Cover Letter for IJCAI 2022 Submission

This is a resubmission of the following paper:

- Paper ID – 11671;
- Title – DeepExtrema: A Deep Learning Approach for Extreme Value Prediction in Time Series Data;
- Conference – AAAI2022

In this document, we have provided the main reasons for rejections, the changes made to address the concerns, the reviewers' comments, the authors' response to the reviewers, as well as the submitted paper.

Main Reasons for Rejection and Changes Made

1. The reviewers have pointed out that the study lacks recent baseline models to compare against. Specifically, we did not consider the recent methods, such as transformer and extreme event forecasting method.

   To address this concern, we have added two new baseline models to compare against the proposed method. Primarily, a Transformer-based model consistent with time series forecasting is implemented as a baseline. In addition, a recent method called EVL [Ding *et al.*, 2019] for modeling extreme events in time series forecasting is also incorporated as another baseline.

2. The reviewers also questioned on its practicality and proof of performance as it was only evaluated on a single data set. More evaluation on other data sets are crucial to justify the contribution of the study.

   To address this concern, we have evaluated the proposed model on two new data sets. We have considered solar home electricity (half-hour) dataset [aus, 2013] from the Ausgrid. The data for this study is based on half-hourly energy use (kWh) for 55 families over the course of 284 days. Also, we have used a weather dataset from Kaggle competition [Muthukumar.J, 2017]. The data used in this study is based on hourly temperature data for a city over a ten-year period.

3. Also, the reviewers have pointed out that this study lacks clarity in terms of the presentation. Like, how the overall architecture is related to the proposed theorem, how Figure 1 is weakly related, how the Model Bias Offset helps, and how the hyperparameters are selected?

   To address this concern, we have clarified the connection among different parts more carefully, we have updated the Figure 1 (overall architecture). We have made major revision on the text to clarify the overall model and explain its individual parts.

AAAI Responses to Reviewers:

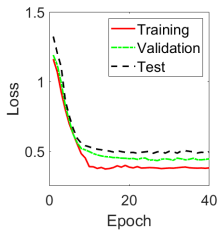
Response to Reviewer 2

Q1. I wonder the practicality of the proposed approach ... only by collecting, storing, and feeding data sequences in a long time span could the proposed DNN architecture fit the predicting function well. The use of historical sequences for hurricane prediction is common (see, Alemany et al., AAAI 2019; Kordmahalleh et al., 2016). Statistical models used by the National Hurricane Center also incorporate historical climatology and persistence to generate their forecasts. Hurricane forecasting mostly depends on current environmental conditions. It is possible to use the values from recent time steps to predict the hurricane intensity in the next few time steps with reasonably high accuracy, as shown by our results and others.

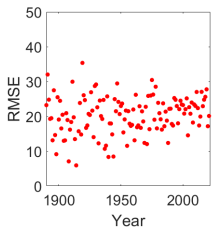
Q2. How much does the system tend to *regret* before it estimates a parameter set including μ, σ, ξ, ξ'? Figure 1a shows the convergence of our algorithm. Observe that it does not require too many epochs (nor regrets) to estimate the parameters (μ, σ, ξ, ξ') and achieve the reported test accuracy.

Q3. How could the system deal with sequences drawn from a time-varying (drifting) distribution? More importantly, can the trained system execute inference in real-time? Fig. 1b shows RMSE of the persistence model (using block maxima from previous time steps to predict maxima for the next 8 time steps). The average RMSE does not change much over the past 130 years, suggesting its near stationarity. DeepExtrema can be extended to non-stationary data in two ways. First, the sequences can be augmented with outputs from climate models to account for greenhouse gas emission, which causes the drift. Second, it can be extended to online learning setting, similar to (Ding et al., 2021). Both of thse are interesting topic for future research. Finally, DeepExtrema can perform inference in real-time by executing a simple forward pass given the values from its previous time steps.

Q4. The connection between Figure 1 (DNN architecture) and Theorem 1 is weak. The LSTM component of our architecture (Fig. 1) generates the GEV parameters as its output. These outputs must satisfy the constraints given in Eq. (6). Theorem 1 restates the constraints in terms of the upper and lower bounds of ξ. With this reparameterization, as long as the LSTM is "properly" initialized, Eq (11) guarantees that the bounds are satisfied at all times. Fig. 2 shows our strategy to ensure the initial LSTM outputs will preserve the bounds in Theorem 1. In the first iteration, the randomly initialized LSTM may output μ_0, σ_0, ξ_0, and ξ'_0, which do not satisfy the bounds. Let μ_{desired}, σ_{desired}, $\xi_{0,\text{desired}}$, and $\xi'_{0,\text{desired}}$ be the desired initial values consistent with the bounds. We denote the difference between the initial and desired outputs as μ_{fix}, σ_{fix}, ξ_{fix}, and ξ'_{fix}. Following Eq. (13), we only need to subtract μ_{fix}, σ_{fix}, ξ_{fix}, and ξ'_{fix} from the LSTM output in all subsequent iterations to ensure they satisfy the bounds.



(a) Convergence plot.



(b) RMSE over time.

Figure 1: Convergence of DeepExtrema (left) and RMSE of persistence model over time (right).

Q5. Can the proposed method be compared with recent competitors such as time-series anomaly detection methods? Most anomaly detection methods will detect/predict the presence of anomalies, rather than the magnitude of the extremes, which is our focus here (i.e., finding maximum hurricane intensity over a time period). A block maxima does not have to be an anomaly but its magnitude is still useful.

Response to Reviewer 4

Q1. Are the results from a single(best) run or average over multiple runs? The results in Table 2 and Table 3 are based on the average over 10 runs.

Q2. There are statistical track model that track the hurricane's movement over the ocean, such as PepC [1]. I am very curious to know how will DeepExtrema perform compare with such model. PepC focuses on point prediction, with lead time up to 24 hours, whereas DeepExtrema focuses on block maxima prediction for up to 48 hours. Hence, PepC will likely perform better for now-

casting tasks whereas DeepExtrema is better at inferring maximum intensity value within a time interval as it employs extreme value theory. Using LSTM or any sequential model (e.g., Markov model in PepC) without GEV will unlikely predict the block maxima well as such models are more focused on predicting the overall trend instead of block maxima.

Q3. Unclear what exactly the hyper-parameters of FCN, LSTM architectures are. The hyperparameters of FCN and LSTM are selected using Ray Tune, a tuning framework with ASHA (asynchronous successive halving algorithm) scheduler for early stopping. The FCN baseline has a depth of 3 and a width of 18 while the LSTM baseline has a depth of 3 and a width of 11. We have added this into the paper.

Q4. Also there are more advanced models like transformer that handles the time dependence better. While Transformer can handle time dependence better, it is not clear it will perform well for predicting block maxima and extremes. DeepExtrema can use Transformer as its underlying DNN instead of LSTM. This could be a future extension of the framework.

Response to Reviewer 5

Q1. λ_1 in the objective function of DeepExtrema denotes the trade-off between GEV loss and RMSE loss. However, by observing Equation 14 and Equation 15, I think that λ_2 represents this. Thank you for the comment. We have corrected the typo.

Q2. There are few baseline methods, which are not enough to prove the performance. I suggest to add other baseline methods for comparison. We will add transformer as another baseline. Our results for transformer are 14.03 (RMSE) and 0.84 (correlation), which are slightly worse as the transformer is used in a sequence to single output modeling, which differs from typical sequence to sequence modeling.

