An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks

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

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

Abstract

Forecasting time series with *extreme* events has been a challenging and prevalent research topic, especially when the time series data are affected by complicated uncertain factors, such as is the case in hydrologic prediction. Diverse traditional and deep learning models have been applied to discover the nonlinear relationships and recognize the complex patterns in these types of data. However, existing methods usually ignore the negative influence of imbalanced data, or severe events, on model training. Moreover, methods are usually evaluated on a small number of generally well-behaved time series, which does not show their ability to generalize. To tackle these issues, we propose a novel probability-enhanced neural network model, called NEC+, which concurrently learns *extreme* and *normal* prediction functions and a way to choose among them via selective back propagation. We evaluate the proposed model on the difficult 3-day ahead hourly water level prediction task applied to 9 reservoirs in California. Experimental results demonstrate that the proposed model significantly outperforms state-of-the-art baselines and exhibits superior generalization ability on data with diverse distributions.

Introduction

Time series forecasting is an important technique for many domains in which most types of data are stored as time sequences, including traffic (Hua, Kapoor, and Anastasiu 2018), weather forecasting (Hewage et al. 2021), biology (Bose et al. 2022), stock price forecasting (Mohan et al. 2019), and water resource management (Zhang et al. 2021). These data usually contain seasonality, long term trends, and non-stationary characteristics which usually are taken into account by traditional models during prediction. However, in hydrologic prediction, the water level of dams and reservoirs are also affected by complicated uncertain factors like weather, geography, and human activities, which makes the task of precisely predicting them challenging. Most reservoirs are large hydraulic constructions that serve multiple purposes, including power generation, flood control, irrigation, and navigation, making them critical components in the safety and quality of life of the general population. Therefore, a large number of studies and architectures have explored the problem of reservoir water level prediction.

For a long time, water level prediction was mainly based on traditional machine learning and statistics-based models. However, methods such as Autoregressive Integrated Moving Average (ARIMA) (Box and Jenkins 1976) seem to adjust poorly to extreme changes in the water level values and cannot easily find the nonlinear relationships among the data. Recently, deep neural networks (DNNs) have shown their great advantages in various areas (Yang et al. 2019; Anastasiu et al. 2020; Pei et al. 2021). Both conventional Neural Network (NN) and Recurrent Neural Network (RNN) models have been used to overcome the disadvantages of traditional methods for time series forecasting (Qi et al. 2019), since they can map time series data into latent representations by capturing the non-linear relationships of data in sequences. In particular, Long Short-Term Memory (LSTM) models generally outperform other models in long-term predictions. However, imbalanced data or severe events might hurt deep learning models when it comes to long-term predictions. In the context of reservoir water level forecasting, most of the works mentioned above falter when predicting *extreme* events. They also usually focus on predicting only one or two sensors, putting in question the generalizability of these models. To solve these two challenges, we provide an extreme-adaptive solution for reservoir water level prediction which we evaluate extensively on data from 9 different reservoirs across more than 30 years.

The fundamental contribution of this research is the proposal of NEC+[1], a probability-enhanced neural network framework. We use clustering in NEC+ to dynamically produce distribution indicators, which improves the model's robustness to the occurrence of severe events. To improve training performance, we present a selected backpropagation approach and a two-level sampling algorithm to accommodate imbalanced extreme data, along with a customizable weighted loss function for implementing a binary classifier, which is a crucial component of NEC+.

Related Work

Time series prediction has been studied extensively. Traditionally, there were several techniques used to effectively forecast future values in the time series, including univariate Autoregressive (AR), Moving Average (MA),

---

[1]https://github.com/davidanastasiu/NECPlus

Simple Exponential Smoothing (SES), Extreme Learning Machine (ELM) (Huang et al. 2012), and more notably ARIMA (Box and Jenkins 1976) and its many variations. In particular, the ARIMA model has demonstrated it can outperform even deep learning models in predicting future stock (Ariyo, Adewumi, and Ayo 2014) and dam reservoir inflow levels (Valipour, Banihabib, and Behbahani 2013). Prophet (Taylor and Letham 2018) is an additive model that fits nonlinear trends with seasonal and holiday impacts at the annual, weekly, and daily levels. A number of other classical machine learning models have been used for the task of water level prediction. Due to lack of space, we details them in the additional related work section in the appendix (which can be found in (Li, Xu, and Anastasiu 2022)).

With the recent success of deep neural network (DNN) models, hybrid models incorporating different prediction methodologies were also used for water level prediction. Zhang et al. (Zhang et al. 2021) designed CNNLSTM, a deep learning hybrid model based on the Convolutional Neural Network (CNN) and LSTM models, to predict downstream water levels. Du and Liang (Du and Liang 2021) created an ensemble LSTM and Prophet model which was shown to outperform any of the single models used in the ensemble. Le et al. (Le et al. 2021) added an attention mechanism (Xu et al. 2015; Chorowski et al. 2015) to an encoder-decoder architecture to solve the hydro prediction problem. Ibañez et al. (Ibañez et al. 2022) examined two versions of the LSTM based DNN model exactly for the reservoir water level forecasting problem, a univariate encoder-decoder model (DNN-U) and a multivariate version (DNN-M). Both models used trigonometric time series encoding.

