input predictor, x_{il}^s, we employ a deep set architecture (Zaheer et al. 2017) to transform the variable-length input into fixed size vector. The transformation consists of the following two stages. The first stage is responsible for transforming each element of the set, $x_{il,j}^s$, from its raw representation into a high-level vector representation, $h_{il,j}$ by using a fully connected network, ϕ. These element-wise representations are then aggregated to obtain a fixed-length vector representation for the set. This set-level representation is then used as input to a fully connected network, ρ, to produce the final output representation, $z_{il}^s = \rho\left[\sum_j \phi(x_{il,j}^s)\right]$

Proposed DeepGPD Framework

Figure 2 shows the architecture of our `DeepGPD` framework, which has the following three major components:

1. **Local Feature Extraction** - This component is responsible for transforming both the (fixed-length) vector-valued, x_{il}^v, and (variable-length) set-valued predictors, x_{il}^s, at each location into a fixed-length feature vector.
2. **Spatial Feature Extraction** - This component models the spatial relationships among the predictors in the data.
3. **Extreme Value Modeling (EVM)** - This component is responsible for ensuring that the constraints on the GPD parameters are satisfied by the induced model.

Local Feature Extraction

This component is responsible for learning a representation of the predictors associated with each location l by utilizing both the set-valued predictors x_{il}^s and the vector-valued predictors x_{il}^v. Learning a representation of the predictors is challenging for two reasons. First, because the set-valued predictors are variable length we must transform them into a fixed length vector so that it can be used by the later stages of the model. Second, the set-valued predictors may not always be available for some locations.

To address the first challenge, we employ the deep set architecture described in subsection to transform the set-valued predictors into a fixed-length vector, z_{il}^s. For the second challenge, there may be some cases where a given grid cell lacks set-valued predictors, x_{il}^s. In these cases we set $z_{il}^s = 0$. However, zeroing the inputs in this way risks the possibility that predictions at locations without set predictors will be distorted. To address this, an indicator variable, I_{il} is introduced to indicate whether set-valued predictors are available at a given location and time. This indicator variable is then concatenated with the vector-valued predictors and the deep set representation of the set-valued predictors to generate the following vector: $z_{il} = z_{il}^s \parallel I_{il} \parallel x_{il}^v$, where $\parallel$ denotes the concatenation operator.

Our deep set implementation assumes there is a maximum set size. Sets that are smaller than this maximum size are padded with dummy values of zeros so that each set can be represented by a fixed length vector. Each dummy element is processed in the same way as the real set elements. After

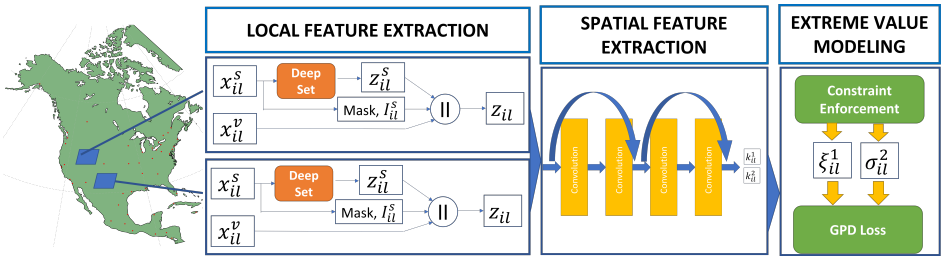


Figure 2: Proposed DeepGPD framework.

each set element is processed by several fully connected layers, only the representations of the actual set elements (i.e. dummy elements excluded) are averaged together. This is implemented through the use of a masking array multiplied by the set member representations element-wise.

Spatial Feature Extraction

After extracting a separate representation for each location, we need to model the spatial relationships between the representations at different locations. DeepGPD uses a CNN to capture the geospatial relationships in the data. In our architecture, we arrange the representation extracted from all the gridded locations into a 3-dimensional tensor (excluding the batch dimension) and then provide the tensor as input to a CNN with residual layers (He et al. 2016). The final linear layer of the CNN produces a response map for each location, $k_{il} \in \mathbb{R}^2$, for the prediction time window $(t_i, t_{i+1}]$.