Q3. The main contribution this paper makes is incremental and has limited impact. Incorporating the parameter constraints of GEV into deep learning is non-trivial. It is an important contribution as DNN become widely used in environmental domains, where modeling extremes is critical.

References

[aus, 2013] Solar home electricity data, Dec 2013.

[Ding *et al.*, 2019] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. Modeling extreme events in time series prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1114–1122, 2019.

[Muthukumar.J, 2017] Muthukumar.J. Weather dataset, Dec 2017.

View Reviews

Paper ID
11671

Paper Title
DeepExtrema: A Deep Learning Approach for Extreme Value Prediction in Time Series Data

Track Name
Main Track

Reviewer #2

Questions

1. {Summary} Please briefly summarize the main claims/contributions of the paper in your own words. (Please do not include your evaluation of the paper here).
The authors proposed a deep learning framework for extreme value detection in time-series, which is novel and challenging.

2. {Novelty} How novel are the concepts, problems addressed, or methods introduced in the paper?
Fair: The paper contributes some new ideas.

3. {Soundness} Is the paper technically sound?
Fair: The paper has minor, easily fixable, technical flaws that do not impact the validity of the main results.

4. {Impact} How do you rate the likely impact of the paper on the AI research community?
Fair: The paper is likely to have moderate impact within a subfield of AI.

5. {Clarity} Is the paper well-organized and clearly written?
Fair: The paper is somewhat clear, but some important details are missing or unclear.

6. {Evaluation} If applicable, are the main claims well supported by experiments?
Fair: The experimental evaluation is weak: important baselines are missing, or the results do not adequately support the main claims.

7. {Resources} If applicable, how would you rate the new resources (code, data sets) the paper contributes? (It might help to consult the paper's reproducibility checklist)
Fair: The shared resources are likely to be moderately useful to other AI researchers.

8. {Reproducibility} Are the results (e.g., theorems, experimental results) in the paper easily reproducible? (It may help to consult the paper's reproducibility checklist.)
Good: key resources (e.g., proofs, code, data) are available and key details (e.g., proofs, experimental setup) are sufficiently well-described for competent researchers to confidently reproduce the main results.

9. {Ethical Considerations} Does the paper adequately address the applicable ethical considerations, e.g., responsible data collection and use (e.g., informed consent, privacy), possible societal harm (e.g., exacerbating injustice or discrimination due to algorithmic bias), etc.?
Good: The paper adequately addresses most, but not all, of the applicable ethical considerations.

10. {Reasons to Accept} Please list the key strengths of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).
The authors proposed a deep learning framework for extreme value detection in time-series, which is novel and indeed challenging.

11. {Reasons to Reject} Please list the key weaknesses of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).

However, I wonder the practicality of the proposed approach. In particular, as a data-hungry system, only by collecting, storing, and feeding data sequences in a long time span could the proposed DNN architecture fit the predicting function well. The question then is: can we afford such time to wait the system converging? In other words, how much does the system tend to *regret* before it estimates a parameter set including \mu, \sigma, \epsilon, \epsilon' that delivers a saddle point? Even how could the system deal with data sequences drawn from a time-varying (drifting) distribution? More importantly, can the trained system execute inference in real-time?
In addition, the theoretical part is questionable for me. The connection between Figure 1 (DNN architecture) and Theorem 1 is weak. Can please supplement some intuition behind this? Why the proposed DNN is guaranteed to satisfy the bounds as desired?
Also, the empirical study is not supportive. The comparison methods are vanilla. Can the proposed method be compared with more recent competitors such as time-series anomaly detection methods? The fact that only two datasets are used for benchmarking undermines the solidness of the proposal. No statistical evidence could substantiate the viability and superiority of the proposed method.

12. {Questions for the Authors} Please provide questions that you would like the authors to answer during the author feedback period. Please number them.

However, I wonder the practicality of the proposed approach. In particular, as a data-hungry system, only by collecting, storing, and feeding data sequences in a long time span could the proposed DNN architecture fit the predicting function well. The question then is: can we afford such time to wait the system converging?
In other words, how much does the system tend to *regret* before it estimates a parameter set including \mu, \sigma, \epsilon, \epsilon' that delivers a saddle point?
Even how could the system deal with data sequences drawn from a time-varying (drifting) distribution? More importantly, can the trained system execute inference in real-time?
Can please supplement some intuition behind this? Why the proposed DNN is guaranteed to satisfy the bounds as desired?
Can the proposed method be compared with more recent competitors such as time-series anomaly detection methods?

13. {Detailed Feedback for the Authors} Please provide other detailed, constructive, feedback to the authors.

The authors proposed a deep learning framework for extreme value detection in time-series, which is novel and indeed challenging. However, I wonder the practicality of the proposed approach. In particular, as a data-hungry system, only by collecting, storing, and feeding data sequences in a long time span could the proposed DNN architecture fit the predicting function well. The question then is: can we afford such time to wait the system converging? In other words, how much does the system tend to *regret* before it estimates a parameter set including \mu, \sigma, \epsilon, \epsilon' that delivers a saddle point? Even how could the system deal with data sequences drawn from a time-varying (drifting) distribution? More importantly, can the trained system execute inference in real-time?
In addition, the theoretical part is questionable for me. The connection between Figure 1 (DNN architecture) and Theorem 1 is weak. Can please supplement some intuition behind this? Why the proposed DNN is guaranteed to satisfy the bounds as desired?
Also, the empirical study is not supportive. The comparison methods are vanilla. Can the proposed method be compared with more recent competitors such as time-series anomaly detection methods? The fact that only two datasets are used for benchmarking undermines the solidness of the proposal. No statistical evidence could substantiate the viability and superiority of the proposed method.

14. (OVERALL EVALUATION) Please provide your overall evaluation of the paper, carefully weighing the reasons to accept and the reasons to reject the paper. Ideally, we should have: - No more than 25% of the submitted papers in (Accept + Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 20% of the submitted papers in (Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 10% of the submitted papers in (Very Strong Accept + Award Quality) categories - No more than 1% of the submitted papers in the Award Quality category

Reject: For instance, a paper with technical flaws, weak evaluation, inadequate reproducibility, incompletely addressed ethical considerations.

Reviewer #4

Questions

1. {Summary} Please briefly summarize the main claims/contributions of the paper in your own words. (Please do not include your evaluation of the paper here).
The paper tackles the problem of extreme value forecasting in time series. It combines the generalized extreme value distribute with deep learning and proposes DeepExtrema which enables both point-wise and quantile prediction of block maxima in a forecast window. They applied DeepExtrema to predict the intensity of a hurricane.

2. {Novelty} How novel are the concepts, problems addressed, or methods introduced in the paper?
Fair: The paper contributes some new ideas.

3. {Soundness} Is the paper technically sound?
Good: The paper appears to be technically sound, but I have not carefully checked the details.

4. {Impact} How do you rate the likely impact of the paper on the AI research community?
Fair: The paper is likely to have moderate impact within a subfield of AI.

5. {Clarity} Is the paper well-organized and clearly written?
Good: The paper is well organized but the presentation could be improved.

6. {Evaluation} If applicable, are the main claims well supported by experiments?
Fair: The experimental evaluation is weak: important baselines are missing, or the results do not adequately support the main claims.

7. {Resources} If applicable, how would you rate the new resources (code, data sets) the paper contributes? (It might help to consult the paper's reproducibility checklist)
Fair: The shared resources are likely to be moderately useful to other AI researchers.

