Modeling Extreme Events in Time Series Prediction

Daizong Ding, Mi Zhang*
Fudan University, China
{17110240010,mi_zhang}@fudan.edu.cn

Xudong Pan, Min Yang
Fudan University, China
{18110240010,m_yang}@fudan.edu.cn

Xiangnan He
University of Science and Technology of China, China
xiangnanhe@gmail.com

ABSTRACT

Time series prediction is an intensively studied topic in data mining. In spite of the considerable improvements, recent deep learning-based methods overlook the existence of *extreme events*, which result in weak performance when applying them to real time series. Extreme events are rare and random, but do play a critical role in many real applications, such as the forecasting of financial crisis and natural disasters. In this paper, we explore the central theme of improving the ability of deep learning on modeling extreme events for time series prediction.

Through the lens of formal analysis, we first find that the weakness of deep learning methods roots in the conventional form of quadratic loss. To address this issue, we take inspirations from the Extreme Value Theory, developing a new form of loss called Extreme Value Loss (EVL) for detecting the future occurrence of extreme events. Furthermore, we propose to employ Memory Network in order to memorize extreme events in historical records. By incorporating EVL with an adapted memory network module, we achieve an end-to-end framework for time series prediction with extreme events. Through extensive experiments on synthetic data and two real datasets of stock and climate, we empirically validate the effectiveness of our framework. Besides, we also provide a proper choice for hyper-parameters in our proposed framework by conducting several additional experiments.

CCS CONCEPTS

• **Mathematics of computing** → *Probabilistic algorithms*; • **Computing methodologies** → *Neural networks*;

KEYWORDS

Extreme Event, Memory Network, Attention Model

1 INTRODUCTION

Time series prediction as a classical research topic, has been intensively studied by interdisciplinary researchers over the past several decades. As its application increasingly ventures into safety-critical real-world scenarios, such as climate prediction [35] and stocks price monitoring [16], how to obtain more accurate predictions remains an open problem to solve.

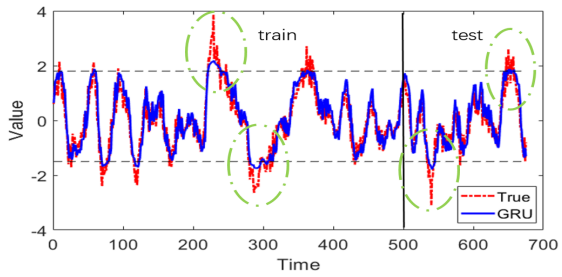
Historically, traditional methods such as autoregressive moving average (ARMA) [46] and nonlinear autoregressive exogenous (NARX) [31] use statistical models with few parameters to exploit patterns in time series data. Recently, with the celebrated success of Deep Neural Network (DNN) in many fields such as image classification [28] and machine translation [4], a number of DNN based techniques have been subsequently developed for time-series prediction tasks, achieving noticeable improvements over traditional methods [11, 49]. As a basic component of these models, Recurrent Neural Network (RNN) module serves as an indispensable factor for these note-worthy improvements [31, 48]. Compared with traditional methods, one of the major advantages of RNN structure is that it enables deep non-linear modeling of temporal patterns. In recent literature, some of its variants show even better empirical performance, such as the well-known Long-Short Term Memory (LSTM) [22, 36, 50] and Gated Recurrent Unit (GRU) [10], while the latter appears to be more efficient on smaller and simpler dataset [10]. However, most previously studied DNN are observed to have troubles in dealing with data imbalance [15, 42, 44]. Illustratively, let us consider a binary classification task with its training set including 99% positive samples and only 1% negative samples, which is said to contain *data imbalance*. Following the discussion in Lin et al., such an imbalance in data will potentially bring any classifier into either of two unexpected situations: **a.** the model hardly learns any pattern and simply chooses to recognize all samples as positive. **b.** the model memorizes the training set perfectly while it generalizes poorly to test set.

In fact, we have observed that, in the context of time-series prediction, imbalanced data in time series, or *extreme events*, is also harmful to deep learning models. Intuitively, an extreme event in time series is usually featured by extremely small or large values, of irregular and rare occurrences [24]. As an empirical justification of its harmfulness on deep learning models, we train a standard GRU to predict one-dimensional time series, where certain thresholds are used to label a small proportion of datasets as extreme events in prior (dotted line in Fig 1). As clearly shown, the learning model indeed falls into the two priorly discussed situations: **a.** In Fig. 1(a), most of its predictions are bounded by thresholds and therefore it fails to recognize future extreme events, we claim this as *underfitting phenomenon*. **b.** In Fig. 1(b), although the model learns extreme events in the train set correctly, it behaves poorly on test sets, we cliam this as *overfitting phenomenon*. Previously, people always tend to tolerate the underfitting phenomenon since models would still have an averagely tolerable performance on test sets. However from our perspective, it would be really valuable if a time-series prediction model could recognize future extreme events with reasonable predictions. With more accurate modeling

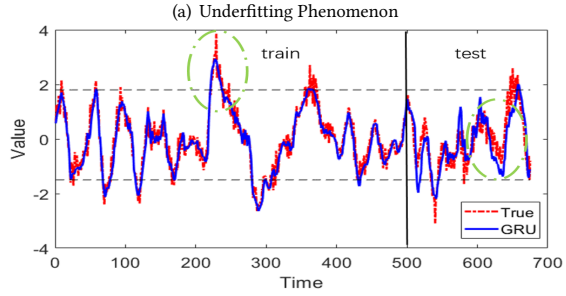
*The corresponding author is Mi Zhang <mi_zhang@fudan.edu.cn>

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330896



(a) Underfitting Phenomenon



(b) Overfitting Phenomenon

Figure 1: The extreme event problem in time-series prediction. The data are sampled from climate dataset.

of extreme events in many real-world cases, prediction models are expected to aid influential decisions by providing alarms on future incidents such as extreme winds [35] or financial crisis [41].