Statistical methods also provide promising solutions when they are combined with DNN models, especially in the field of sales forecasting. DeepAR (Salinas et al. 2020) approximates the conditional distribution using a neural network. Deep State Space Models (DeepState) (Rangapuram et al. 2018) is a probabilistic forecasting model that fuses state space models and deep neural networks. By choosing the appropriate probability distribution, the bias in the objective function becomes further reduced and the prediction accuracy can be improved. Tyralis and Papacharalampous showed that the architecture can be simply calibrated using the quantile (Tyralis and Papacharalampous 2021) or the expectile (Waltrup et al. 2015) loss functions for delivering quantile or expectile hydrologic predictions and forecasts. N-BEATS (Oreshkin et al. 2019), builds a pure deep learning solution which outperforms well-established statistical approaches in more general time series problems. The N-BEATS interpretable architecture is composed of 2 stacks, namely a trend model and a seasonality model.

While many recent water level prediction methods showed they can outperform traditional or simple DNN models, none of them consider the imbalance of *extreme* vs. *normal* events in the time series and hence ignore the negative influence of *extreme* values on model training. Generally, these *extreme* values could be deemed as outliers and be recognized and even removed during data preprocessing. However, in our problem, accurate prediction of *extreme* events is generally even more important than the prediction

of normal ones. However, we focus on achieving the *best overall prediction performance*, without sacrificing either the quality of *normal* or of *extreme* predictions.

Preliminaries

Problem Statement

We take on a challenging univariate time series forecasting problem, considering that the data contain a majority of normal values that significantly contribute to the overall prediction performance, along with a minority of extreme values that must be precisely forecasted to avoid disastrous events.

The problem can be described as,

$$[x_1, x_2, \ldots, x_T] \in \mathbb{R}^T \rightarrow [x_{T+1}, \ldots, x_{T+H}], \in \mathbb{R}^H$$

which means predicting the vector of length-H horizon future values, given a length-T observed series history, where x_1 to x_T are inputs and x_{T+1} to x_{T+H} are the outputs. Root mean square error (RMSE) and mean absolute percentage error (MAPE), as standard scale-free metrics, are used to evaluate forecasting performance.

For our experiments, we obtained approximately 31 years of hourly reservoir water level sensor data, along with rain gauge data from a number of sensors in the same area. The Santa Clara reservoirs were built for water conservation in the 1930s and 1950s in order to catch storm runoff that would otherwise drain into the San Francisco Bay. The reservoirs also provide flood protection by controlling runoff early in the rainy season, recreational opportunities, and they aid the ecology by storing water to keep rivers flowing.

Our models predict 72 (hours) future reservoir water level values, i.e., 3 days ahead. Table 1 in the appendix shows the location and type of sensors used in this study. In the remainder of the paper, we will refer to the sensors and theirs associated time series by their given sensor ID in the table.

Extreme Events

Extreme Value Theory (EVT) tries to explain the stochastic behavior of extreme events found in the tails of probability distributions, which often follow a very different distribution than "normal" values. Towards that end, the Generalized Extreme Value (GEV) distribution is a continuous probability distribution that generalizes extreme values that follow the Gumbel (Type I), Fréchet (Type II), or Weibull (Type III) distributions. Its cumulative distribution function (CDF) is described as

$$F(x; \mu, \sigma, \xi) = \exp\left\{ -\left[1 + \xi\left(\frac{x - \mu}{\sigma}\right) \right]^{-1/\xi} \right\}, \quad (1)$$

where $\mu \in \mathbb{R}$, $\sigma > 0$, and ξ are the location, scale, and shape parameters, respectively, conditioned on $1 + \xi(x - \mu)/\sigma > 0$.

Figure 1 shows water levels for five of our 9 sensors across a period of 20 years. In order to understand whether extreme events were present in these data, we fit GEV and Gaussian probability density functions (pdf) to the water level values and found that the GEV distribution provides a better fit. In particular, the RMSE of the Gaussian distribution fit is 26.9%, 46.0%, and 37.2% higher than that of the GEV
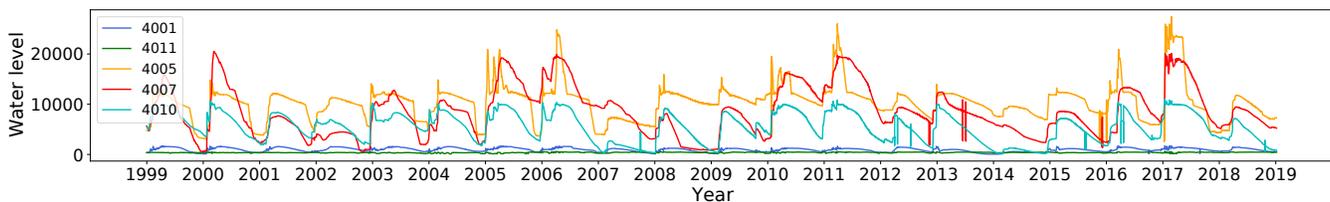
Figure 1: Water level values for 5 reservoirs across 20 years.