8. {Reproducibility} Are the results (e.g., theorems, experimental results) in the paper easily reproducible? (It may help to consult the paper's reproducibility checklist.)
Good: key resources (e.g., proofs, code, data) are available and key details (e.g., proofs, experimental setup) are sufficiently well-described for competent researchers to confidently reproduce the main results.

9. {Ethical Considerations} Does the paper adequately address the applicable ethical considerations, e.g., responsible data collection and use (e.g., informed consent, privacy), possible societal harm (e.g., exacerbating injustice or discrimination due to algorithmic bias), etc.?
Excellent: The paper comprehensively addresses all of the applicable ethical considerations.

10. {Reasons to Accept} Please list the key strengths of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).
The idea of combining the generalized extreme value distribute with deep learning to predict extremes in time series is very interesting and would have tremendous value in practice. Predict hurricane is just one of such application.

The reformulation of the GEV maximum log-likelihood estimation constraint, and the introduction of the GEV loss to estimate the GEV parameters using deep neural networks are somewhat non-trivial.

11. {Reasons to Reject} Please list the key weaknesses of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).
Overall, DeepExtrema presented in Table 3 is doing slightly better than other baselines on real-world data.

However, it is unclear what exactly are the hyper-parameters of FCN, LSTM architectures are, e.g., depth, width,

etc., and how the hyper-parameters are selected. Table 3 only reports a single number of each method. I am not sure how to interpret such result, since it could merely just from a lucky run.

Also there are more advanced models like transformer that handles the time dependence better.

12. {Questions for the Authors} Please provide questions that you would like the authors to answer during the author feedback period. Please number them.
1. Are the results presented in Table 2 and 3 from a single(best) run or they are the average over multiple runs.

2. There are statistical track model that track the hurricane's movement over the ocean, such as PepC [1]. I am very curious to know how will DeepExtrema perform compare with such model.

[1] Jing, R., & Lin, N. (2020). An environment-dependent probabilistic tropical cyclone model. Journal of Advances in Modeling Earth Systems, 12, e2019MS001975. https://doi.org/10.1029/2019MS001975

13. {Detailed Feedback for the Authors} Please provide other detailed, constructive, feedback to the authors.
Overall the idea of combining the generalized extreme value distribute with deep learning to predict extreme values in time series is interesting and would have substantial societal value. However, I think the experimental evaluation can be better presented. See my previous comments.

14. (OVERALL EVALUATION) Please provide your overall evaluation of the paper, carefully weighing the reasons to accept and the reasons to reject the paper. Ideally, we should have: - No more than 25% of the submitted papers in (Accept + Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 20% of the submitted papers in (Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 10% of the submitted papers in (Very Strong Accept + Award Quality) categories - No more than 1% of the submitted papers in the Award Quality category
Borderline accept: Technically solid paper where reasons to accept, e.g., novelty, outweigh reasons to reject, e.g., limited evaluation. Please use sparingly.

Reviewer #5

Questions

1. {Summary} Please briefly summarize the main claims/contributions of the paper in your own words. (Please do not include your evaluation of the paper here).
The paper proposes a novel deep learning framework (DeepExtrema) that combines extreme value theory with deep learning to address the challenges of predicting extremes in time series. In addition, the paper offers a reformulation and reparameterization technique for satisfying constraints as well as a model bias offset technique for proper model initialization. The experiments comparing with baselines on synthetic and real-world data show the effectiveness of the model.

2. {Novelty} How novel are the concepts, problems addressed, or methods introduced in the paper?
Fair: The paper contributes some new ideas.

3. {Soundness} Is the paper technically sound?
Good: The paper appears to be technically sound, but I have not carefully checked the details.

4. {Impact} How do you rate the likely impact of the paper on the AI research community?
Fair: The paper is likely to have moderate impact within a subfield of AI.

5. {Clarity} Is the paper well-organized and clearly written?
Excellent: The paper is well-organized and clearly written.

6. {Evaluation} If applicable, are the main claims well supported by experiments?
Good: The experimental evaluation is adequate, and the results convincingly support the main claims.

7. {Resources} If applicable, how would you rate the new resources (code, data sets) the paper contributes? (It might help to consult the paper's reproducibility checklist)
Fair: The shared resources are likely to be moderately useful to other AI researchers.

8. {Reproducibility} Are the results (e.g., theorems, experimental results) in the paper easily reproducible? (It may help to consult the paper's reproducibility checklist.)
Good: key resources (e.g., proofs, code, data) are available and key details (e.g., proofs, experimental setup) are sufficiently well-described for competent researchers to confidently reproduce the main results.

9. {Ethical Considerations} Does the paper adequately address the applicable ethical considerations, e.g., responsible data collection and use (e.g., informed consent, privacy), possible societal harm (e.g., exacerbating injustice or discrimination due to algorithmic bias), etc.?
Not Applicable: The paper does not have any ethical considerations to address.

10. {Reasons to Accept} Please list the key strengths of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).
The proposed problem is interesting and important. Generally speaking, the paper is well-written and well-presented. In addition, the experiments results show the superiority of the proposed model and can prove the claims. Finally, the main idea of this work is novel and some useful examples are descried in the introduction part to support the claims.

11. {Reasons to Reject} Please list the key weaknesses of the paper (explain and summarize your rationale for your evaluations with respect to questions 1-9 above).
a. The main contribution this paper makes is incremental and has limited impact.
b. There are few experimental baseline methods, which are not enough to prove the performance of the model.

12. {Questions for the Authors} Please provide questions that you would like the authors to answer during the author feedback period. Please number them.
a. The ablation study shows that The hyperparameter λ1 in the objective function of DeepExtrema denotes the trade-off between GEV loss and RMSE loss. However, by observing Equation 14 and fEquation 15, I think that λ2 represents this. Therefore, I suggest that the author reconfirm.
b. There are few experimental baseline methods, which are not enough to prove the performance of the model. So, I suggest that the author add other baseline methods for comparison.

13. {Detailed Feedback for the Authors} Please provide other detailed, constructive, feedback to the authors.
Please see my questions listed above.

14. (OVERALL EVALUATION) Please provide your overall evaluation of the paper, carefully weighing the reasons to accept and the reasons to reject the paper. Ideally, we should have: - No more than 25% of the submitted papers in (Accept + Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 20% of the submitted papers in (Strong Accept + Very Strong Accept + Award Quality) categories; - No more than 10% of the submitted papers in (Very Strong Accept + Award Quality) categories - No more than 1% of the submitted papers in the Award Quality category
Borderline accept: Technically solid paper where reasons to accept, e.g., novelty, outweigh reasons to reject, e.g., limited evaluation. Please use sparingly.

DeepExtrema: A Deep Learning Approach for Extreme Value Prediction in Time Series Data

Submitted for Review

Abstract

Accurate forecasting of extreme values in time series is critical due to their significant impact on human and natural systems. Unfortunately, extreme values are difficult to predict due to their rare frequency of occurrence compared to other typical values observed in the data. Extreme value theory provides a systematic approach for capturing the tail distribution of stochastic processes but has limited capacity in terms of modeling complex nonlinear relationships present in the data. To overcome this limitation, this paper presents DeepExtrema, a novel framework that combines the generalized extreme value (GEV) distribution with deep learning in a principled way to leverage the strengths of both approaches. Implementing such a network is a challenge as the framework must preserve the inter-dependent constraints of the GEV model parameters. Model initialization is another challenge as the DNN must be parameterized in a way that can generate a feasible set of GEV parameters satisfying the constraints. In this paper, we describe our proposed strategies to address these challenges and present an architecture that enables both point-wise and quantile estimation of block maxima in a forecast window. Experiments performed on both real-world and synthetic data demonstrated the superiority of DeepExtrema compared to other baseline methods.