With motivations above, in this paper, we focus on improving the performance of DNN on predicting time series with extreme values. First, besides the empirical validation above, we present a formal analysis on why DNN could easily fall into underfitting or overfitting phenomenons when it is trained with time series with extreme events. Through the lens of Extreme Value Theory (EVT), we observe that the main reason lies in previous choices of loss function, which inherently lacks the ability to model extreme events in a find-grained way. Therefore we propose a novel form of loss called Extreme Value Loss (EVL) to improve predictions on occurrences of extreme events. Furthermore, Inspired by previous studies on dynamics of extreme events, which pointed out that the randomness of extreme events have limited degrees of freedom (DOF) [33]. As a result, its patterns could indeed be memorized [2, 8]. We informatively propose a neural architecture to memorize extreme events from historical information, with the aid of Memory Network [45]. Together with our proposed EVL, our end-to-end framework is thus constructed for better predictions on time series data with extreme events. Our main contributions are

- We provide a formal analysis on why deep neural network suffers underfitting or overfitting phenomenons during predicting time series data with extreme value.
- We propose a novel loss function called Extreme Value Loss (EVL) based on extreme value theory, which provides better predictions on future occurrences of extreme events.
- We propose a brand-new Memory Network based neural architecture to memorize extreme events in history for better predictions of future extreme values. Experimental results validates the superiority of our framework in prediction accuracy compared with the state-of-the-arts.

2 PRELIMINARIES

In this section, we briefly describe the time-series prediction problem and introduce *extreme events* in time-series data.

2.1 Time Series Prediction

Suppose there are N sequences of fixed length T. For the i-th sequence the time series data can be described as,

$$(X_{1:T}^{(i)}, Y_{1:T}^{(i)}) = [(x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), \cdots, (x_T^{(i)}, y_T^{(i)})] \quad (1)$$

, where $x_t^{(i)}$ and $y_t^{(i)}$ are input and output at time t respectively. In one-dimensional time series prediction we have $x_t^{(i)}, y_t^{(i)} \in \mathbb{R}$ and $y_t^{(i)} := x_{t+1}^{(i)}$. For the sake of convenience, we will use $X_{1:T} = [x_1, \cdots, x_T]$ and $Y_{1:T} = [y_1, \cdots, y_T]$ to denote general sequences without referring to specific sequences.

The goal of time-series prediction is that, given observations $(X_{1:T}, Y_{1:T})$ and future inputs $X_{T:T+K}$, how to predict outputs $Y_{T:T+K}$ in the future. Suppose a model predicts o_t at time t given input x_t, the common optimization goal can be written as,

$$\min \sum_{t=1}^{T} \|o_t - y_t\|^2 \quad (2)$$

Then after the inference the model could predict the corresponding outputs $O_{1:T+K}$ give inputs $X_{1:T+K}$. Traditional methods such as autoregressive moving average model (ARMA) [46] and Nonlinear autoregressive exogenous (NARX)[31] predicts outputs by conducting linear or non-linear regression on past inputs. Recently, deep neural network such as Recurrent Neural Network (RNN) shows superior advantages compared with traditional methods in modeling time-series data. Numerous improvements have been made on RNN such as Long-short Term Memory [22] and Gated Recurrent Unit [9].

2.2 Extreme Events

Although DNN such as GRU has achieved noticeable improvements in predicting time-series data, this model tends to fall into either overfitting or underfitting if trained with imbalanced time series, as we have demonstrated in introductory part. We will refer to such a phenomenon as *Extreme Event Problem*. Towards a formal understanding of this phenomenon, it will be convenient to introduce an auxiliary indicator sequence $V_{1:T} = [v_1, \cdots, v_T]$:

$$v_t = \begin{cases} 1 & y_t > \epsilon_1 \\ 0 & y_t \in [-\epsilon_2, \epsilon_1] \\ -1 & y_t < -\epsilon_2 \end{cases} \quad (3)$$

where large constants $\epsilon_1, \epsilon_2 > 0$ are called *thresholds*. For time step t, if $v_t = 0$, we define the output y_t as *normal event*. If $v_t > 0$, we define the output y_t as *right extreme event*. If $v_t < 0$, we define the output y_t as *left extreme event*.

2.2.1 Heavy-tailed Distributions. There are many researches pay attention to these large observations in other tasks, e.g., previous work notices that empirical distribution of real-world data always appear to be *heavy-tailed* [37]. Intuitively, if a random variable Y is said to respect a *heavy-tailed distribution*, then it usually has a non-negligible probability of taking large values (larger than a threshold) [37]. In fact, a majority of widely applied distributions including

Table 1: Mathematical Notations

Symbol	Size	Description
$x_t^{(i)}$	$\mathbb{R}$	Input of time t in i-th sequence
$y_t^{(i)}$	$\mathbb{R}$	Output of time t in i-th sequence
$v_t^{(i)}$	$\{-1, 0, 1\}$	Extreme event indicator of time t in i-th sequence
N	$\mathbb{N}$	Number of sequences
T	$\mathbb{N}$	Train length of each sequence
H	$\mathbb{N}$	Size of latent factors in GRU
M	$\mathbb{N}$	Size of memory module
Δ	$\mathbb{N}$	Size of each window
o_t	$\mathbb{R}$	Prediction of time t in i-th sequence
h_t	$\mathbb{R}^H$	Hidden state from GRU at time t
w_j	$\mathbb{R}^\Delta$	Window j of memory network
s_j	$\mathbb{R}^H$	Latent representation of window j
q_j	$\{-1, 0, 1\}$	Extreme event indicator of window j
p_j	$[-1, 1]$	Prediction of extreme event indicator of window j
$\tilde{o}_t$	$\mathbb{R}$	Prediction from GRU part at time t
α_t	$\mathbb{R}^M$	Attentive weights at time t
u_t	$[-1, 1]$	Prediction of extreme event at time t

Gaussian, Poisson are not heavy-tailed, therefore, *light-tailed*. Only a few number of parametric distributions are heavy-tailed, e.g. Pareto distribution and log-Cauchy distribution. Therefore modeling with light-tailed parametric distributions would bring unavoidable losses in the tail part of the data. Such a statement can be illustratively presented with Fig. 2(a), where we choose a light-tailed truncated normal distribght-tailed distribution fits data around the center quite well, the inaccuracy on the tail part is intolerable.

2.2.2 Extreme Value Theory. Historically, Extreme Value Theory (EVT) take a further step on studying these heavy-tailed data. EVT studies the distribution of maximum in observed samples [43]. Formally speaking, suppose T random variables $y_1, \ldots, y_T$ are i.i.d sampled from distribution F_Y, then the distribution of the maximum is,

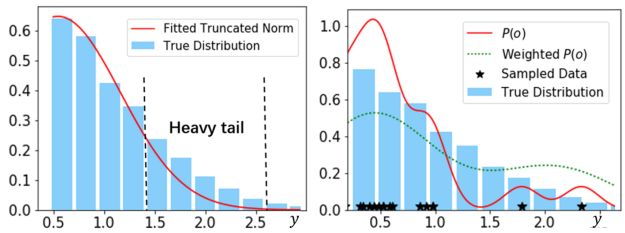
$$\lim_{T\to\infty} P\{\max(y_1, \cdots, y_T) \le y\} = \lim_{T\to\infty} F^T(y) = 0 \tag{4}$$

In order to obtain a non-vanishing form of $P\{\max(y_1, \cdots, y_T) \le y\}$, previous researches proceeded by performing a linear transformation on the maximum. As a fundamental result in EVT, the following theorem states that the distribution of Y after linearly transformed is always limited to few cases.