Extreme Value Modeling (EVM)

The EVM component is designed to predict the conditional distribution of excess values by utilizing the response map generated by the CNN. Specifically, it will convert the CNN output for each location and time window $(t_i, t_{i+1}]$ to the generalized Pareto model parameters, ξ_{il} and σ_{il}. These parameters enable us to infer various statistics about the excess values in the predicted time window, such as the expected values at varying quantiles (including maximum and median value) as well as their return level.

Unlike previous work such as (Ding et al. 2019), which assumes that ξ and σ are hyperparameters provided by users, DeepGPD enables both parameters to be automatically learned from the data. Specifically, the deep architecture is trained to minimize the following negative log-likelihood function of the excess values in the next time step:

$$\mathcal{L}(\{\xi_{il}, \sigma_{il}\}) = \sum_{i,l,j} \left[\log \sigma_{il} + (1 + \frac{1}{\xi_{il}}) \log(1 + \xi_{il} \frac{y_{ilj}}{\sigma_{il}}) \right] \tag{4}$$

where i is the window number, l is the location and j is the sample number.

One major computational challenge in estimating the GPD parameters using a deep learning architecture is the need to enforce positivity constraints on the solution of (4) during training. To address this challenge, DeepGPD employs a re-parameterization trick to transform (ξ_{il}, σ_{il}) into a pair of unconstrained variables $k_{il} = (k_{il}^{(1)}, k_{il}^{(2)})$ that can be learned by the convolutional neural network.

Theorem 1 *Let $\{\xi_{il}^*, \sigma_{il}^*\} = \text{argmin } \mathcal{L}(\{\xi_{il}, \sigma_{il}\})$ subject to the following positivity constraints:*

$$\forall i, j, l: \ \sigma_{il} > 0 \ \text{and} \ 1 + \xi_{il} \frac{y_{ilj}}{\sigma_{il}} > 0$$

By re-parameterizing $(\xi_{il}, \sigma_{il}) \mapsto (k_{il}^{(1)}, k_{il}^{(2)})$ as follows:

$$\sigma_{il} = \exp(k_{il}^{(1)}), \quad \xi_{il} = \exp(k_{il}^{(2)}) - \frac{\exp(k_{il}^{(1)})}{M_{il}} \tag{5}$$

and solving for $\{\hat{k}_{il}^{(1)}, \hat{k}_{il}^{(2)}\} = \text{argmin } \hat{\mathcal{L}}(\{u_{il}, v_{il}\})$, where

$$\hat{\mathcal{L}}(\{u_{il}, v_{il}\}) = \sum_{ilj} \left[u_{il} + \left(1 + \frac{M_{il}}{M_{il} e^{v_{il}} - e^{u_{il}}}\right) \times \log\left(1 + e^{v_{il}} \frac{y_{ilj}}{e^{u_{il}}} - \frac{y_{ilj}}{M_{il}}\right) \right] \tag{6}$$

and $M_{il} = \max_j Y_{ilj}$, then the solution set $\{\xi_{il}^*, \sigma_{il}^*\}$ can be derived from the solution for $\{\hat{k}_{il}^{(1)}, \hat{k}_{il}^{(1)}\}$ by applying the mapping given in Equation (5).

The proof for the preceding theorem can be shown by substituting (5) into (4), which yields the equivalent objective function for $\hat{\mathcal{L}}(\{k_{il}^{(1)}, k_{il}^{(2)}\})$. Furthermore, since Equation (4) can be re-written as follows:

$$\begin{aligned} \sigma_{il} &= e^{k_{il}^{(1)}} \geq 0 \\ 1 + \xi_{il} \frac{y_{ilj}}{\sigma_{il}} &= 1 - \frac{y_{ilj}}{M_{il}} + e^{k_{il}^{(2)}} \frac{y_{ilj}}{e^{k_{il}^{(1)}}} \geq 0 \end{aligned}$$

the positivity constraints are automatically satisfied given the fact that $\forall i, l, j: y_{ilj} \leq M_{il}$, $e^{k_{il}^{(1)}} > 0$ and $e^{k_{il}^{(2)}} > 0$ as long as $k_{il}^{(1)}$ and $k_{il}^{(2)}$ are not equal to $-\infty$.

Corollary 1 *The DeepGPD framework trained to optimize the loss function in Equation* (6) *will generate the maximum likelihood solution for $\{\xi_{il}^*, \sigma_{il}^*\}$ in Equation* (4) *given the one-to-one mapping with $\{\hat{k}_{il}^{(1)}, \hat{k}_{il}^{(2)}\}$ in Equation* (5).