Introduction

Extreme events such as droughts, floods, and severe storms occur when the values of the corresponding geophysical variables (such as temperature, precipitation, or wind speed) reach their highest or lowest point during a period or surpass a threshold value. Extreme events have far-reaching consequences for both humans and the environment. For example, 4 of the most expensive hurricanes in the United States since 2005—Katrina, Sandy, Harvey, and Irma—have each incurred over $50 billion in damages with enormous death tolls (US GAO 2020). Accurate modeling and forecasting of extreme events are therefore crucial as it not only helps provide timely warnings to the public but also enables stakeholders and policymakers to assess the risk of potential hazards caused by the extreme events.

Modeling of time series with extremes can be tricky as the extreme values might be inadvertently discarded as outliers to improve the overall generalization performance of the prediction model. To help better capture the extremes, alternative loss functions beyond mean-square error can be employed, such as quantile loss, to better estimate the values at higher quantiles beyond the conditional mean estimates generated by the least-square models. While the quantile loss can be shown to be a maximum likelihood estimator of the asymmetric Laplace distribution (Koenker and Machado 1999), characterizing the asymptotic tail behavior of a random variable in terms of the asymmetric Laplace distribution is not well-motivated.

Towards this end, extreme value theory (EVT) is more statistically well-grounded, as it offers a principled approach to derive the limiting distribution governing a sequence of extreme values through the extremal types theorem (Coles et al. 2001). The two most popular distributions studied in EVT are the generalized extreme value (GEV) distribution and the generalized Pareto (GP) distribution. GEV is concerned with the distribution of block maxima, whereas GP is concerned with the distribution of excesses over a threshold. In this work, we focus solely on the GEV distribution as the block maxima allow us to assess the worst-case scenarios of the forecasts and avoids making ad-hoc decisions associated with selecting the GP distribution threshold. Unfortunately, classical EVT has limited capacity in terms of modeling highly complex, nonlinear relationships present in time series data. It is typically used in conjunction with simple statistical models that can only infer simple relationships among the predictors and response variables. For example, (Kharin and Zwiers 2005) uses a linear model with predictors to predict the parameters of a GEV distribution.

In recent years, deep learning methods have grown in popularity due to their ability to capture nonlinear temporal dependencies and other patterns in the data. Previous work has utilized a range of deep learning architectures for time series modeling, including long short-term memory (LSTM) (Sagheer and Kotb 2019; Masum, Liu, and Chiverton 2018), convolutional neural network (Bai, Kolter, and Koltun 2018; Zhao et al. 2017; Yang et al. 2015), encoder-decoder based RNN (Peng et al. 2018), and attention-based models (Zhang et al. 2019; Aliabadi et al. 2020). (Wang et al. 2020) proposed a hybrid uncertainty quantification approach for point estimations and quantile estimations. Most of the existing works focus on generating point estimates of the conditional mean of the predictions with only a few of them (Ding et al.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2019; Polson and Sokolov 2020) focusing on forecasting extremes in time series. Although some of these methods incorporate EVT into their deep learning formulation, they either make unrealistic assumption that the distribution parameters are known (Ding et al. 2019) or do not enforce the necessary inter-dependent constraints on the distribution parameters (Polson and Sokolov 2020). In addition, these methods are designed for modeling excesses over a threshold using the GP distribution instead of forecasting the block maxima with GEV distribution, which is the focus of this paper.

Incorporating GEV distribution into the deep learning formulation is a challenge for several reasons. First, using the GEV maximum likelihood estimation (MLE) imposes two positivity constraints on the optimization problem to ensure that the predicted distribution has a finite bound (Coles et al. 2001). Maintaining these constraints during training is a challenge especially when the model parameters depend on the observed predictor values in a mini-batch. Second, inferring the distinct GEV parameters for different time series is difficult as there is a limited number of extreme events in the time series. Finally, the training process is highly sensitive to the model initialization. Third, improper initialization may lead to violations of the positivity constraints as the initial output values from the DNN may not be consistent with the data. Also, it may lead to unfeasible GEV parameters which are inconsistent with the GEV maximum log-likelihood estimation. For instance, for GEV shape parameter $\xi < -1$, MLE estimates do not exist; and its mean is not defined for $\xi > 1$ (Coles et al. 2001).

To overcome these challenges, we propose a novel deep learning framework called `DeepExtrema` that utilizes the GEV distribution from EVT to parameterize the distribution of extreme values and deep learning for learning nonlinear relationships among the predictors. It combines a hybrid loss function associated with GEV negative log-likelihood and mean-square-error loss for point prediction. To begin, it reformulates and re-parameterizes the GEV parameter constraints to make them compatible with deep learning architecture and for restricting the model output such that it complies with the GEV constraints. Also, it offers a model bias offset technique to initialize the DNN so as to produce a feasible set of GEV parameters.

In summary, the main contributions of this paper are:

1. We propose a novel framework to predict extreme events by incorporating GEV distribution from extreme value theory into deep learning architecture.

2. We propose a reformulation of the GEV maximum log-likelihood estimation (MLE) constraint to make it compatible with deep learning. We also develop a reparametrization technique for constraining the model output in such a way that it satisfies the GEV constraints.

3. We propose a simple model initialization technique to ensure the neural network output is initialized correctly.

4. We demonstrate the effectiveness of the proposed approach on real-world and synthetic datasets.

Related Work

Deep learning architectures are widely used in time series forecasting due to their immense success in learning nonlinear relationships. In particular, several deep learning architectures are capable of learning underlying complex temporal dependencies. The majority of the research is concentrated on point estimation and often utilizes architectures based on LSTMs (Hochreiter and Schmidhuber 1997) or convolutional neural networks (LeCun, Bengio et al. 1995).

Long short-term memory networks were proposed by (Hochreiter and Schmidhuber 1997) to overcome the vanishing and exploding gradient problem in other recurrent neural networks. Since its inception, LSTM and its variants have been successfully applied to forecasting time series in various application domains. For example, (Masum, Liu, and Chiverton 2018) used LSTM based model to predict time series electric load. (Peng et al. 2018) employed an encoder-decoder-based GRU, which is a variation of the LSTM architecture, in multi-step host load prediction in cloud computing utilizing an encoder-decoder based approach. (Sagheer and Kotb 2019) proposed a variation of LSTM in time series prediction of petroleum production.

Some of the earliest successful applications of neural networks have involved convolutional neural networks. In applications to time series, a small filter is passed over the temporally localized region to model temporal relationships. (Yang et al. 2015) proposed a CNN architecture for activity recognition. (Zhao et al. 2017) employed a CNN-based architecture for time series classification which they evaluate on 8 real-world data sets. (Bai, Kolter, and Koltun 2018) empirically demonstrated the effectiveness of CNNs as alternatives to LSTMs in sequence-based modeling.

For uncertainty quantification in the time series problem, (Wang et al. 2020) proposed a distribution-free novel approach called DeepPIPE. It simultaneously predicted point estimations as well as quantile estimations without any prior assumption about the data distribution. They proposed a hybrid loss function based on point estimations and point intervals to leverage point and quantile estimations. Though it might be a sound baseline for quantile estimation, it did not incorporate EVT theory.