Theorem 2.1 ([17, 20]). *If there exists a linear transformation on Y which makes the distribution in Eq. 4 non-degenetated to 0. Then the class of the non-degenerated distribution $G(y)$ after the transformation must be the following distribution:*

$$G(y) = \begin{cases} \exp\left(-(1-\frac{1}{\gamma}y)^\gamma\right) & ,\gamma \neq 0, 1-\frac{1}{\gamma}y > 0 \\ \exp\left(-e^{-y}\right) & ,\gamma = 0 \end{cases} \tag{5}$$

Usually, the form $G(y)$ is called Generalized Extreme Value distribution, with $\gamma \neq 0$ as *extreme value index*. Such a statement sometimes is also regarded as the law of large numbers for the maximum [27]. In fact, the theorem above has a natural extension



(a) Illustration of Heavy Tail Distribution (b) Illustration of Optimized $P(o)$

Figure 2: Distributions of y_t in time-series data, where y_t are sampled from climate dataset as introduced in experiments.

to observations which exceed certain fixed threshold as follows, which would be useful in the next part.

2.2.3 Modeling The Tail. Previous works extend the above theorem to model the tail distribution of real-world data by [18, 47],

$$1 - F(y) \approx (1 - F(\xi))\left[1 - \log G\left(\frac{y-\xi}{f(\xi)}\right)\right], y > \xi \tag{6}$$

where $\xi > 0$ is a sufficiently large threshold.Previous researches point that the approximation in Eq. 6 can fit the tail distribution well [12]. Although there are many methods for modeling the distributions of extreme values [1], due to the rare and irregular essence of extreme events, it is always hard to forecast these pumping points [19]. What is worse, these extreme events could affect the learning of deep neural networks, where we will discuss the reason in detail in the next section.

3 PROBLEMS CAUSED BY EXTREME EVENTS

In this part we will deliver our explanation on why extreme event problem is almost inevitable for previously studied DNN models in time-series prediction.

3.1 Empirical Distribution After Optimization

We further investigate the influence of extreme events in time series prediction. For the sake of simplicity, we only pay our attention to one sequence, that is, $\boldsymbol{X}_{1:T}$ and $\boldsymbol{Y}_{1:T}$. From the probabilistic perspective, minimization of the loss function in Eq. 2 is in essence equivalent to the maximization of the likelihood $P(y_t|x_t)$. Based on Bregman's theory [5, 40], minimizing such square loss always has the form of Gaussian with variance τ, that is, $p(y_t|x_t, \theta) = \mathcal{N}(o_t, \tau^2)$, where θ is the parameter of the predicting model, $\boldsymbol{O}_{1:T}$ are outputs from the model.

Therefore, Eq. 2 can be replaced with its equivalent optimization problem as follows

$$\max_\theta \prod_{t=1}^{T} P\Big(y_t|x_t, \theta\Big) \tag{7}$$

With Bayes's theorem, the likelihood above can be written as,

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{8}$$

By assuming the model has sufficient learning capacity with parameter θ [23, 29], we claim the inference problem will yield an optimal approximation to $P(Y|X)$. It is worth to notice that our

assumption on learning capacity is a widely adopted assumption in previous researches [3, 21] and can be implemented with a deep neural network structure in practice. Furthermore, if $P(Y|X)$ has been perfectly learned, so as the distributions $P(Y), P(X), P(X|Y)$, which are therefore totally independent of inputs X. By considering the following observations,

- The discriminative model (Eq. 2) has no prior on y_t
- The output o_t is learned under likelihood as normal distribution

it is therefore reasonable to state that empirical distribution $P(Y)$ after optimization should be of the following form,

$$\hat{P}(Y) = \frac{1}{N}\sum_{t=1}^{T} \mathcal{N}(y_t, \hat{\tau}^2) \tag{9}$$

where constant $\hat{\tau}$ is an unknown variance. In consideration of its similarity to Kernel Density Estimator (KDE) with Gaussian Kernel [38], we can reach an intermediate conclusion that such a model would perform relatively poor if the true distribution of data in series is heavy-tailed, according to [7].

3.2 Why Deep Neural Network Could Suffer Extreme Event Problem

As discussed above, the distribution of output from a learning model with optimal parameters can be regarded as a KDE with Gaussian Kernel (Eq. 7).

Since nonparametric kernel density estimator only works well with sufficient samples, the performance therefore is expected to decrease at the tail part of the data, where sampled data points would be rather limited [7]. The main reason is that the range of extreme values are commonly very large, thus few samples hardly can cover the range. As depicted in Fig. 2(b), we sample y_t from the true distribution and fit a KDE with Gaussian Kernel. As is shown, since there are only two samples with $y_t > 1.5$, the shape of fitted KDE peaks inconsistently around these points. Moreover, as a large majority of samples are centered at 0, therefore the probability density around origin estimated by KDE tends to be much higher than the true distribution.

Formally, let us suppose x_1, x_2 are two test samples with corresponding outputs as $o_1 = 0.5, o_2 = 1.5$. As our studied model is assumed to have sufficient learning capacity for modeling $P(X), P(X|Y)$, thus we have

$$P(y_1|x_1,\theta) = \frac{P(X|Y)\hat{P}(Y)}{P(X)} \geq \frac{P(X|Y)P_{\text{true}}(Y)}{P(X)} = P_{\text{true}}(y_1|x_1) \tag{10}$$

Similarly $P(y_2|x_2,\theta) \leq P_{\text{true}}(y_2|x_2)$. Therefore, in this case, the predicted value from deep neural network are always bound, which immediately disables deep model from predicting extreme events, i.e. causes the *underfitting phenomenon*.