distribution fit for the 4001, 4003, and 4009 reservoirs, respectively, which we also show graphically in the appendix. However, our data has distinct *seasonality* (rain in winter will increase water levels) and *trends* (reservoirs slowly deplete over the year). A time series with trends, or with seasonality, is not stationary and will generally lead to inferior predictions. Therefore, we follow a standard time series analysis preprocessing approach and obtain a stationary time series by applying first-order differencing and then standardizing the resulting time series values, $x'_t = x_t - x_{t-1}$, and $x' = \frac{x' - \mu}{\sigma}$, where μ and σ here are the mean (location parameter) and standard deviation (scale parameter) of the Gaussian distribution of the time series x'. After obtaining predictions for a time series, we use the same location and scale parameters to inverse the standardization, and the last ground truth value in the time series to inverse the first-order differencing, obtaining values in the same range as the original time series. When the mean of the distribution is 0 ($\mu = 0$) and its standard deviation is 1 ($\sigma = 1$), as is the case in our standardized time series, 68% of the values lie within 1 standard deviation, 95% within 2 standard deviations, and 99.7% within 3 standard deviations from the mean. Yet our time series show values that are up to 100 standard deviations away from the mean in both directions. In our work, we define *normal* values as those within $\epsilon \times \sigma$ of the mean of the preprocessed time series, in both directions, where ϵ is a meta-parameter we tune for each time series, i.e., $x'_n \in [-\epsilon, \epsilon]$, since $\sigma = 1$. The remaining values in the time series are then labeled as *extreme* values.

Methods

NEC

We designed our NEC framework to account for the distribution shift between *normal* and *extreme* values in the time series. NEC is composed of three separate models, which can be trained in parallel. The Normal (N) model is trained to best fit *normal* values in the time series, the Extreme (E) model is trained to best fit *extreme* time series values, and a third Classifier (C) model is trained to detect when a certain value may be categorized as *normal* or *extreme*. The framework is flexible and may use any prediction models for the 3 components, yet in this work, given the evidence presented in related work section, we focus on deep learning models that use a fixed set of h consecutive past values as input to predict the next f values in the time series. At prediction time, the C model is used to decide, for each of the following f time points, whether the value will be *normal* or *ex-*

treme, and the appropriate regression model is then applied to obtain the prediction for those points.

The middle section of Figure 2 shows the configuration for our chosen N, E, and C models. The N and E models each have 4–6 LSTM layers followed by 3 fully connected (FC) layers that consecutively reduce the width of the layer down to f, which is 72 in our case (3 days). The number of inputs was set to 15 days, i.e., $h = 15 \times 24 = 360$. Since there are much fewer extreme values in the data than normal ones, we set the LSTM layer width to only 512 in the E model, while we set it to 1024 in the N model. Finally, the C model uses the same size LSTM layers as the N model, followed by a 72 node FC layer with a Sigmoid activation function.

GMM Indicator

A Gaussian mixture model (GMM) (Day 1969) can be described by the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \, g(x|\mu_i, \mathbf{\Sigma_i}),$$

where x is a D-dimensional continuous-valued vector, $w_i \; \forall i = 1, \ldots, M$ are the mixture weights, and $g(x|\mu_i, \Sigma_i)$, are the component Gaussian densities. Each component density is a D-variate Gaussian function, and the overall GMM model is a weighted sum of M component Gaussian densities,

$$g(x|\mu_\mathbf{i}, \mathbf{\Sigma_i}) = \frac{1}{2\pi^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x - \mu_i)^T \, \Sigma_i^{-1}(x - \mu_i)\right\},$$

where μ_i is the mean vector and Σ_i is the covariance matrix of the ith component. The mixture weights are constrained such that $\sum_{i=1}^{M} w_i = 1$. The GMM's capacity to produce smooth approximations to arbitrarily shaped densities is one of its most impressive features (Day 1969).