Kharin et al (Kharin and Zwiers 2005) is representative of much of the traditional statistical work utilizing EVT. They analyzed GEV parameters assuming there was a simple relationship between those parameters and a single predictor, i.e., time. As previously mentioned, prior work combining deep learning with EVT suffers from significant limitations. For instance, (Ding et al. 2019) incorporated the GP distribution in their loss function to forecast excesses over a threshold. However, instead of predicting the GP parameters from covariates, they assume the GP parameters values are known *a-priori* and can be provided as hyperparameters of their algorithm. Thus, the GP parameters are assumed to be fixed (constant) for all-time series, which is a strong assumption especially if the time series is generated for different locations. (Polson and Sokolov 2020) proposed to combine deep learning model with extreme value theory to model tail behavior of a time series. They applied the GP distribution for modeling excess values and used its negative log-likelihood

as the loss function. However, a major issue with their proposed framework is that it does not incorporate a mechanism to enforce constraints on parameters of the learned GP distribution, which are essential to ensure the predicted distribution is well-behaved.

Preliminaries

Let $z_1, z_2, \cdots, z_T$ be a time series of length T. Assume we can partition the time series into a set of predictor, $x_t = (z_{t-\alpha}, z_{t-\alpha+1}, \cdots, z_t)$, and target, $\tilde{y}_t = (z_{t+1}, z_{t+2}, \cdots, z_{t+k})$ windows, where k is the forecast horizon. For each target window, let $y_t = \max_{\tau \in \{1,\cdots,k\}} z_{t+\tau}$ be the block maxima of the target variable at time t. The goal of our time series forecasting task is to estimate the block maxima value, $\hat{y}_t$, of a future time window based on current and past data, x_t; as well as upper and lower quantile estimations, $\hat{y}_U$ and $\hat{y}_L$.

Extreme Value Theory (EVT)

Extreme value theory is concerned with the limiting probability distribution of a random variable that deviates significantly from its median. The generalized extreme value (GEV) and generalized Pareto (GP) distributions are the two most commonly studied distributions in EVT. The GEV distribution models the block maxima, while the GP distribution models the distribution of excesses over a given threshold. To avoid the difficulties associated with the choice of GP threshold, this paper focuses on the GEV distribution.

According to EVT (Coles et al. 2001), if there exist sequences of constants $a_n > 0$ and b_n such that

$$Pr(M_n - b_n)/a_n \leq y \rightarrow G(y) \quad \text{as } n \rightarrow \infty$$

for a non-degenerate distribution G, then

$$G(y) = exp\{-[1 + \xi(\frac{y-\mu}{\sigma})]^{-1/\xi}\} \tag{1}$$

which defines a family of CDF functions that governs the GEV distribution. Here, GEV parameters: location is denoted by μ; scale is denoted by σ; and shape is denoted by ξ. From the CDF of GEV (1), quantile and log-likelihood function can be easily derived . The *pth* quantile of the GEV distribution, y_p can be calculated as .

$$y_p = \mu + \frac{\sigma}{\xi}[(-\log p)^{-\xi} - 1] \tag{2}$$

and the log-likelihood function of GEV with respect to parameters $\mu, \sigma,$ and ξ is as follows:

$$\log L(\mu, \sigma, \xi) = -n \log \sigma - (\frac{1}{\xi} + 1) \sum_{i=1}^{n} \log(1 + \xi \frac{y_i - \mu}{\sigma}) - \sum_{i=1}^{n} (1 + \xi \frac{y_i - \mu}{\sigma})^{-1/\xi} \quad (\text{if } \xi \neq 0) \tag{3}$$

subject to the following positivity constraints:

$$\sigma > 0 \quad \text{and} \quad \forall i : 1 + \frac{\xi}{\sigma}(y_i - \mu) > 0 \tag{4}$$

Minimizing the log-likelihood function (3) can be used to estimate the GEV parameters: $\mu, \sigma,$ and ξ.

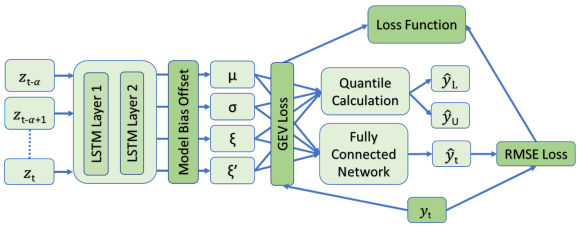


Figure 1: Proposed `DeepExtrema` framework for predicting block maxima using GEV distribution.

Proposed Framework: DeepExtrema

This section presents the proposed framework, `DeepExtrema`, for characterizing and predicting the block maxima using a GEV distribution. Our goal is to predict the distribution of the block maxima conditioned on observations of our predictors. We assume that the GEV parameters are related in a potentially non-linear way to the predictors x. Formally, our goal is to estimate the following functions:

$$\mu = f_\mu(x), \quad \sigma = f_\sigma(x), \quad \xi = f_\xi(x) \tag{5}$$

Figure 1 provides an overview of the architecture for `DeepExtrema`. Given an input x, the framework uses a Long short-term memory (LSTM) architecture to capture temporal dependencies of the time series. The LSTM will output a latent representation to be used by a fully connected network to generate the GEV parameters. The proposed Model Bias Offset (MBO) component estimates the GEV parameters primarily to ensure that the network has an appropriate initialization. The GEV parameters are subsequently fed into a fully connected network to obtain a point estimate of the block maxima. The GEV parameters are also used to calculate GEV loss. The root-mean-square error (RMSE) of the predicted block maxima is also computed and incorporated into the final loss function. Finally, using the GEV parameters, predictions can be generated for different quantiles using the formula given in Equation 2. Details of the different components are described below.

GEV Parameter Estimation

The maximum likelihood estimation of the GEV parameters (when $\xi \neq 0$) has two positivity constraints, as shown by the inequalities in (4). While the first positivity constraint on σ is straightforward to enforce, maintaining the second one is harder as it involves a nonlinear relationship among the predicted values of the GEV parameters, ξ, μ, and σ. The GEV parameters may vary from one input x to another, and thus, learning them from the limited training samples is also a challenge. Worse still, some of the estimated GEV parameters could be erroneous, especially in the early rounds of training, making it difficult to preserve the constraints throughout the learning process. In addition, ξ can be either positive or negative, which further complicates matters, especially when trying to enforce the second constraint in (4). To address these challenges, we propose a reformulation of the second constraint in (4). The positivity constraint

on σ and the reformulation of the second constraint in (4) are then re-parameterized to ensure the predicted GEV parameter values satisfy the constraints.