On the other side, as we have discussed in related work, several methods propose to accent extreme points during the training by, for example, increasing the weight on their corresponding training losses. In our formulation, these methods are equivalent to repeating extreme points for several times in the dataset when fitting KDE. Its outcome is illustrated by dot line in Fig. 2(b). As a consequence, we have

$$P(y_2|x_2,\theta) = \frac{P(X|Y)\hat{P}(Y)}{P(X)} \geq \frac{P(X|Y)P_{\text{true}}(Y)}{P(X)} = P_{\text{true}}(y_2|x_2) \tag{11}$$

Intuitively, the inequality above indicates, with the estimated probability of extreme events being added up, the estimation of normal events would simultaneously become inaccurate. Therefore, normal data in the test set is prone to be mis-classified as extreme events, which therefore marks the *overfitting phenomenon*.

As we can see, the extreme events problem in DNN is mainly caused by that there is no sufficient prior on tail part of observations y_t. Through maximizing likelihood could lead to a nonparametric estimation of y_t, which could easily cause *underfitting* problem. On the other side, if we increase the weight on those large values, DNN could easily suffer the *overfitting* problem. In order to alleviate these problems in DNN, we will provide an elegant solution, which aims at imposing prior on extreme events for DNN in predicting time series data.

4 PREDICTING TIME-SERIES DATA WITH EXTREME EVENTS

In order to impose prior information on tail part of observations for DNN, we focus on two factors: *memorizing extreme events* and *modeling tail distribution*. For the first factor we propose to use memory network to memorize the characteristic of extreme events in history, and for the latter factor we propose to impose approximated tailed distribution on observations and provide a novel classification called Extreme Value Loss (EVL). Finally we combine these two factors and introduce the full solution for predicting time series data with extreme values.

4.1 Memory Network Module

As pointed out by Ghil et al., extreme events in time-series data often show some form of temporal regularity [19]. Inspired from this point, we propose to use memory network to *memorize* these extreme events, which is proved to be effective in recognizing inherent patterns contained in historical information [45]. First, define the concept of *window* in our context.

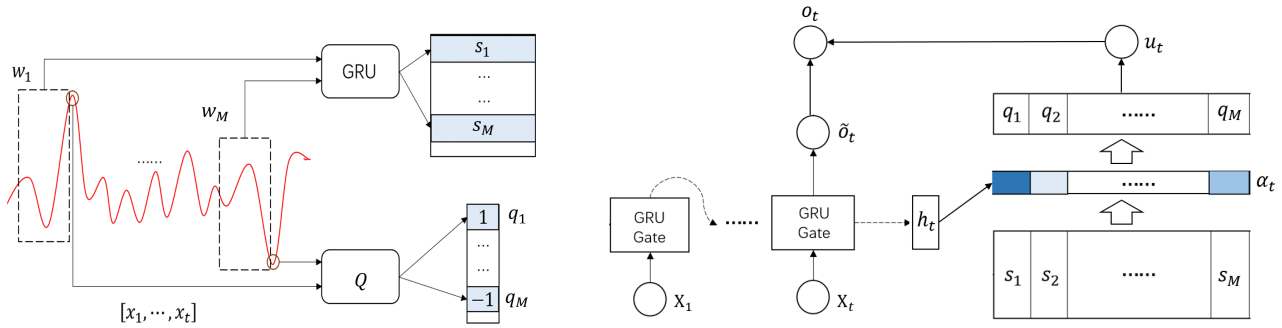
4.1.1 Historical Window. For each time step t, we first randomly sample a sequence of *windows* by $\mathcal{W} = \{w_1, \cdots, w_M\}$, where M is the size of the memory network. Each window w_j is formally defined as $w_j = [x_{t_j}, x_{t_j+1}, \cdots, x_{t_j+\Delta}]$, where Δ as the size of the window satisfying $0 < t_j < t - \Delta$.

Then we propose to apply GRU module to embed each window into feature space. Specifically, we use w_j as input, and regard the last hidden state as the latent representation of this window, denoted as $s_j = \text{GRU}([x_{t_j}, x_{t_j+1}, \cdots, x_{t_j+\Delta}]) \in \mathbb{R}^H$. Meanwhile, we apply a memory network module to memorize whether there is a extreme event in $t_j + \Delta + 1$ for each window w_j. In implementation, we propose to feed the memory module by $q_j = v_{t_j+\Delta+1} \in \{-1, 0, 1\}$.

For an overview of our memory network based module, please see Fig. 3(a). In summary, at each time step t, the memory of our proposed architecture consists of the following two parts:

- Embedding Module $S \in \mathbb{R}^{M\times H}$: s_j is the latent representation of history window j.
- History Module $Q \in \{-1, 0, 1\}^M$: q_j is the label of whether there is a extreme event after the window j.

4.1.2 Attention Mechanism. In this part, we further incorporate the module demonstrated above into our framework for imbalanced



(a) Illustration of Memory Network Module at time step t

(b) Attention mechanism for prediction.

Figure 3: Illustration of predicting process. For each time step t, we sample M windows and use GRU to build the memory network module.

time-series prediction. At each time step t, we use GRU to produce the output value:

$$\tilde{o}_t = W_o^T h_t + b_o, \quad \text{where } h_t = GRU([x_1, x_2, \cdots, x_t]) \tag{12}$$

, where h_t and s_j share the same GRU units. As we have discussed previously, the prediction of $\tilde{o}_t$ may lack the ability of recognizing extreme events in the future. Therefore we also require our model to retrospect its memory to check whether there is a similarity between the target event and extreme events in history. To achieve this, we propose to utilize attention mechanism [4] to our ends. Formally,

$$\alpha_{tj} = \frac{\exp(c_{tj})}{\sum_{j=1}^{M} \exp(c_{tj})}, \quad \text{where } c_{tj} = h_t^T s_j \tag{13}$$

Finally, the prediction of whether an extreme event would happen after referring historical information can be measured by imposing attentive weights on q_j. The output of our model at time step t is calculated as

$$o_t = \tilde{o}_t + b^T \cdot u_t, \quad \text{where } u_t = \sum_{j=1}^{M} \alpha_{tj} \cdot q_j \tag{14}$$

In the definition $u_t \in [-1, 1]$ is the prediction of whether there will be an extreme event after time step t, and $b \in \mathbb{R}_+$ is the scale parameter. Intuitively, the main advantage of our model lies in, it enables a flexible switch between yielding predictions of normal values and extreme values. When there is a similarity between the current time step and certain extreme events in history, then u_t will help detect such a pumping point by setting u_t non-vanishing, while when the current event is observed to hardly have any relation with the history, then the output would choose to mainly depend on $\tilde{o}_t$, i.e. the value predicted by a standard GRU gate. The loss function can be written as square loss defined in Eq. 2 in order to minimize the distance between output o_t and observation y_t.