The preceding corollary demonstrates the advantages of using our re-parameterization trick to train DeepGPD as the values of $\hat{k}_{il}^{(1)}$ and $\hat{k}_{il}^{(2)}$ are less constrained compared to

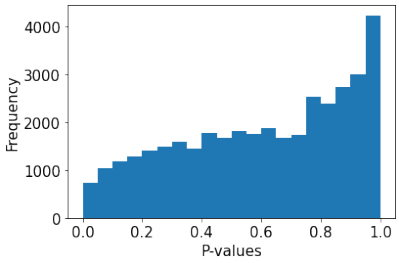


Figure 3: Fitted p-Value distribution of K-S test.

$\{\xi^*_{il}$ and $\sigma^*_{il}\}$. This enables the parameters to be more easily learned by `DeepGPD`. All three components of the framework, including deep set and CNN, are trained in an end-to-end fashion using Adam (Kingma and Ba 2015). Once the parameters for $\hat{k}^{(1)}_{il}$ and $\hat{k}^{(2)}_{il}$ are obtained, we can apply Equation (5) to derive the corresponding GPD parameters.

Experimental Results

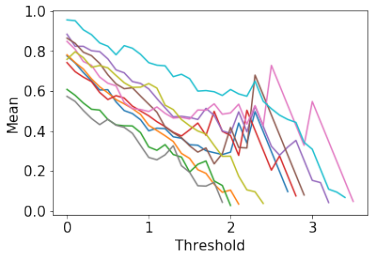
We evaluate our proposed framework on a 44-year global precipitation data from 1970 to 2013. Specifically, we use daily precipitation values collected from the Global Historical Climatology Network[1] (GHCN) for 1,112 stations located in the Northern Hemisphere (between 22.5°N to 67.5°N) as our target variable. The data is partitioned into 45 non-overlapping one-year time windows. Excess daily precipitation values are considered as any value exceeding one standard deviation above the mean for the station. For predictor variables, we consider the excess values in the previous year as set-valued attributes and the mean and standard deviation of monthly climate values (e.g., convective precipitation rate, solar radiation flux, relative humidity, and sea level pressure) from the NCEP re-analysis project[2] as fixed-length vector-valued attributes.

Our objective is to predict the conditional distribution of the excess precipitation values for next year based on the observed excess values and statistics of the NCEP climate variables for the current year. The precipitation data at each location is de-seasonalized separately using its own monthly means and standard deviations. Each 1 year window of predictor and target values are assigned to either training, validation or test sets, with 34 windows in training, 5 in validation and 4 in test. We repeat our experiment 10 times with different train-validation-test splits.

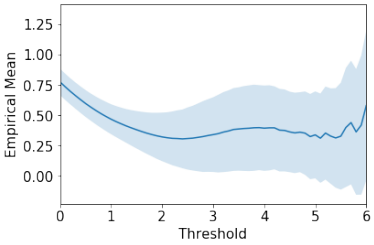
To verify that the excess values follow the GPD, we perform the Kolmogorov-Smirnov goodness of fit test. The KS-test is a non-parametric approach to determine whether a given set of samples is drawn from a given distribution. To do this, we first infer values of the GPD statistics, ξ and σ, from the excess values observed at each location and time window using SciPy genpareto class and then apply the KS-test to assess whether the excess values were indeed drawn from the inferred distribution. We observe that the average p-value over all the locations and time windows is 0.60, which suggests that the inferred distributions accurately fit the data. The distribution of the fitted p-values is shown in Figure 3.

[1]https://www.ncdc.noaa.gov/ghcn-daily-description

[2]https://www.ncep.noaa.gov/



(a) Excess mean for random samples of locations and time windows. Colors represent different samples.



(b) Excess mean for all locations and time windows.

Figure 4: Mean residual life plot for excess precipitation.

Method	NLL	$\rho(\xi)$	$\rho(\sigma)$
Persistence	0.6271 ± 0.0092	0.40	0.77
Linear Regression	0.6514 ± 0.0114	0.49	0.80
ViT Regression	0.6372 ± 0.0088	0.56	**0.87**
CNN Regression	0.6332 ± 0.0102	0.41	0.75
Linear GPD	0.6101 ± 0.0035	0.38	0.57
`DeepGPD`	**0.5688** ± 0.0036	**0.57**	0.85

Table 1: Comparison between `DeepGPD` against baseline methods in terms of negative log-likelihood (NLL) and correlation (ρ) of predicted ξ and σ to ground truth values.