To ensure that the constraints are still satisfied even if some of the GEV parameters (μ, σ, or ξ) are incorrectly predicted, we relax the second constraint in (4), by adding a tolerance factor (slack variable), τ, as follows:

$$\forall\, i : 1 + \frac{\xi}{\sigma}(y_i - \mu) + \tau > 0. \tag{6}$$

The tolerance factor allows for a minor violation of the second constraint in (4) as long as $1+\frac{\xi}{\sigma}(y_i-\mu) > -\tau$. With the given tolerance factor, the following theorem can be used to provide an upper and lower bound on ξ:

Theorem 1. *Assuming $\xi \neq 0$, the soft constraint in* (6) *can be reformulated into the following bounds on ξ:*

$$-\frac{\sigma}{y_{\max} - \mu}\,(1+\tau) < \xi < \frac{\sigma}{\mu - y_{\min}}\,(1+\tau) \tag{7}$$

where τ is the tolerance on the constraint in 4

Proof. Theorem 1 can be proven using (6) when $\xi \neq 0$. Let $y_{\max} = \max_i y_i$ and $y_{\min} = \min_i y_i$. To obtain the lower bound on ξ, we set y_i to be $y_{\max}$ in (6):

$$\begin{aligned} 1 + \frac{\xi}{\sigma}(y_{\max} - \mu) + \tau > 0 \implies \frac{\xi}{\sigma}(y_{\max} - \mu) &> -(1+\tau) \\ \implies \xi &> -\frac{\sigma}{(y_{\max} - \mu)}\,(1+\tau) \end{aligned}$$

Conversely, to obtain the upper bound on ξ, we set y_i to be $y_{\min}$ in (6):

$$\begin{aligned} 1 + \frac{\xi}{\sigma}(y_{\min} - \mu) + \tau > 0 \implies -\frac{\xi}{\sigma}(\mu - y_{\min}) &> -(1+\tau) \\ \implies \xi &< \frac{\sigma}{(\mu - y_{\min})}\,(1+\tau) \end{aligned}$$

□

Following Theorem 1, the upper and lower bound constraints on ξ in (7) can be restated as follows:

$$\begin{aligned} \frac{\sigma}{\mu - y_{\min}}\,(1+\tau) - \xi &> 0 \\ \xi + \frac{\sigma}{y_{\max} - \mu}\,(1+\tau) &> 0 \end{aligned} \tag{8}$$

The reformulation imposes lower and upper bounds on ξ, which can be used to re-parameterize the second constraint in (4). `DeepExtrema` employs the following approach to enforce the positivity constraints in (8).

Given an input x, `DeepExtrema` outputs the following four parameters: μ, P_1, P_2, and P_3. A softplus activation function (Torch Contributors 2019), $softplus(x) = \log(1 + \exp(x))$, which is a smooth approximation to the ReLU function, is used to enforce the non-negativity constraints associated with the GEV parameters. Using Softplus, σ is re-parameterized as follows:

$$\sigma = softplus(P_1) \tag{9}$$

while ξ is re-parameterized into its upper and lower bound estimates, P_2 and P_3:

$$\begin{aligned} \frac{\sigma}{\mu - y_{\min}}\,(1+\tau) - \xi_u &= softplus(P_2) \\ \frac{\sigma}{y_{\max} - \mu}\,(1+\tau) + \xi_l &= softplus(P_3) \end{aligned} \tag{10}$$

By re-arranging the above equation, we obtain

$$\begin{aligned} \xi_u &= \frac{\sigma}{\mu - y_{\min}}\,(1+\tau) - softplus(P_2) \\ \xi_l &= softplus(P_3) - \frac{\sigma}{y_{\max} - \mu}\,(1+\tau) \end{aligned} \tag{11}$$

`DeepExtrema` uses both formulas in (10) to generate a pair of ξ estimates for each data point. It then tries to minimize the distance between ξ_u and ξ_l. Upon training, both $\xi's$ will converge into a single value as they are derived from Theorem 1.

Model Bias Offset (MBO)

Employing a deep neural network with a negative GEV log-likelihood loss function is challenging due to the initialization of the network. Here, two issues may occur. First, the initial values of the parameters may violate the constraints as the initial values may not be consistent with the data, more specifically, with the $y_{\min}$ and the $y_{\max}$ used in the reformulated constraint. Second, even if it does not violate a constraint, it may produce unreasonable values which are inconsistent with GEV maximum likelihood estimation. For instance, the maximum likelihood estimates of the GEV parameter ξ have usual asymptotic properties when $\xi > -0.5$, but for $-1 < \xi < -0.5$, MLE estimates are obtainable but do not have standard asymptotic properties, and for $\xi < 1$, MLE estimates do not exist (Coles et al. 2001). Also, for $\xi > 1$, the mean is not defined. Given a random initialization of the network, it is likely that these conditions will be violated for some samples and we find empirically that the network struggles to converge to good solutions when this occurs. So, the initial output of the network needs to be reasonable to maintain constraints. Unfortunately, controlling the initial output of a neural network is difficult, especially in a complex architecture.

To address this problem, a simple but effective technique called Model Bias Offset (MBO) is introduced, as depicted in Figure 2. Let $\mu_{\text{desired}}, \sigma_{\text{desired}}, \xi_{\text{desired}}$, and ξ'_{desired} be vectors representing the reasonable initial GEV parameters for the first mini-batch of data in the network. We find in practice that fitting all the $y's$ into a single global GEV distribution, with a common μ, σ, and ξ for all data points, works well in practice as the initial desired GEV parameters. With these common μ, σ, and ξ serving as reasonable parameters for all $y's$ thus making effective values of $\mu_{\text{desired}}, \sigma_{\text{desired}}, \xi_{\text{desired}}$, and ξ'_{desired}.

Using MBO, at first, the network is randomly initialized as usual and the input (only the first batch of the first iteration) is passed to the network like a standard forward pass to obtain initial predictions for our GEV parameters which we denote μ_0, σ_0, ξ_0, and ξ'_0. However, no loss is computed and no backward pass is performed with this pass.

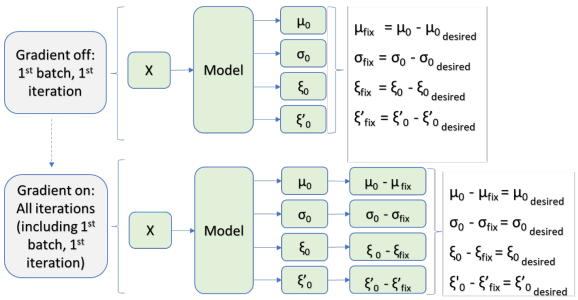


Figure 2: Model Bias Offset (DBO): a simple but effective technique to get desired initial output from a deep neural network

.

Using the initial output of the forward pass, the fixed vectors: $\mu_{\text{fix}}, \sigma_{\text{fix}}, \xi_{\text{fix}},$ and ξ'_{fix} are calculated as follows:

$$\mu_{\text{fix}} = \mu_0 - \mu_{\text{desired}} \qquad \sigma_{\text{fix}} = \sigma_0 - \sigma_{\text{desired}}$$
$$\xi_{\text{fix}} = \xi_0 - \xi_{\text{desired}} \qquad \xi'_{\text{fix}} = \xi'_0 - \xi'_{\text{desired}} \tag{12}$$

From then on, using (12), the fixed vectors are subtracted from the network output (parameters) to get the desired output:

$$\mu_0 - \mu_{\text{fix}} = \mu_{\text{desired}} \qquad \sigma_0 - \sigma_{\text{fix}} = \sigma_{\text{desired}}$$
$$\xi_0 - \xi_{\text{fix}} = \xi_{\text{desired}} \qquad \xi'_0 - \xi'_{\text{fix}} = \xi'_{\text{desired}} \tag{13}$$

The fixed vectors (12) are calculated at the very beginning (first batch; first iteration) and remained constant throughout training and evaluation. They can be regarded as the initial model bias that needs to be offset at every subsequent iterations. By offsetting the output of the DNN in this way, we guarantee that the initial GEV parameters are reasonable and satisfy the GEV constraints.