4.2 Extreme Value Loss

Although memory network could forecast some extreme events, such loss function still suffer extreme events problem. Therefore we continue to model the second factor. As we have discussed in Sec. 3, the common square loss could lead to a nonparametric approximation on y_t. Without imposed prior $P(Y)$, the empirical estimation $\hat{P}(Y)$ could easily lead to two kinds of phenomenons. Therefore, in order to influence the distribution of $P(Y)$, we propose to impose

Algorithm 1 The proposed framework (Non-batch).

Input: $\boldsymbol{X}_{1:T} = [x_1, \cdots, x_T, x_{T+1}, \cdots, x_{T+K}]$, extreme indicator $\boldsymbol{V}_{1:T} = [v_1, \cdots, v_T]$, training output $\boldsymbol{Y}_{1:T} = [y_1, \cdots, y_T]$, hyper-parameters γ, memory size M, size of the window Δ, learning rate and regularization parameter.
Output: Predicted value $[o_{T+1}, \cdots, o_{T+K}]$
Initialize: Parameters θ, ϕ, ψ, b

```
procedure TRAIN(Sample i)
    for t = 1, ..., T do
        Calculating weights β in Eq. 16
        Construct memory module S, Q                    (Sec. 4.1)
        for j = 1, ..., M do
            Calculate p_j for window j.
        end for
        Minimizing loss function L_2(θ, ψ)              (Eq. 18)
        Calculate u_t and output o_t                    (Eq. 14)
    end for
    Minimizing loss function L_1(θ, φ, b).              (Eq. 17)
end procedure
function PREDICT(Sample i)
    Construct memory module S, Q with t ≤ T             (Sec. 4.1)
    for t = 1, ..., T + K do
        Calculate u_t and output o_t                    (Eq. 14)
    end for
    return {o_t}_{t=1}^{T+K}
end function
```

prior of tailed data on loss function. Rather than modeling the output o_t directly, we pay our attention to the extreme event indicator u_t. For the sake of simplicity, we first consider right extreme events.

In order to incorporate the tail distribution with $P(Y)$, we first consider the approximation defined in Eq. 6, which can approximate the tail distribution of observations. In our problem, for observation y_t, the approximation can be written as,

$$1 - F(y_t) \approx \big(1 - P(v_t = 1)\big) \log G\Big(\frac{y_t - \epsilon_1}{f(\epsilon_1)}\Big) \tag{15}$$

, where positive function f is the scale function. Further, if we consider a binary classification task for detecting right extreme events. In our model the predicted indicator is u_t, which can be regarded as a hard approximation for $(y_t - \epsilon_1)/f(\epsilon_1)$. We regard

the approximation as weights and add them on each term in binary cross entropy,

$$\begin{aligned} EVL(u_t) =& -\big(1-P(v_t=1)\big)\Big[\log G(u_t)\Big]v_t\log(u_t) \\ & -\big(1-P(v_t=0)\big)\Big[\log G(1-u_t)\Big](1-v_t)\log(1-u_t) \\ =& -\beta_0\Big[1-\frac{u_t}{\gamma}\Big]^{\gamma}v_t\log(u_t) \\ & -\beta_1\Big[1-\frac{1-u_t}{\gamma}\Big]^{\gamma}(1-v_t)\log(1-u_t) \end{aligned} \tag{16}$$

, where $\beta_0 = P(v_t = 0)$, which is the proportion of normal events in the dataset and $P(v_t = 1)$ is the proportion of right extreme events in the dataset. γ is the hyper-parameter, which is the *extreme value index* in the approximation. We call the proposed classification loss function as Extreme Value Loss (EVL). Similarly we have the binary classification loss function for detecting whether there will be a left extreme event in the future. Combining two loss functions together we can extend EVL to the situation of $v_t = \{-1, 0, 1\}$.

As we have discussed in Sec. 3, without proper weight on non-parameteric estimator DNN will suffer *overfitting* problem. The key point of EVL is to find the proper weights by adding approximation, e.g., $\beta_0[1-u_t/\gamma]^{\gamma}$, on tail distribution of the observations through extreme value theory. Intuitively, for detecting right extreme event, term β_0 will increase the penalty when the model recognizes the event as normal event. Meanwhile, the term $[1-u_t/\gamma]^{\gamma}$ also increase the penalty when the model recognize the extreme event with little confidence. In the following part we will demonstrate how to incorporate EVL with the proposed memory network based framework.

4.3 Optimization

In this part, we present the optimization algorithm for our framework. First, in order to incorporate EVL with the proposed memory network, a direct thought is to combine the predicted outputs o_t with the prediction of the occurrence of extreme events,

$$L_1 = \sum_{t=1}^{T}\|o_t - y_t\|^2 + \lambda_1 EVL\big(u_t, v_t\big) \tag{17}$$

Furthermore, in order to enhance the performance of GRU units, we propose to add the penalty term for each window j, which aims at predicting extreme indicator q_j of each window j:

$$L_2 = \sum_{t=1}^{T}\sum_{j=1}^{M} EVL\big(p_j, q_j\big) \tag{18}$$

, where $p_j \in [-1, 1]$ is calculated through s_j, which is the embedded representation of window j, by a full connection layer. Finally we list the whole parameters to learn as follows.

- Parameters θ in GRU units: $W_z, W_r, W_h, b_z, b_r, b_h \in \mathbb{R}^H, U_z, U_r, U_h \in \mathbb{R}^{H\times H}$
- Parameters ϕ in output gate: $W_o \in \mathbb{R}^H$, $b \in \mathbb{R}$.
- Parameters ψ in predicting the happening of extreme events in history data: $W_p, b_p \in \mathbb{R}^H$.
- Parameters b in attention model: $b \in \mathbb{R}^3_+$.

We use Adam [26] to learn the parameters. For the whole learning process, please refer to Algorithm 1.

5 EMPIRICAL RESULTS

In this section, we present empirical results for our proposed framework. In particular, the main research questions are:

- **RQ1**: Is our proposed framework effective in time series prediction?
- **RQ2**: Is our proposed loss function worked in detecting extreme events?
- **RQ3**: What is the influence of hyper-parameters in the framework?