In our work, we use Expectation-Maximization to fit a GMM model using the time series training data. Then, each model component can generate a probability for each point in the time series. Finally, for each value in the time series, we compute an *indicator* feature as the weighted sum of all component probabilities, given the weights learned when fitting the GMM model. In our framework, the number of components M is a hyper-parameter which we tune for each time series. As an illustration, Figure 3 shows the indicator values for GMM with $M = 4$ for Sensor 4009 and with $M = 3$ for the other 9 sensors. The x-axis in the figures represents the preprocessed time series input, which we limited to the
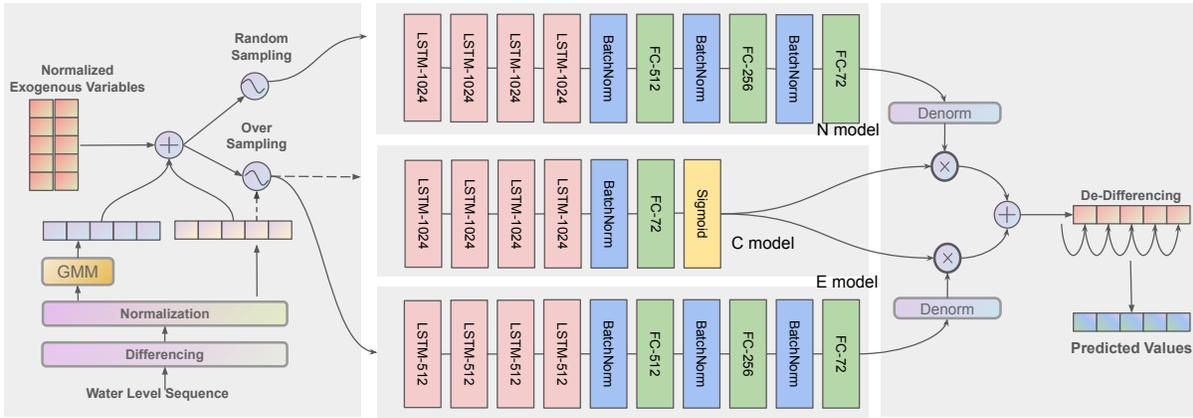
Figure 2: The NEC+ framework; $\bigoplus$ denotes the element-wise addition and $\bigotimes$ denotes the element-wise product.
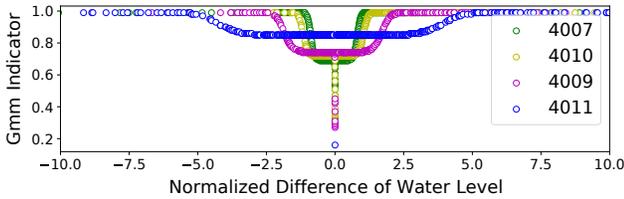
Figure 3: GMM indicator distribution.

$[-10, 10]$ range for visibility, while the y-axis represents the indicator values. It is easy to see that the learned *normal* and *extreme* indicator bounds vary depending on the sensors. We hypothesize that providing this indicator as an additional input feature for our models will help differentiate between *normal* and *extreme* values and minimize prediction errors. Therefore, we extend our N, E, and C models by providing both the water level and its associated GMM indicator as input for each of the h input values.

Exogenous Variables

For some time series, we may provide additional exogenous inputs which may help improve the overall prediction of future values. For example, rain fall in the region around the reservoir is not affected by the reservoir water level but, when it is raining, it can be a strong indicator that the reservoir level may increase soon, as water drains into streams and rivers that may flow into the reservoir. For a given region, a *watershed* is a land area that channels rainfall and snowmelt to creeks, streams, and rivers, and eventually to outflow points such as reservoirs, bays, and the ocean. In our work, as shown in Table 1 in the appendix, we define several watersheds and use several rain gauge sensors in those watersheds as exogenous variables to aid in the predictions associated with several of the reservoirs, which were chosen in consultation with domain experts. Namely, for reservoir 4005 we used rain sensor 6017, for 4010 we used 6135, and for 4007 we used both 6044 and 6069.

NEC+

Our NEC+ framework is described in Figure 2. Unlike the base NEC framework, which relies only on historical time series values for future predictions, NEC+ adds GMM indicator and watershed exogenous variables (when available) to create multivariate regression N and E models. Given k watershed variables, the NEC+ models will be $k + 2$-variate models, after adding the original input and the GMM indicator. In addition, to account for the differences between the distributions of the *normal* and *extreme* values during training, we define custom sampling policies, regression back-propagation, and classification loss function.

Sampling Policies

Our models require h values from the time series to predict the following f values, and $h, f \ll |x|$, the length of the time series. Moreover, while the number of *extreme* values differs based on the choice of ϵ, it is still quite small in comparison to the number of normal values. In our experiments, $h = 360$, $f = 72$, $|x| \sim 276K$, and *extreme* values ranged from 0.08% to 4.08% of the time series values across our 9 sensors. Therefore, sampling plays a crucial role during training. However, oversampling cannot be used to mitigate this problem. In an experiment we detail in the appendix (due to lack of space), we found that, while oversampling extreme events improves predictions in that area, it leads to worse overall predictions for the rest of the time series.

When training our NEC+ model, we apply a two-stage sampling policy. First, given the high cardinality of our time series, we randomly sample subsections of length $h + f$ from the series as samples to use in training our models, while avoiding sections included in the test and validation sets. Specifically, the validation and test sets each include 24 randomly chosen f-length sections from the years 2014 and 2016 for the validation and 2017 and 2018 for the test set, respectively, and the training set includes all other values in the time series. Second, we perform stratified sampling of regions with and without *extreme* values, allowing the E and C models to oversample up to OS% samples with at least 1 *extreme* value in the prediction zone.