Loss Function

The GEV maximum log-likelihood estimation function (3) can be used to learn the GEV parameters. Maximizing the log-likelihood gives the optimal GEV parameters, which is equivalent to minimizing the negative log-likelihood using a deep neural network. `DeepExtrema` employs the GEV negative log-likelihood as the basis of its loss function. Also, `DeepExtrema` combines both the GEV negative log-likelihood loss as well as the RMSE loss of the $\hat{y}$ and y. First, it calculates loss from the GEV theory. It also combines the RMSE of the two $\xi's$ generated from the network. Here, ξ is extracted from the upper bound constraint on ξ: first formula of (11). And, ξ' is extracted from the lower bound constraint on ξ: second formula of (11). Then, using the GEV loss, it calculates the overall loss by combining the RMSE of the estimations and the ground truths:

$$Loss_{GEV} = \lambda_1 * (-\log L(\mu, \sigma, \xi)) + (1 - \lambda_1) * \frac{1}{n} \sum_{i=1}^{n} (\xi_i - \xi'_i)^2 \tag{14}$$

$$Loss = \lambda_2 * Loss_{GEV} + (1 - \lambda_2) * \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{15}$$

Here, λ_1 and λ_2 are hyperparameters which denote the weights of the various terms of the loss function.

Experimental Evaluation

This section presents the experimental results comparing `DeepExtrema` against various baseline methods.

Data

Synthetic Data As the ground truth GEV parameters are often unknown for real-world data, we have created synthetic data to evaluate the performance of various methods in terms of their ability to correctly infer parameters of the GEV distribution. The synthetic data is generated assuming the GEV parameters are functions of some input predictors $x \in \mathbb{R}^d$ (where $d = 6$). We first generate x by random sampling from a uniform distribution. We then assume a nonlinear mapping from x to the GEV parameters μ, σ, and ξ, via the following nonlinear equations:

$$\mu(x) = w_\mu^T(\exp(x) + x)$$
$$\sigma(x) = w_\sigma^T(\exp(x) + x)$$
$$\xi(x) = w_\xi^T(\exp(x) + x) \tag{16}$$

where w_μ, w_σ, and w_ξ are generated from a standard normal distribution. Using the generated μ, σ, and ξ parameters, we then randomly sample y from the GEV distribution governed by the GEV parameters. Here, y denotes the block maxima as it is generated from a GEV distribution. We then train the different methods on the synthetic data, $\{(x_i, y_i)\}_{i=1}^N$, where $N = 8192$.

Real-world data: Forecasting the intensity (i.e., maximum sustained wind speed) of a hurricane is challenging due to its fluctuating nature. As the strong hurricane-force winds can cause extensive damages and even loss of lives, accurate forecasting of the block maxima intensity is critical. The best track database (HURDAT2) (Landsea and Franklin 2013) from the National Hurricane Center is used in this study. The database contains hurricane data of the Atlantic region (from 1851-2019) and Northeast & North Central Pacific region (from 1949-2019). There are altogether 3,111 distinct hurricanes in the given period. For each hurricane, wind speeds (intensities) were reported at every 6-hour interval. The number of 6-hourly time steps available for each hurricane is not fixed, ranging from 1 to 132-time steps. On average, each hurricane has 26 6-hourly time steps, with a standard deviation of 16. As the NHC best track data consists of variable length time steps, they have to be pre-processed carefully to ensure consistency. We employ a moving time window approach to generate more samples from each hurricane. Apart from the best track data, NHC also provides their official forecasts, which will be used as a gold standard for comparison. The NHC official forecasts

RMSE of GEV Parameters		
μ	σ	ξ
0.046	0.029	0.034

Table 1: RMSE of GEV parameters using synthetic data

Negative Log-likelihood		
DeepExtrema	Ground Truth	Global GEV Estimate
4410	4451	4745

Table 2: Negative log-likelihood of `DeepExtrema` with respect to ground truth and global parameter estimation using synthetic data

from 2012 to 2020 were used to compare against various competing methods. We consider only hurricanes that have at least 24-time steps at minimum for our experiments. For each hurricane, we have created non-overlapping time windows of length 24 time steps (6 days). We use the first 16 time steps (4 days) in the window as the predictor variables and the block maxima of the last 8 time steps (2 days) as the target variable.

Experimental Setup

For evaluation purposes, we split the data into separate training, validation, testing with a ratio of 7:2:1. The data is standardized to have zero mean and unit variance. We compare `DeepExtrema` against the following baseline methods: Persistence, fully-connected network (FCN), LSTM, and DeepPIPE (Wang et al. 2020). Root mean squared error (RMSE) of the predicted block maxima and negative log-likelihood of the data (for synthetic data only), correlation of the predictions and ground truth values, and prediction interval coverage probability (PICP) (Wang et al. 2020) are used to evaluate the performance of the different methods. PICP is given by the ratio of the number of captured data points to the total data points. It denotes the number of block maxima (ground truth) values that are within the upper and lower quantile predictions. Finally, hyperparameter tuning is performed by assessing the model performance on the validation set.

Experimental Results

Results on Synthetic Data: In this experiment, we have compared the performance of `DeepExtrema` against using a single (global) GEV parameter to fit the data. Based on the results shown in Table 2, `DeepExtrema` achieves a significantly lower negative log-likelihood of 4410 compared to the negative log-likelihood for global GEV estimate, which is 4745. This result supports the assumption that each block maxima comes from different GEV distributions rather than a single (global) GEV distribution. The results also suggest that the negative log likelihood estimated by `DeepExtrema` is lower than that for the ground truth.

Results on Real-world Data: Evaluation on real-world data shows that `DeepExtrema` outperforms other baseline methods used for comparison (see Table 3). Not only it does well on the RMSE of block maxima, but it also achieves a high correlation. It exceeds the 80% confidence interval with 92% PICP: 92% of the block maxima are inside the upper and lower quantile estimations. Though DeepPIPE has higher PICP due to their architecture design, their RMSE of block maxima and correlation values are worse.

Finally, the RMSE of the NHC official forecasts is 11.35 while its correlation is 0.92. This suggests that our proposed model, which uses only historical data, can achieve comparable performance as the gold standard. This is not surprising since the official NHC forecasts use output from an ensemble of dynamical and statistical models along with expert's knowledge to generate its forecasts whereas `DeepExtrema` uses only historical information of hurricane intensities at previous time steps as its predictors. The performance of the official NHC forecasts would give an estimate of how far off the deep learning models perform compared to the gold standard despite using far less information.

Methods	RMSE of block maxima	Correlation
Persistence	28.6	0.60
FCN	14.14	0.87
LSTM	13.31	0.88
DeepPIPE	13.67	0.87
DeepExtrema	**12.80**	**0.90**

Table 3: Performance comparison on real world data.

To demonstrate how well the model predicts the hurricane intensity, Figure 3 shows a scatter plot of the point estimate (actual vs estimated hurricane intensity) generated by `DeepExtrema` on the test data. The results suggest that `DeepExtrema` can accurately predict the hurricane intensities, especially those below 140 knots. Figure 4 shows the 90% confidence interval of the predictions. Apart from the point and quantile estimations, `DeepExtrema` can also estimate the GEV parameter values for each hurricane. Validation of these estimations is not possible, but the following figures provide important insights into characteristics of the predicted distribution of the hurricane intensities.

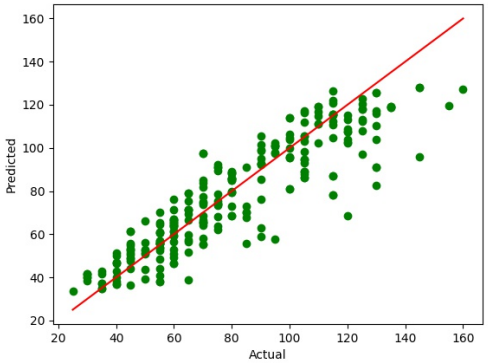


Figure 3: Comparison between actual and predicted block maxima of hurricane intensities for `DeepExtrema`.