5.1 Experimental Settings

We conduct experiments on three different kinds of datasets:

- **Stock Dataset** We collect the stock price of 564 corporations in Nasdaq Stock Market, with one sample per week. Our collected data covers the time duration from September 30,2003 to December 29,2017.
- **Climate Datasets** The climate datasets are composed of "Green Gas Observing Network Dataset" and "Atmospheric Co2 Dataset" which are built by Keeling and Whorf and Lucas et al. respectively [25, 34]. The green house dataset contains green house gas concentrations at 2921 grid cells in California covering an area of 12 km by 12 km which are spaced 6 hours apart (4 samples per day) over the period May 10-July 31,2010. The Co2 dataset contains atmospheric Co2 concentrations collected from Mount Monalo in Hawaii week by week over the period March 1958 - December 2001.
- **Pseudo Periodic Synthetic Dataset** The original dataset contains one million data points which has been split into 10 sections. All values are in range $[-0.5, 0.5]$.

For these first two datasets, we set the time length as 500 for training and 200 for testing, while for the last dataset, we randomly extract 150 time series per section with 400 data points, setting the time length as 300 for training and 100 for testing. It is worth to notice that the regularity of our selected there datasets increase in order, i.e. the pattern of pseudo datasets is totally fixed while stock data has intensive noises. We further preprocessed the data by replacing the raw data with the difference between time t and $t-1$. By a subsequent normalization of data value to a fixed range, the final datasets were therefore constructed.

For each experiment, we conducted a 10-fold cross validation and reported the averaged results. For the choice of hyper-parameter γ in EVL, on one hand, previous works have pointed out that maximizing likelihood methods only work well when $\gamma > 2$ on estimating extreme value distribution [12]. On the other hand, we have noticed that it is improper to set γ a large value due to the fact that $p \in [0, 1]$. Therefore in most cases we chose γ optimally from $(2, 3]$. On the influence of different γ, an analysis will be presented in experiments. We set the memory size M as 80 and the window size Δ as 50. We will also analyze the different choices of M and Δ later. The learning rate was commonly set as 0.0005.

5.2 Effectiveness of Time Series Prediction

We first validate our complete framework for time series data prediction. We chose Rooted Mean Square Error (RMSE) as the metric, where a smaller RMSE means a better performance. We compared

Table 2: F1 value of detecting extreme events with different classification loss function.

Model	Climate			Stock			Pseudo		
	Micro	Macro	Weighted	Micro	Macro	Weighted	Micro	Macro	Weighted
LSTM+CE	0.435	0.833	0.786	0.247	0.617	0.527	0.830	0.900	0.899
GRU+CE	0.471	0.717	0.733	0.250	0.617	0.547	0.854	0.917	0.917
GRU+EVL ($\gamma = 0.5$)	0.644	0.883	0.859	0.281	0.583	0.523	0.833	0.900	0.902
GRU+EVL ($\gamma = 1.0$)	**0.690**	**0.900**	**0.881**	0.267	**0.667**	0.547	**0.874**	**0.933**	**0.933**
GRU+EVL ($\gamma = 2.0$)	0.646	0.867	0.851	**0.324**	0.617	**0.555**	0.869	0.917	0.920
GRU+EVL ($\gamma = 3.0$)	0.508	0.867	0.825	0.295	0.617	0.548	0.810	0.900	0.897
GRU+EVL ($\gamma = 4.0$)	0.617	0.817	0.813	0.295	0.617	0.543	0.825	0.883	0.886

Table 3: RMSE results of time series prediction.

	Climate	Stock	Pseudo
LSTM	0.188	0.249	5.2×10^{-3}
Time-LSTM	0.193	0.256	4.7×10^{-3}
GRU	0.174	0.223	5.3×10^{-3}
Mem	0.181	0.197	3.6×10^{-3}
Mem+EVL	**0.125**	**0.168**	$\mathbf{2.5 \times 10^{-3}}$

our model with several state-of-the-art baselines: GRU, LSTM and Time-LSTM [50], where Time-LSTM considers differences between x_t. We also compare our model with memory network without EVL, by replacing EVL with cross entropy (CE). The results are in Table 3. Surprisingly, GRU outperformed other baselines although it has the simplest structure in real-world data. We infer the reason is that, there are much noises in real-world data, which could easily cause overfitting problem on one-dimensional data as we have described before. Furthermore, as we can see, the RMSE of our model were uniformly lower than GRU. Noticeably, on the synthetic dataset we successfully made a near 50% improvement in RMSE.

We also visualize the output from each module as an case study (Fig. 4). As we can see from the results, the empirically success of our model can be mainly attributed to two parts: the predicting value $\tilde{o}_t$ and the extreme events label u_t. The output from $\tilde{o}_t$ approximated the trend of the data correctly but, usually, the value it predicted commonly small. As a complement, u_t came to rescue by amplifying the prediction value if it predicts the occurrence of extreme event at the current step with a high probability. Illustratively, it is worth to notice the visualization around time step 600 in Fig. 4. Although $\tilde{o}_t$ predicted the trend as up, however, it only gave a small positive value. As a complement, the memory module detected there would be an right extreme event at this time step, therefore it yielded a near-1 output and imposed an amplification on $\tilde{o}_t$ to form the final output, while GRU could hardly do such a complicated decision.

5.3 Effectiveness of EVL

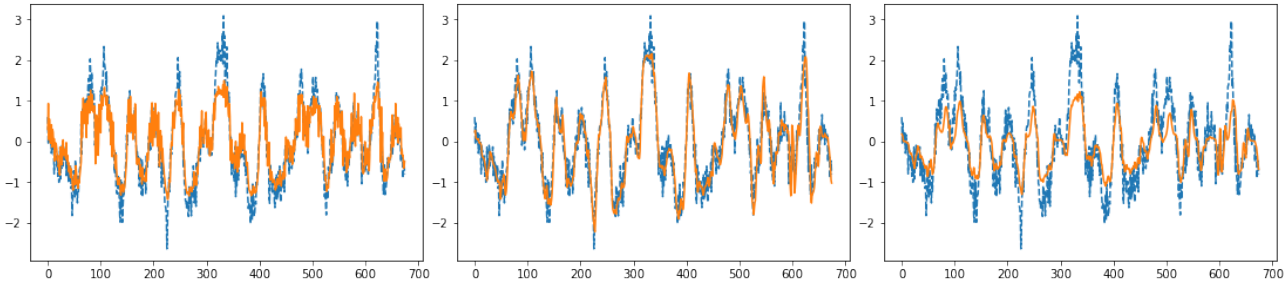
As we can see from Table 3, EVL plays an important role during the prediction. We further validated the effectiveness of EVL on predictions of future occurrences of extreme events. We used F1 score to measure the effectiveness of predictions. Specifically, we adopted the Macro, Micro and Weighted F1 score for an overall evaluation.The results are presented in Fig. 2. We compared our proposed EVL with GRU classifier and LSTM classifier. Also we studied the influence of different hyper-parameter γ for EVL. First as we can see from Fig. 2, our proposed loss function outperformed all the baselines on each dataset. Especially on the climate dataset, EVL outperformed the best baseline by 47% in Micro F1 score. Interestingly, we have observed that γ indeed influenced the final classification results a lot. For example, when $\gamma = 4.0$, the performance of EVL was worse than the baseline on the synthetic dataset. As we have discussed before, γ intuitively described the characteristics of tail distribution of the data, therefore, an improper γ could mislead the model to an incorrect modeling of the tail distribution.