Selected Backpropagation in the N and E Models

In addition to a custom sampling policy, one important approach we use to ensure proper training of the N and E models is *selected backpropagation*, which we describe visually in Figure 4. Each prediction sample in our data contains f values, only a few of which may be extreme. The rarity of extreme events would cause the E model to be unduly influenced by the loss on normal values, and vice-versa. As a result, our backpropagation ignores predictions on normal values in the E model and on extreme values in the N model, forcing the model to only focus on the values important for the given model. Specifically, when training the N model, only *normal* values add to the loss, and when training the E model, only *extreme* values add to the loss. This is equivalent to perfect predictions (predicting the ground truth) for *normal* values when training the E model, and perfect predictions for *extreme* values when training the N model. In this way, only the positions and values of appropriate *normal* or *extreme* data will affect the hidden parameters in the network during backpropagation when training the N and E models.
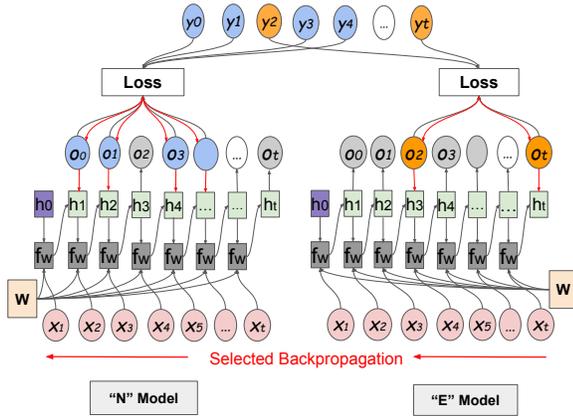
Figure 4: Computational graph for the N and E models. Blue and orange inputs represent *normal* and *extreme* values, respectively.

Parameterized Loss Function in the C Model

The Binary Cross Entropy (BCE) loss is usually the most appropriate loss function for binary classification tasks. Based on BCE, we propose a tunable loss function to accommodate the serious imbalance problem in the prediction of time series with extreme events.

BCE loss compares the target, which in our case is whether the value is *normal* (0) or *extreme* (1), with the prediction, which takes values close to 0 or 1 after the transformation of of the Sigmoid function. The loss increases exponentially when the difference between the prediction and target increases linearly. It can be defined as

$$BCE(t, p) = -(t \times \log(p) + (1 - t) \times \log(1 - p)), \quad (2)$$

where t and p are the target and predicted values, respectively. However, for datasets with a high imbalance between

the two classes, such as our time series, BCE will favor the prominent class. To solve this problem, we propose a parameterized tunable loss as follows,

$$\mathcal{L} = \beta \times BCE(t, p^\alpha) + (1 - \beta) \times RMSE(t, p), \quad (3)$$

where α and β are parameters that can be tuned. Values $\alpha > 1$ cause the model to predict p values that are higher in general in order to minimize the distance between t and p^α. The BCE part of the loss can be thought of as a blunt instrument that grossly exaggerates all miss-classifications in order to more accurately predict the obscure class, while the RMSE part allows for a more gentle penalty based on the distance between t and p. In other words, the higher α is set, the more *extreme* (class 1) predictions can be obtained. The β meta-parameter controls the strength of the two components of the loss. For time series that are more balanced, β can be small, or even 0.

Evaluation

In this section, we present empirical results for our proposed framework. We are interested in answering the following research questions with regards to prediction effectiveness: (1) What is the effect of adding the GMM indicator to a model? (2) What is the effect of introducing exogenous features? (3) How do the loss function parameters affect performance? (4) How does NEC+ compare against state-of-the-art baselines?

Experimental Settings

Dataset Our dataset includes over 31 years of hourly water level sensor readings for 9 reservoirs and 5 rain gauges in Santa Clara County, CA, which are described in the appendix and listed there in Table 1. After reducing all time series to a common date range, each reservoir and rain sensor time series has 276,226 values. When training all baseline models and the N and C models in NEC+, at least 100,000 random samples were selected, with replacement, from the training set. However, due to the sparsity of *extreme* events, only 50,000 random samples were selected, with replacement, when training the E model. The N model did not use any oversampling ($OS = 0$), but we set $OS = 1$ for both the E and C models, ensuring that all training samples had at least 1 *extreme* event in the prediction section of the sample.

Model Parameters For reservoir 4009, we set M to 4 and ϵ to 1.8. For all other reservoirs, $M = 3$ and $\epsilon = 1.5$. For each reservoir, we tested models with 4 or 6 LSTM layers, and 5 reservoirs use 6 LSTM layers while the rest use 4. We also tested LSTM layer widths of 512 and 1024 nodes and found 1024 node layers were better suited for the N and C models, while E models performed better with 512 nodes across all reservoirs. While $f = 72$ (3 days) was set by our problem definition, we tested $h \in 72, 168, 360, 720$, i.e., $3, 7, 15, 30$ days, and found $h = 360$ to work the best for all reservoirs.