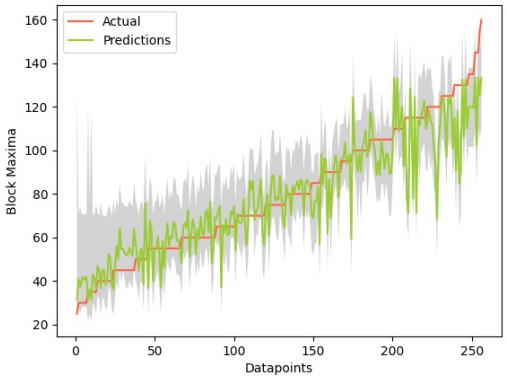


Figure 4: 90% confidence interval of the hurricane intensity predictions for `DeepExtrema`, sorted in increasing block maxima values. Ground truth values are shown in red.

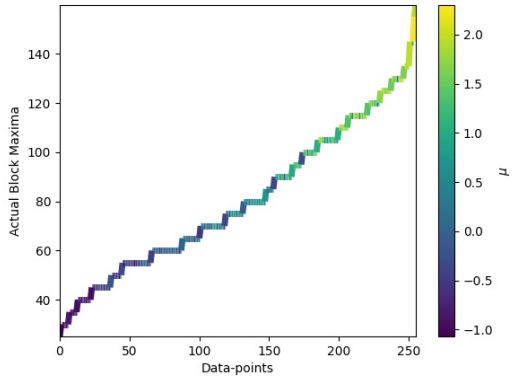


Figure 6: Magnitude of μ with respect increasing intensities of hurricane.

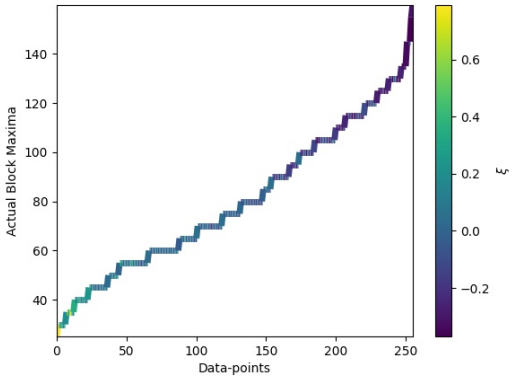


Figure 5: Magnitude of ξ with respect increasing intensities of hurricane.

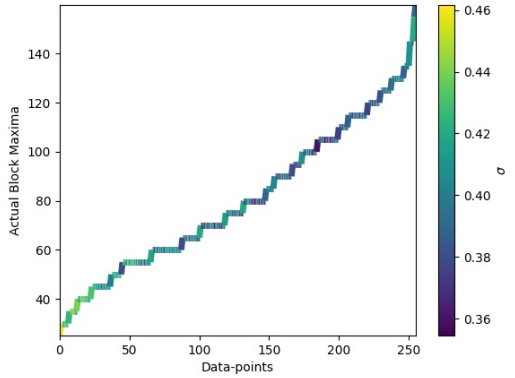


Figure 7: Magnitude of σ with respect increasing intensities of hurricane.

According to Figure 5, the magnitude of ξ decreases in the upper tail part of the distribution, which is consistent our expectation. Similarly, Figures 6 and 7 show an increasing trend in μ and a decreasing trend in σ as the block maxima value of the hurricane intensity increases. So, tail end extremes, i.e., upper extremes have much larger μ, smaller σ, and lower ξ. The observed trend in GEV parameters enable us to better characterize properties of the distribution of hurricane intensities.

Ablation Studies The hyperparameter λ_1 in the objective function of `DeepExtrema` denotes the trade-off between GEV loss and RMSE loss. Experimental results show that with the decrease of GEV loss weight, the RMSE of block maxima also decreases (Table 4). It suggests incorporation of GEV theory plays a positive role in block maxima estimations rather than using mean squared-based loss function.

Weights on GEV Loss	$\lambda_1 = 0.9$	$\lambda_1 = 0.5$	$\lambda_1 = 0.0$
RMSE (Block maxima)	**12.80**	13.01	13.28

Table 4: Effects of different weights on GEV Loss.

Conclusion

This paper presents a novel deep learning framework (`DeepExtrema`) that combines extreme value theory with deep learning to address the challenges of predicting extremes in time series. We offer a reformulation and reparameterization technique for satisfying constraints as well as a model bias offset technique for proper model initialization. We evaluated our framework on synthetic and real-world data and showed its effectiveness. For future work, we plan to extend the formulation to enable more complex deep learning architectures such as those using an attention mechanism. In addition, the framework will be extended to model extremes in spatio-temporal data.

References

Aliabadi, M. M.; Emami, H.; Dong, M.; and Huang, Y. 2020. Attention-based recurrent neural network for multistep-ahead prediction of process performance. *Computers & Chemical Engineering*, 140: 106931.

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.

Coles, S.; Bawa, J.; Trenner, L.; and Dorazio, P. 2001. *An introduction to statistical modeling of extreme values*, volume 208. 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.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

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

Koenker, R.; and Machado, J. A. 1999. Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, 94(448): 1296–1310.

Landsea, C. W.; and Franklin, J. L. 2013. Atlantic hurricane database uncertainty and presentation of a new database format. *Monthly Weather Review*, 141(10): 3576–3592.

LeCun, Y.; Bengio, Y.; et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995.

Masum, S.; Liu, Y.; and Chiverton, J. 2018. Multi-step time series forecasting of electric load using machine learning models. In *International conference on artificial intelligence and soft computing*, 148–159. Springer.

Peng, C.; Li, Y.; Yu, Y.; Zhou, Y.; and Du, S. 2018. Multi-step-ahead host load prediction with gru based encoder-decoder in cloud computing. In *2018 10th International Conference on Knowledge and Smart Technology (KST)*, 186–191. IEEE.

Polson, M.; and Sokolov, V. 2020. Deep learning for energy markets. *Applied Stochastic Models in Business and Industry*, 36(1): 195–209.

Sagheer, A.; and Kotb, M. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323: 203–213.

Torch Contributors. 2019. SOFTPLUS. https://pytorch.org/docs/stable/generated/torch.nn.Softplus.html. Accessed: 2021-04-09.

US GAO. 2020. Natural Disasters: Economic Effects of Hurricanes Katrina, Sandy, Harvey, and Irma. https://www.gao.gov/products/gao-20-633r. Accessed: 2021-04-09.

Wang, B.; Li, T.; Yan, Z.; Zhang, G.; and Lu, J. 2020. DeepPIPE: A distribution-free uncertainty quantification approach for time series forecasting. *Neurocomputing*, 397: 11–19.

Yang, J.; Nguyen, M. N.; San, P. P.; Li, X. L.; and Krishnaswamy, S. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*.

Zhang, X.; Liang, X.; Zhiyuli, A.; Zhang, S.; Xu, R.; and Wu, B. 2019. AT-LSTM: An attention-based LSTM model for financial time series prediction. In *IOP Conference Series: Materials Science and Engineering*, volume 569, 052037. IOP Publishing.

Zhao, B.; Lu, H.; Chen, S.; Liu, J.; and Wu, D. 2017. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1): 162–169.