Furthermore, the experimental results also support our suggestion of γ discussed in Sec. 5.1, as the optimal γ was always found around 2.0 on each data (Table. 2). On the other hand, γ was also observed to capture the characteristics of heavy-tailed data distribution well. For instance, since there were more noises in the stock dataset, i.e. the extreme events occurs more frequently. In this case, the observed optimal γ, which took a larger value than other datasets, corresponds well to this prior knowledge. Therefore, in the following experiments, we commonly took the hyper-parameter γ as 2.0.

5.4 Influence of Hyper-parameters in Memory Network

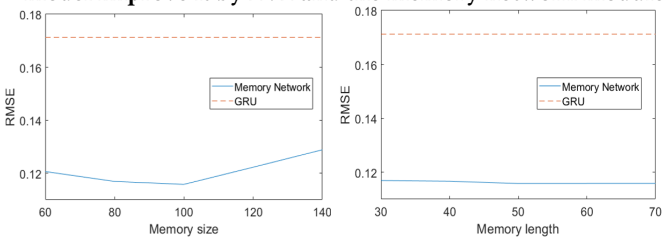
Finally we investigate the influence of hyper-parameters in memory module, where the results are in 5. As we can see from Fig. 5(a), the optimal choice of memory size varied on different datasets. For instance, $M = 100$ for the climate dataset and $M = 80$ for the stock dataset, where $M = 60$ for the synthetic dataset, with an average optimal choice for memory size around 80. We infer the reason is that, with small memory size, the model is not able to memorize extreme events sufficiently in the history. On the contrary, if we keep enlarging the size of the memory, the model would otherwise pay inessential attentions to historical events rather than making the right decision based on the current situation. One special case is the synthetic dataset, where the smaller memory size performs better. We speculate the reason as, considering the inherent regularity of the data set is extremely high, a learning model with sufficient learning capacity would easily captured the regularity without referring to historical records.

After we investigated the influence of M for different datasets, we also would like to understand the influence of different window size on our memory network module. As we have observed 5(b), the influence of the window size was much smaller than the memory size. However, the window size Δ still have non-negligible effect on the final prediction. For instance, the best Δ was 60 for the stock dataset. Such a phenomenon mainly comes from, when the length is too small, the information or the pattern of extreme events is not modeled well, which therefore would influence the effectiveness



(a) Output from GRU (b) Output from our model (c) Output from $\tilde{o}_t$

Figure 4: Case study for Our Framework via Visualizations. Although outputs from $\tilde{o}_t$ still suffer extreme events problem, our model improve it by EVL and the memory network module.



(a) Influence of memory size M (b) Influence of window size Δ

Figure 5: Influence of different hyper-parameters in memory network module. We only demonstrate the results in climate datasets due to the limit of space.

of the prediction, while on the other hand, when the length is set too large, the GRU module in the memory network would lack the ability to memorize all the characteristics inside the window.

6 RELATED WORK

Time series prediction is a classical task in machine learning. Traditional methods such as autoregressive moving average model [46] uses linear regression on latest inputs to predict the next value, while Nonlinear autoregressive exogenous (NARX) adopts non-linear mapping to improve the prediction [31]. However, their learning capacity are limited because traditional methods use shallow architecture [6]. Recently, deep neural network (DNN) achieve great success in many fields , e.g., computer vision [28] and neural language processing [4], mainly because of its deep nonlinear architecture which allows extraction of better features from the inputs [39]. A number of DNN based models has also been applied on time series prediction. Dasgupta and Osogami proposed to use Boltzmann Machine [11], while Yang et al. proposed to use Convolutional Network [49] to improve the performance. Among these deep learning models, recurrent neural network (RNN) shows superior advantages in modeling time-series data. For instance, Diaconescu proposes to combine RNN with NARX model, which largely reduces the errors in prediction [13, 48]. Numerous improvements have been made on RNN, such as Long-short Term Memory [22] and Gated Recurrent Unit [9], with various empirical results reporting the effectiveness of these deep recurrent models [30, 36, 50].

Data imbalance is always an important issue in machine learning and data mining. For instance, considering certain classification task, if there exists one label which only has few samples in training set, then the model will lack the ability to detect them [15]. Actually, the cutting-edge deep learning models are also observed to suffer from the data imbalance problem in most cases. As pointed out by Lin et al., data imbalance problem mainly causes two haunting phenomenons in DNN: (1) model lacks the ability to model rarely occurred samples; (2) the generalization performance of model is degenerated [32]. As we have discussed in the introductory part, in the context of time-series data, these imbalanced samples are often called extreme events. Previous studies mainly focus on modeling the distributions of these extreme values [1]. Intuitively, due to the rare and irregular essence of extreme events, it is always hard to forecast these pumping points [19].

7 CONCLUSION

In this paper, we focus on improving the performance of deep learning methods on time series prediction, specifically with a more find-grained modeling on the part of extreme events. As discussed in Sec.3, without prior knowledge on tailed observations, DNN is innately weak in capturing characteristics of the occurrences of extreme events. Therefore, as a novel technique delicately designed for extreme events, we have proposed a framework for forecasting time series data with extreme events. Specifically we consider two factors for imposing tailed priors: **(1)** memorizing extreme events in historical data **(2)** modeling the tail distribution of observations. For the first factor we utilize the recently proposed Memory Network technique by storing historical patterns of extreme events for future reference. For the second factor we propose a novel classification loss function called Extreme Value Loss (EVL) for detecting extreme events in the future, which contains the approximated tail distribution of observations. Finally we combine them together and form our end-to-end solution for predictions of time series with extreme events. With intensive experiments, we have validated both the effectiveness of EVL on extreme event detection and the superior performance of our proposed framework on time series prediction, compared with the state-of-the-arts. For future works, we plan to work on an efficient extension of our framework to multi-dimensional time series prediction in consideration of its significance in practice and its challenges in theory. Furthermore, we suggest it would also be interesting to exploit the possibility of applying our proposed EVL to derive solutions for other tasks featured with data imbalance such as Point-of-Interest (POI) recommendation [14].