Method	Negative log-likelihood
`DeepGPD` with GHCN only	0.5706 ± 0.0035
`DeepGPD` with NCEP only	0.5670 ± 0.0040
`DeepGPD`	0.5688 ± 0.0036

Table 2: Results of ablation study.

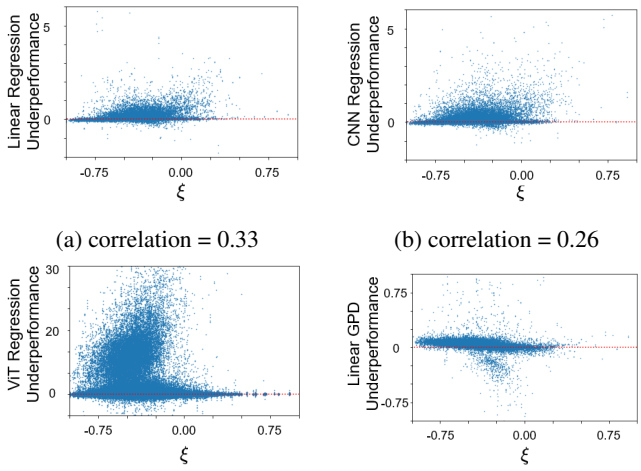
Next we evaluate our choice of excess threshold. Equation 3 shows a linear relationship between the choice of threshold and the mean value of the excesses. We empirically verify this linear relationship by plotting the chosen threshold against the mean of excesses in Figure 4. The resulting diagram is also known as the mean residual life plot. Ordinarily, the mean residual life plot is used to evaluate the choice of threshold for a single GPD distribution. However, in our case, each window and location has its own GPD distribution. Thus we must evaluate our choice of threshold for this entire family of excess distributions.

In Figure 4a, we plot a random selection of mean residual life plots at different locations and windows and we find that the relationship between thresholds and the mean residual is approximately linear around 1. In addition, Figure 4b shows the average excess mean across all time windows and locations at any given threshold with shading representing 1 standard deviation. In the vicinity of 1, the average excess mean across all distributions varies linearly with the choice of threshold and the relatively narrow shade suggests this behavior is shared across most distributions. Notice that the slope of both plots is negative around thresholds of 1 which, based on Equation 3, indicates that ξ is negative-valued. The approximately linear behavior in the mean residual life plot around 1 justifies our choice of 1 standard deviation as our threshold.

We compare `DeepGPD` against the following baselines. Similar to `DeepGPD`, each baseline generates the GPD parameters of a location for the next time window as its output. (1) **Persistence** - A GPD is fitted to the excess values in the current window and used to predict the next window. (2) **Linear Regression** - A linear regression model is trained to predict the GPD parameters (ξ_{il} and σ_{il}) for each location using the NCEP climate variables as well as the fitted GP parameters from previous window as its predictors. (3) **ViT Regression** - This baseline uses the Vision Transformer architecture (Dosovitskiy et al. 2020) to predict the GPD parameters and is trained using mean squared error loss (MSE). To accommodate our pixel-wise regression problem setting we remove the class token embeddings and set the final MLP to output 2 scalars for each pixel in each patch corresponding to the 2 GPD parameters. (4) **CNN Regression** - This baseline uses the same architecture (including deep set and CNN) and predictors as `DeepGPD` except it replaces the maximum likelihood loss with a mean square error loss on the GPD parameters, similar to the linear regression baseline. This is similar to the architecture proposed in (He et al. 2016) but consists only of residual layers and a final linear layer while omitting the pooling and fully connected layers because it performs pixel-wise regression. (5) **Linear GPD** (Coles 2001) - This is a linear GPD model for predicting the GPD parameters using NCEP and the GPD parameters from the previous window as its predictors (see Equation (2)). Hyper-parameters for all deep learning models were selected as follows: learning rates between 10^{-2} and 10^{-5}, number of layers ranged from 3 to 10, and hidden dimensions between 5 and 50 units were explored. [3]