All models were trained using PyTorch 1.9.1+cu102 on a Linux server running CentOS 7.9.2009 equipped with 2x 20-core Intel(R) Xeon(R) Gold 6148 CPUs, 768 GB RAM, and 3 NVIDIA V100 GPUs. Finally, the LSTM layers were trained using an SGD optimizer with learning rate 1E-3,

while the fully connected layers were trained using an Adam optimizer with learning rate 5E-4.

Baseline Methods

We compared our proposed method, NEC+, against a wide array of traditional and state-of-the-art time series and reservoir level prediction methods, which are introduced in the related work section, including,

- ARIMA (Box and Jenkins 1976), a standard statistics-based time series analysis method,
- Prophet (Taylor and Letham 2018), a powerful non-linear regression technique which accounts for seasonality,
- LSTM, which is the standard LSTM (Hochreiter and Schmidhuber 1997) recurrent neural network with the same configuration as our *normal* model,
- DNN-U (Ibáñez et al. 2022), a state-of-the-art univariate LSTM-based encoder-decoder *hydrologic* model used to predict reservoir lagged water levels,
- Attention-LSTM (Le et al. 2021), a state-of-the-art *hydrologic* model used to predict stream-flow, and
- N-BEATS (Oreshkin et al. 2019), a state-of-the-art time series prediction method that outperformed all competitors on the standard M3 (Makridakis and Hibon 2000), M4 (Makridakis, Spiliotis, and Assimakopoulos 2018) and TOURISM (Athanasopoulos et al. 2011) datasets.

Effect of Adding the GMM Indicator Variable
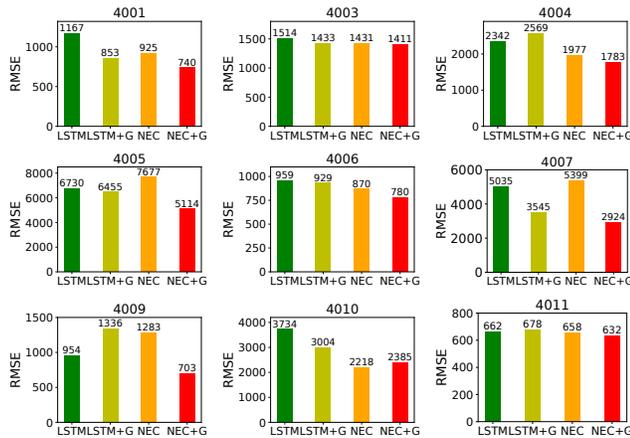
Figure 5: Effectiveness comparison of NEC and LSTM variants with (LSTM+G and NEC+G) and without (LSTM and NEC) GMM indicator variables across all 9 sensors.

In the methods section, we hypothesized that adding the GMM indicator variable as an additional input to our model would help the component N and E models better distinguish between *normal* and *extreme* values. In order to test this hypothesis, we added GMM indicators to both the baseline LSTM model and our basic NEC model, and compared the test set effectiveness (RMSE). Figure 5 shows the result of this experiment for our 9 sensors. In the figure, LSTM+G and NEC+G are the LSTM and NEC models with GMM indicator variables, respectively. Interestingly, adding the

GMM indicator seems detrimental for the baseline LSTM, producing much worse RMSE scores for sensors 4009, and more than 5% RMSE improvements only in sensors 4001, 4007, and 4010. The average RMSE improvement of the LSTM model when adding the GMM indicator across the 9 sensors is only 4%. On the other hand, adding the GMM indicator produces significantly better results in 8 of the 9 sensors for our NEC model, and only slightly worse results for sensors 4010. The average RMSE improvement for NEC is 18%, which is significantly better than the 4% average RMSE improvement of the LSTM model. This shows that, while the GMM indicator may provide some limited benefit to a one-shot model like LSTM, it plays a much more important role in the success of our NEC+ framework.

Effect of Adding Exogenous Variables

An additional benefit may be obtained in NEC+ by including exogenous variables that may provide an additional signal that the model may use to learn the proper prediction function. In our experiments, we used watershed rain gauge time series data from the same times as our primary reservoir water level data to enhance models for reservoirs 4005, 4007, and 4010, as described in the methods section. We compared our NEC+ model with (NEC+G+W) and without (NEC+G) the watershed variables against variations of the baseline LSTM model with (LSTM+W, LSTM+G+W) and without (LSTM, LSTM+G) those same variables. The letter G in all model names indicates the presence of the GMM indicator variable.

Table 2 shows the results of our analysis. We use bold to denote the best results. Interestingly, including the watershed variables in the baseline LSTM and LSTM+G models leads to significantly worse results in most cases, but significantly better results (5%–28% lower RMSE) in the case of the NEC+G model. Our model can benefit more by focusing on normal or extreme prediction individually.

Effect of Loss Function Parameters
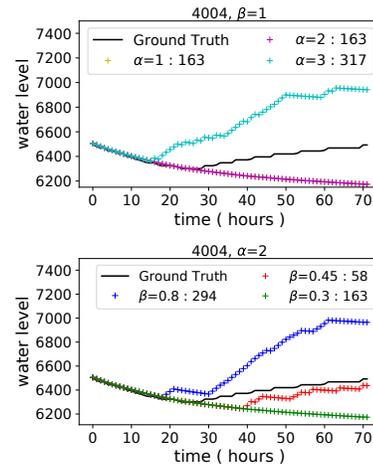
Figure 6: Loss function parameter choices for sensor 4004.