ACKNOWLEDGMENTS

The work is supported in part by the National Program on Key Basic Research (NO. 2015CB358800), the National Natural Science Foundation of China (U1636204, 61602121, U1736208, 61602123, U1836213, U1836210) and Thousand Youth Talents Program 2018. Min Yang is also a member of Shanghai Institute of Intelligent Electronics & Systems, Shanghai Institute for Advanced Communication and Data Science, and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China.

REFERENCES

[1] Sergio Albeverio, Volker Jentsch, and Holger Kantz. 2006. *Extreme events in nature and society.* Springer Science & Business Media.

[2] Eduardo Altmann and Kantz H. 2005. Recurrence time analysis, long-term correlations, and extreme events. *Physical Review E Statistical Nonlinear and Soft Matter Physics* (2005).

[3] Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017).

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[5] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. 2005. Clustering with Bregman divergences. *Journal of machine learning research* 6, Oct (2005), 1705–1749.

[6] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*. 153–160.

[7] Tine Buch-Larsen, Jens Perch Nielsen, Montserrat Guillén, and Catalina Bolancé. 2005. Kernel density estimation for heavy-tailed distributions using the Champernowne transformation. *Statistics* 39, 6 (2005), 503–516.

[8] Armin Bunde, Jan F Eichner, Shlomo Havlin, and Jan W Kantelhardt. 2003. The effect of long-term correlations on the return periods of rare events. *Physica A: Statistical Mechanics and its Applications* 330, 1-2 (2003), 1–7.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[11] Sakyasingha Dasgupta and Takayuki Osogami. 2017. Nonlinear Dynamic Boltzmann Machines for Time-Series Prediction.. In *AAAI*. 1833–1839.

[12] Laurens De Haan and Ana Ferreira. 2007. *Extreme value theory: an introduction*. Springer Science & Business Media.

[13] Eugen Diaconescu. 2008. The use of NARX neural networks to predict chaotic time series. *Wseas Transactions on computer research* 3, 3 (2008), 182–191.

[14] Daizong Ding, Mi Zhang, Xudong Pan, Duocai Wu, and Pearl Pu. 2018. Geographical Feature Extraction for Entities in Location-based Social Networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 833–842.

[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28, 4 (2006), 594–611.

[16] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal Relational Ranking for Stock Prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 27.

[17] R. A. Fisher and L. H. C. Tippett. 1928. Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society* 24, 2 (1928), 180–190.

[18] JANOS GALAMBOS. 1977. The asymptotic theory of extreme order statistics. In *The Theory and Applications of Reliability with Emphasis on Bayesian and Nonparametric Methods*. Elsevier, 151–164.

[19] M Ghil, P Yiou, Stéphane Hallegatte, BD Malamud, P Naveau, A Soloviev, P Friederichs, V Keilis-Borok, D Kondrashov, V Kossobokov, et al. 2011. Extreme events: dynamics, statistics and prediction. *Nonlinear Processes in Geophysics* 18, 3 (2011), 295–350.

[20] Gnedenko. 1943. Sur la distribution limite du terme maximum dâĂŹune sÃľrie alÃľatoire. *Annals of Mathematics* 44, 3 (1943), 423–453.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[22] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[23] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[24] Holger Kantz, Eduardo G Altmann, Sarah Hallerberg, Detlef Holstein, and Anja Riegert. 2006. Dynamical interpretation of extreme events: predictability and predictions. In *Extreme events in nature and society*. Springer, 69–93.

[25] Charles David Keeling and Timothy P Whorf. 2004. Atmospheric CO2 concentrations derived from flask air samples at sites in the SIO network. *Trends: a compendium of data on Global Change* (2004).

[26] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[27] Samuel Kotz and Saralees Nadarajah. 2000. *Extreme value distributions. Theory and applications*. Prentice Hall,. 207–243 pages.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[29] Holden Lee, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. 2017. On the ability of neural nets to express distributions. *arXiv preprint arXiv:1702.07028* (2017).

[30] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid neural networks for learning the trend in time series. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2273–2279.

[31] Tsungnan Lin, Bill G Horne, Peter Tino, and C Lee Giles. 1996. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks* 7, 6 (1996), 1329–1338.

[32] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[33] Edward N Lorenz. 1963. Deterministic nonperiodic flow. *Journal of the atmospheric sciences* 20, 2 (1963), 130–141.

[34] DD Lucas, C Yver Kwok, P Cameron-Smith, H Graven, D Bergmann, TP Guilderson, R Weiss, and R Keeling. 2015. Designing optimal greenhouse gas observing networks that consider performance and cost. *Geoscientific Instrumentation, Methods and Data Systems* 4, 1 (2015), 121–137.

[35] T Okubo and N Narita. 1980. On the distribution of extreme winds expected in Japan. *National Bureau of Standards Special Publication* 560 (1980), 1.

[36] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).

[37] Tomasz Rolski, Hanspeter Schmidli, Volker Schmidt, and Jozef L Teugels. 2009. *Stochastic processes for insurance and finance*. Vol. 505. John Wiley & Sons.

[38] Murray Rosenblatt. 1956. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics* (1956), 832–837.

[39] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. 3856–3866.

[40] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 650–658.

[41] Jeroen Van den Berg, Bertrand Candelon, and Jean-Pierre Urbain. 2008. A cautious note on the use of panel models to predict financial crises. *Economics Letters* 101, 1 (2008), 80–83.

[42] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. 3630–3638.

[43] Ladislaus von Bortkiewicz. 1921. *Variationsbreite und mittlerer Fehler*. Berliner Mathematische Gesellschaft.

[44] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[45] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *CoRR* abs/1410.3916 (2014).

[46] Peter Whitle. 1951. *Hypothesis testing in time series analysis*. Vol. 4. Almqvist & Wiksells.

[47] Rym Worms. 1998. Propriété asymptotique des excès additifs et valeurs extrêmes: le cas de la loi de Gumbel. *Comptes Rendus de l'Academie des Sciences Series I Mathematics* 5, 327 (1998), 509–514.

[48] Linjun Yan, Ahmed Elgamal, and Garrison W Cottrell. 2011. Substructure vibration NARX neural network approach for statistical damage inference. *Journal of Engineering Mechanics* 139, 6 (2011), 737–747.

[49] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.. In *Ijcai*, Vol. 15. 3995–4001.

[50] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 3602–3608.