Comparison against Baseline Methods

Table 1 compares the performance of `DeepGPD` against baselines using negative log-likelihood and correlation between the predicted and actual ξ values as evaluation metrics. The results in Table 1 suggest that, with one exception, `DeepGPD` significantly outperforms all the baselines regardless of the metric chosen. The performance of CNN regression and linear regression are poor relative to other baselines. Since both methods employ the mean-square error (MSE) of the predicted GPD parameters as its loss function, this shows the importance of explicitly incorporating extreme value theory and its corresponding negative log-likelihood loss to train the model. The only baseline to compare favorably to the proposed method according to any metric is ViT regression which achieves correlations with the ground truth ξ and σ comparable to `DeepGPD` due to being an extremely expressive model trained explicitly to predict the ground truth ξ and σ values through MSE loss. However, because it doesn't incorporate NLL into it's loss in practice it makes poor predictions of the distribution of observed excesses. The relatively strong performance of the persistence method suggests the importance of using information about the excess values in the previous time window to predict their distribution in the next window.



(a) correlation = 0.33 (b) correlation = 0.26

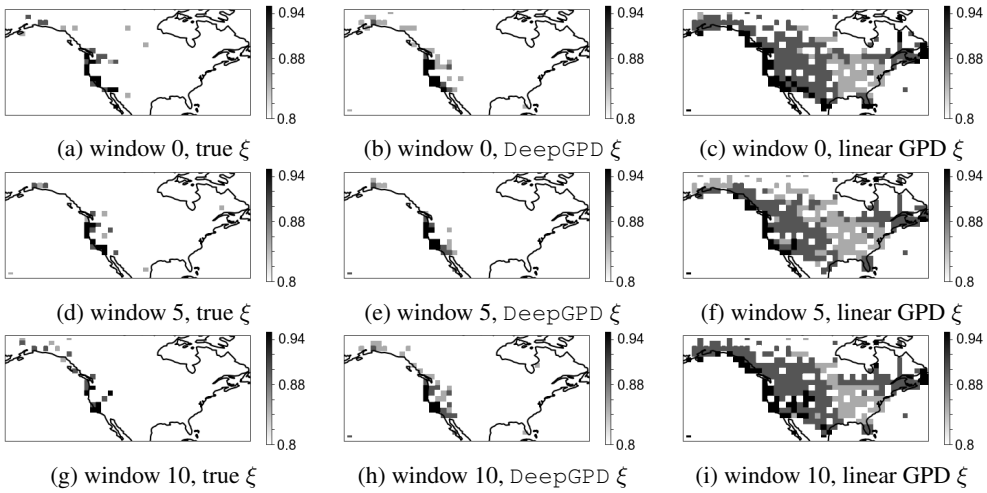
(c) correlation = 0.01 (d) correlation = -0.25

Figure 5: Relationship between predictive improvements over baselines and true ξ.

The linear GPD model employs the same loss function (i.e., MLE) as `DeepGPD` except it uses a linear layer, as opposed to non-linear model, to learn the mapping from the predictors to the GPD parameters. This has two additional implications. First, linear GPD is unable to directly incorporate the set-valued predictors of the GHCN; therefore they can only use the inferred GPD statistics from previous window as one of its predictors. Furthermore, existing linear GPD approach also does not incorporate spatial information since it does not include a spatial component such as CNN. As a result, the proposed `DeepGPD` method outperforms linear GPD by a significantly large margin, demonstrating the importance of non-linearity and incorporating spatial relationship into the modeling task. Nevertheless, the linear GPD still outperforms other baselines, suggesting the value of incorporating GPD into the learning formulation.

Table 2 compares our full model against variations that utilize only the vector-valued (NCEP only) or the set-valued predictors (GHCN only). The results show that all three methods achieve comparable performance with significant overlap in their confidence intervals. This suggests that there

[3]The code and data for our implementation is available at https://github.com/TylerPWilson/deepGPD.



(a) window 0, true ξ (b) window 0, DeepGPD ξ (c) window 0, linear GPD ξ

(d) window 5, true ξ (e) window 5, DeepGPD ξ (f) window 5, linear GPD ξ

(g) window 10, true ξ (h) window 10, DeepGPD ξ (i) window 10, linear GPD ξ

Figure 6: Comparison between the spatial distribution of the true and predicted ξ values for linear GPD and DeepGPD.

exists large amount of redundant information between both types of predictors. Although our model can effectively utilize the set-valued or vector-valued predictors, leveraging them together does not appear to improve the model.