Model/Reservoir	4001	4003	4004	4005	4006	4007	4009	4010	4011
ARIMA	1016.32	1859.70	2501.97	9692.87	1039.38	5854.48	1060.05	3465.20	690.23
Prophet	8469.74	38827.22	95279.31	181607.50	20904.57	187603.80	28629.44	114115.4	2829.26
LSTM	1167.73	1514.90	2342.71	6730.93	959.05	5035.91	954.04	3734.53	662.48
DNN-U	1162.01	1597.72	3989.20	9878.41	983.27	4320.40	1411.63	4257.58	763.73
A-LSTM	878.71	1536.04	2548.56	8919.33	1638.65	13529.86	1064.15	2914.75	700.50
N-BEATS	937.24	1926.74	2280.83	7153.82	960.42	3153.76	1295.90	3162.17	**514.30**
NEC+	**740.19**	**1411.44**	**1783.92**	**4352.74**	**780.46**	**2092.73**	**703.93**	**2275.48**	632.61

Table 1: Effectiveness Comparison (RMSE) of NEC+ Against Baselines for 9 Reservoirs

Model/Reservoir	4005	4007	4010
LSTM	6730.93	5035.91	3734.53
LSTM+W	7568.68	5728.30	4145.16
LSTM+G	6455.90	3545.19	3004.14
LSTM+G+W	9760.62	4128.37	2602.58
NEC+G	5114.49	2924.30	2385.77
NEC+G+W (NEC+)	**4352.74**	**2092.73**	**2275.48**

Table 2: Effectiveness With/Without Exogenous Variables

We proposed a parameterized loss function that we hypothesized would help improve the ability of our C model to pick out the rare extreme values and, as a result, lead to better prediction of future values. As a way to see how the two parameters of our loss function may affect the prediction, we trained several models with different values of α (the BCE power) while keeping β (BCE vs. RMSE strength) constant, and several with varying β while keeping α constant, the results of which can be seen in Figure 6. As expected, increasing the α parameter (top figure) leads to more values being classified as *extreme*, allowing the E model to play a bigger role in the NEC+ model. When α if too large (bottom figure), its effect can be dampened by decreasing the value of β. Therefore, we suggest keeping $\beta = 1$ while tuning α and then tuning β for the best found α.

Effectiveness of NEC+ Against Baselines
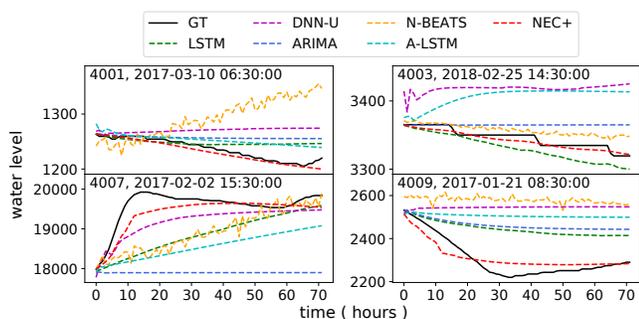
Figure 7: Example 3 days ahead predictions for four sensors.

Our main evaluation question was whether our proposed NEC+ model is an effective method for solving the 3-day ahead prediction problem in time series with *extreme* events such as our 9 reservoirs. To answer this question, we compared NEC+ against a variety of traditional and state-of-the-art methods and Table 1 presents the RMSE results of all

these models. Equivalent MAPE values are included in Table 3in the appendix. Values in bold are the best (lowest) RMSE for each sensor. Our model significantly outperforms all traditional methods (ARIMA, Prophet, and LSTM) and state-of-the-art methods DNN-U and A-LSTM for all 9 sensors. However, the results for NEC+ are, on average, 37% and 33% better than those of DNN-U and A-LSTM, respectively, across all 9 sensors. Moreover, DNN-U and A-LSTM were unable to outperform the traditional ARIMA or LSTM baselines for 6 and 3 out of the 9 sensors, respectively, pointing to their overall instability. The N-BEATS model was the most competitive, outperforming NEC+ on only one sensor, 4011. However, NEC+ results are significantly better than those of N-BEATS (Wilcoxon T test s=1, p=0.0078).

Figure 7 shows some example 3-day predictions from our test set for four of the sensors (due to lack of space). Predicted time series for other sensors are included in the technical appendix. We did not include Prophet in the results as the model performed very poorly and would impede visualizing the performance of the remaining models. Overall, NEC+ is able to more closely predict the ground truth water level values, both in the presence of *extreme* events and during normal conditions. ARIMA often misses the mark and DNN-U, LSTM, A-LSTM, and N-BEATS sometimes follow the trend of the ground truth and sometimes do not. Overall, NEC+ shows it can more closely account for extreme changes in the time series.