Distribution of Estimated GPD Parameters

The previous subsection compares DeepGPD against the baseline methods in terms of their negative log-likelihood (NLL) and correlation with the ground truth GPD shape parameter values. Table 1 shows that DeepGPD significantly outperforms the baselines but this raises the question of the reason for its performance improvement. Since the results in Table 1 are based on an aggregation over multiple time windows and locations, we need to compare the relative performance of DeepGPD against the baselines at a finer level. To do this, we first compute the difference between the NLL of each baseline against DeepGPD for each location and time window. Positive differences suggest that the NLL value of the baseline is worse than the NLL of DeepGPD. Figure 5 displays the relationship between the NLL difference of each baseline relative to the proposed method and the true GPD shape parameter ξ. First, note that there is a positive bias along the y-axis in the plots, which suggests that the performance improvement in DeepGPD is observed for the majority of the locations and time windows. In fact, DeepGPD outperforms the baselines in 58% to 89% of locations and windows. Second, in the case of linear and CNN regression methods, observe that there is a positive correlation between the value of ξ and relatively stronger performance by the proposed method. This suggests that DeepGPD performs best in situations where the tails of the distribution are heaviest. Since extreme events are the ones most important to predict, the strong performance of DeepGPD in these scenarios is promising. This plot also shows there is a considerable number of samples for which the ViT model performs worse than DeepGPD. The plot also suggests that linear GPD outperforms DeepGPD when ξ is in the range between -0.5 and 0. Nevertheless, there are still more data points with positive NLL difference for the same range of ξ values.

Next, we examine the spatial distribution of the predicted ξ values and compare them to their ground truth distribution. Since large values of ξ are especially important we focus on them. In Figure 6 we plot the predicted values of ξ for three time windows with grid cells colored based on their relationship to certain high thresholds. These thresholds range from 85th to 95th quantiles calculated based on the ground truth data with ξ values exceeding the 95th quantile of the ground truth ξ values colored black and everything below the 80th percentile colored white and any ξ in between colored gray. We produce separate plots for the ground truth, DeepGPD, and the linear GPD baseline. Due to space limitations we only show the results for linear GPD. These plots show that DeepGPD does well in predicting the general spatial distribution of the highest ξ values by identifying which locations have relatively high or low ξ values. The worst predictions are from linear GPD, which struggles to capture the variability of the data with almost all locations visualized as black or gray due to the large positive bias of the model as well as the low standard deviation of its predictions.

Conclusions

In this paper we identified the limitations of existing deep learning methods in predicting the distribution of extreme values. To address this limitation we proposed a novel deep learning architecture (DeepGPD) capable of learning the parameters of the generalized Pareto distribution while satisfying the conditions placed on those parameters. We evaluated our results on a real world climate data set and showed that DeepGPD outperformed various baseline methods.

One limitation of this work is that it models the marginal distribution at each location and window separately. However, in some applications it is important to consider the dependence structure of extreme values. In future work we plan to study the joint distribution of extreme values using techniques like copula theory and multivariate extreme value distributions.

Acknowledgments

This work was partially supported by the National Science Foundation under grant IIS-2006633.

References

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271 [cs]*. ArXiv: 1803.01271.

Boriah, S.; Kumar, V.; Steinbach, M.; Tan, P.; Potter, C.; and Klooster, S. 2008. Detecting Ecosystem Disturbances and Land Cover Change using Data Mining. In Kargupta, H.; Han, J.; and Yu, P. S., eds., *Next Generation of Data Mining*.

Cheng, H.; Tan, P.; Potter, C.; and Klooster, S. 2008a. A Robust Graph-based Algorithm for Detection and Characterization of Anomalies in Noisy Multivariate Time Series. In *Proceedings of ICDM Workshop on Spatial and Spatiotemporal Data Mining (STDM 08)*. Pisa, Italy.

Cheng, H.; Tan, P.-N.; Potter, C.; and Klooster, S. 2008b. Data Mining for Visual Exploration and Detection of Ecosystem Disturbaces. In *Proc of 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)*. Irvine, CA.

Cheng, H.; Tan, P.-N.; Potter, C.; and Klooster, S. 2009. Detection and Characterization of Anomalies in Multivariate Time Series. In *Proceedings of SIAM International Conference on Data Mining*. Sparks, NV.