Conclusion

In this work, we presented a novel composite framework and model, NEC+, designed to better account for rare yet important extreme events in long single- and multi-variate time series. Our framework learns distinct regression models for predicting *extreme* and *normal* values, along with a merging classifier that is used to choose the appropriate model for each future event prediction. NEC+ uses clustering to dynamically produce distribution indicators, which improves the model's robustness to the occurrence of severe events. In addition, to improve training performance, our framework uses selected backpropagation and a two-level sampling algorithm to accommodate imbalanced extreme data. A parameterized loss function is also proposed to improve the NEC+ classifier performance. Extensive experiments using more than 31 years of reservoir water level data from Santa Clara County, CA, showed that NEC+ provided significantly better predictions than state-of-the-art baselines (Wilcoxon T test p-values between 0.0039 and 0.0078).

Acknowledgments

References

Anastasiu, D. C.; Gaul, J.; Vazhaeparambil, M.; Gaba, M.; and Sharma, P. 2020. Efficient City-Wide Multi-Class Multi-Movement Vehicle Counting: A Survey. *Journal of Big Data Analytics in Transportation*, 2(3): 235–250.

Ariyo, A. A.; Adewumi, A. O.; and Ayo, C. K. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 106–112. : IEEE Xplore.

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

Bose, B.; Downey, T.; Ramasubramanian, A. K.; and Anastasiu, D. C. 2022. Identification of Distinct Characteristics of Antibiofilm Peptides and Prospection of Diverse Sources for Efficacious Sequences. *Frontiers in Microbiology*, 12.

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

Chorowski, J. K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

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

Du, N.; and Liang, X. 2021. Short-term water level prediction of Hongze Lake by Prophet-LSTM combined model based on LAE. In *2021 7th International Conference on Hydraulic and Civil Engineering Smart Water Conservancy and Intelligent Disaster Reduction Forum (ICHCE SWIDR)*, 255–259. : IEEE Xplore.

Hewage, P.; Trovati, M.; Pereira, E.; and Behera, A. 2021. Deep learning-based effective fine-grained weather forecasting model. *Pattern Analysis and Applications*, 24(1): 343–366.

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

Hua, S.; Kapoor, M.; and Anastasiu, D. C. 2018. Vehicle Tracking and Speed Estimation from Traffic Videos. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, volume 1 of *CVPRW'18*, 153–1537. : IEEE.

Huang, G.-B.; Zhou, H.; Ding, X.; and Zhang, R. 2012. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2): 513–529.

Ibañez, S. C.; Dajac, C. V. G.; Liponhay, M. P.; Legara, E. F. T.; Esteban, J. M. H.; and Monterola, C. P. 2022. Forecasting Reservoir Water Levels Using Deep Neural Networks: A Case Study of Angat Dam in the Philippines. *Water*, 14(1).

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

Li, Y.; Xu, J.; and Anastasiu, D. C. 2022. An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks. arXiv:2211.15891.

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

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

Mohan, S.; Mullapudi, S.; Sammeta, S.; Vijayvergia, P.; and Anastasiu, D. C. 2019. Stock Price Prediction using News Sentiment Analysis. In *2019 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, BDS 2019, 205–208. : IEEE.

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

Pei, Y.; Liu, Y.; Ling, N.; Liu, L.; and Ren, Y. 2021. Class-Specific Neural Network for Video Compressed Sensing. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5.

Qi, Y.; Zhou, Z.; Yang, L.; Quan, Y.-N.; and Miao, Q. 2019. A Decomposition-Ensemble Learning Model Based on LSTM Neural Network for Daily Reservoir Inflow Forecasting. *Water Resources Management*, 33: 4123 – 4139.

Rangapuram, S. S.; Seeger, M. W.; Gasthaus, J.; Stella, L.; Wang, Y.; and Januschowski, T. 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31.

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

Taylor, S. J.; and Letham, B. 2018. Forecasting at Scale. *The American Statistician*, 72(1): 37–45.

Tyralis, H.; and Papacharalampous, G. 2021. Quantile-based hydrological modelling. *Water*, 13(23): 3420.

Valipour, M.; Banihabib, M. E.; and Behbahani, S. M. R. 2013. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *Journal of Hydrology*, 476: 433–441.

Waltrup, L. S.; Sobotka, F.; Kneib, T.; and Kauermann, G. 2015. Expectile and quantile regression—David and Goliath? *Statistical Modelling*, 15(5): 433–456.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. PMLR.

Yang, S.; Yang, D.; Chen, J.; and Zhao, B. 2019. Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model. *Journal of Hydrology*, 579: 124229.

Zhang, Z.; Qin, H.; Yao, L.; Liu, Y.; Jiang, Z.; Feng, Z.; Ouyang, S.; Pei, S.; and Zhou, J. 2021. Downstream Water Level Prediction of Reservoir based on Convolutional Neural Network and Long Short-Term Memory Network. *Journal of Water Resources Planning and Management*, 147(9): 04021060.