Coles, S. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Springer.

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.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2625–2634.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Gagne, D. J.; Haupt, S. E.; Nychka, D. W.; and Thompson, G. 2019. Interpretable Deep Learning for Spatial Analysis of Severe Hailstorms. *Monthly Weather Review*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2010. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35: 221–231.

Kaiser, P.; Wegner, J. D.; Lucchi, A.; Jaggi, M.; Hofmann, T.; and Schindler, K. 2017. Learning Aerial Image Segmentation From Online Maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11): 6054–6068.

Katz, R. W.; Parlange, M. B.; and Naveau, P. 2002. Statistics of Extremes in Hydrology. *Advances in Water Resources*, 25(8-12): 1287–1304.

Kharin, V. V.; and Zwiers, F. W. 2005. Estimating extremes in transient climate change simulations. *Journal of Climate*, 18(8): 1156–1173.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*, 1097–1105. Curran Associates, Inc.

Laptev, N.; Yosinski, J.; Li, L. E.; and Smyl, S. 2017. Time-series Extreme Event Forecasting with Neural Networks at Uber. In *International Conference on Machine Learning*, volume 34, 1–5.

Lebedev, V.; Ivashkin, V.; Rudenko, I.; Ganshin, A.; Molchanov, A.; Ovcharenko, S.; Grokhovetskiy, R.; Bushmarinov, I.; and Solomentsev, D. 2019. Precipitation Nowcasting with Satellite Imagery. *arXiv:1905.09932 [cs]*. ArXiv: 1905.09932.

Liu, X.; Wilson, T.; Tan, P.-N.; and Luo, L. 2019. Hierarchical LSTM Framework for Long-Term Sea Surface Temperature Forecasting. In *Proceedings of 6th IEEE International Conference on Data Science and Advanced Analytics*.

López, J.; and Francés, F. 2013. Non-stationary flood frequency analysis in continental Spanish rivers, using climate and reservoir indices as external covariates. *Hydrology and Earth System Sciences*, 17(8): 3189–3203.

Máttyus, G.; Luo, W.; and Urtasun, R. 2017. Deeproadmapper: Extracting Road Topology from Aerial Images. In *Proceedings of the IEEE International Conference on Computer Vision*, 3438–3446.

Nayak, M. A.; and Ghosh, S. 2013. Prediction of Extreme Rainfall Event Using Weather Pattern Recognition and Support Vector Machine Classifier. *Theoretical and applied climatology*, 114(3-4): 583–603.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; and WOO, W.-c. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems 28*, 802–810.

Sun, T.; Chen, Z.; Yang, W.; and Wang, Y. 2018. Stacked U-Nets with Multi-output for Road Extraction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 187–1874. IEEE.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 4489–4497. IEEE.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6450–6459. IEEE.

Vandal, T.; Kodra, E.; Ganguly, S.; Michaelis, A.; Nemani, R.; and Ganguly, A. R. 2017. Deepsd: Generating Gigh Resolution Climate Change Projections Through Single Image Super-resolution. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, 1663–1672. ACM.

Weng, T.-W.; Zhang, H.; Chen, P.-Y.; Yi, J.; Su, D.; Gao, Y.; Hsieh, C.-J.; and Daniel, L. 2018. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. *arXiv preprint arXiv:1801.10578*.

Wilson, T.; Tan, P.-N.; and Luo, L. 2018. A Low Rank Weighted Graph Convolutional Approach to Weather Prediction. In *Proceedings of IEEE International Conference on Data Mining*.

Wu, M. Z.; Luo, J.; Fang, X.; Xu, M.; and Zhao, P. 2021. Modeling Multivariate Cyber Risks: Deep Learning Dating Extreme Value Theory. *arXiv preprint arXiv:2103.08450*.

Yu, X.; Zhao, Z.; Zhang, X.; Zhang, Q.; Liu, Y.; Sun, C.; and Chen, X. 2021. Deep Learning-Based Open Set Fault Diagnosis by Extreme Value Theory. *IEEE Transactions on Industrial Informatics*.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep sets. In *Advances in neural information processing systems*, 3391–3401.

Zhang, Z.; Liu, Q.; and Wang, Y. 2018. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5): 749–753.