SEED: An Effective Model for Highly-Skewed Streamflow Time Series Data Forecasting

Yanhong Li, Jack Xu, David C. Anastasiu

Santa Clara University
Problem: predicting long-term streamflow values with rain off data

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

\[ x_1 \text{ to } x_T : \text{the input sequence} \]
\[ x_{T+1} \text{ to } x_{T+H} : \text{the output sequence} \]

In our research: \( H = 3 \times 24 \times 4 = 288 \), with majority of normal values and much fewer extreme values which cause the data skewness to one side.

Challenges:

• Long-range dependencies.
• Rare but important extreme values; very imbalanced data.

Goal:

• An end-to-end extreme-adaptive model;
• Long sequence forecasting (predicted length = 288);

Dataset:

• Four groups of hydrologic datasets from Santa Clara County, CA.
• Namely Ross, Saratoga, UpperPen, and SFC, named after their respective locations.
High skewness and kurtosis scores indicate that there is significant deviation from a normal distribution in our data!

Four streams: Ross, Saratoga, UpperPen, and SFC.
Hydro year: from September to May.

<table>
<thead>
<tr>
<th>Statistic / Stream</th>
<th>Ross</th>
<th>Saratoga</th>
<th>UpperPen</th>
<th>SFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>2.91</td>
<td>5.77</td>
<td>6.66</td>
<td>20.25</td>
</tr>
<tr>
<td>max</td>
<td>1440.00</td>
<td>2210.00</td>
<td>830.00</td>
<td>7200.00</td>
</tr>
<tr>
<td>min</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>median</td>
<td>0.17</td>
<td>1.00</td>
<td>3.20</td>
<td>1.20</td>
</tr>
<tr>
<td>variance</td>
<td>597.22</td>
<td>711.09</td>
<td>452.90</td>
<td>12108.14</td>
</tr>
<tr>
<td>skewness</td>
<td>19.84</td>
<td>19.50</td>
<td>13.42</td>
<td>18.05</td>
</tr>
<tr>
<td>kurtosis</td>
<td>523.16</td>
<td>697.78</td>
<td>262.18</td>
<td>555.18</td>
</tr>
</tbody>
</table>
Motivation: achieving the best overall prediction performance, without sacrificing either the quality of normal or of extreme predictions.

Root Mean Square Error (\textit{RMSE})
Mean Absolute Percentage Error (\textit{MAPE})

Proposed Methods:

\textbf{Framework:} Segment-Expandable Encoder-Decoder (SEED) model, which is the first to integrate segment representation learning with a multi-tiered encoder-decoder framework.

\textbf{Importance-enhanced sampling strategy:} embedded within the SEED model, allowing it to skillfully identify key features and trends in datasets.

\textbf{Representation Learning:} A unique regularization strategy that incorporates a Kullback-Leibler divergence regularization loss term across multiple stacked layers, thereby increasing the model's robustness against anomalous events with divergent distributions.
PLR splits a series into several segments such that the maximum error of each segment does not exceed a threshold;

**Prior work:** PLR describes the linear relationship of the multi-segment representation, mainly works as a preprocessing step to reduce both the space and computational cost of storing and transmitting time series.

**Our work:** inspired by PLR, SEED learns **nonlinear segment representations** for heavily skewed long term time series.
SEED framework:

- Comprises three core components: embedding, encoder, and decoder.
- The encoder generates a unique hidden state and a cell state which serve as the initial values for the corresponding layers in the decoder.
- Each decoder layer is assigned a distinct task, as they represent the mean value distribution of different lengths of subsegments in the predicted sequence.
Convolutional Embedding Layers:

<table>
<thead>
<tr>
<th>Embedding</th>
<th>Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONV-1</td>
<td>Tanh</td>
</tr>
<tr>
<td>CONV-2</td>
<td>Tanh</td>
</tr>
<tr>
<td>CONV-3</td>
<td>Tanh</td>
</tr>
<tr>
<td>CONV-3</td>
<td>Tanh</td>
</tr>
<tr>
<td>CONV-5</td>
<td>Tanh</td>
</tr>
</tbody>
</table>

CNN Layers:
- different kernel sizes to extract features at different spatial scales.
- subsequent tanh activation function;
- lower level: larger kernel sizes, capturing broader patterns and global context.
- higher level: smaller kernel sizes, capturing local patterns and fine grained details.
Decoder Architecture:

- First level: the output length is 4, which is meant to predict the mean values of 4 segments, each of which contains $288/4 = 72$ points in the forecasted series.

- Second level: the 4 outputs are expanded to 16, each represents the mean value of $288/16 = 18$ points.

- Expansion: $(a, b, c, d)$ becomes $(a, a, a, a, b, b, b, b, c, c, c, c, d, d, d, d)$, in the high-dimensional hidden space.

- By predicting the mean value of different length sub-segments, extreme values are represented and spread across multiple levels in the hierarchy, leading to higher mean values in the segments containing them.
Multiple-Objective Loss Functions:

\[ \text{KL}(p \| q) = \sum p(x) \log \left( \frac{p(x)}{q(x)} \right) , \]

\[ \mathcal{L}_i = \text{KL} \left( \text{softmax}(p_{m_i}), \text{softmax}(g_{m_i}) \right) , \]

\[ \mathcal{L} = \text{RMSE}(\hat{y}, y) + \lambda \times \left( \sum_{i=1}^{k} \mathcal{L}_i \right) , \]

Motivations:

- Kullback-Leibler divergence loss acts as a regularization term that encourages the model to match the target distribution while balancing the sequence generation loss.

- \( p_{m_i} \) represents the predicted segment mean values in the \( ith \) layer, while \( g_{m_i} \) is the vector of computed ground truth mean values for the segments in the \( ith \) layer.
Importance-Enhanced Oversampling Policy:

**Input**: Dataset with training and inference sequences  
**Output**: Oversampled training set

**Procedure** Oversampling:

while *training set size is not satisfied* do

- Randomly sample a sequence including training and inference sections;
- if *maximum value in inference section > T* then
  - Mark sequence as important;
  - Move maximum value to the middle of the inference section of the sequence;
  - foreach *index I in the sequence with step size S do*
    - Sample starting at *I*;
    - Add sampled sequence to oversampled training set;
  - end
- else
  - Add sequence to oversampled training set;
- end

Steps:

- important sequences: maximum values in the inference section of the series exceed a threshold *T*;
- moving the maximum value to the middle of the inference section;
- multiple iterations of sampling from the beginning with a specified step size *S*. 
Baselines:

• FEDFormer
• InFormer
• NLinear
• DLinear
• NEC+
• EnDecoder, the common encoder-decoder model built with LSTM layers.
### Main results:

In comparison to the three second-best models (NEC+, Nbeats and EnDecoder), SEED achieved, on average, relative RMSE reduction of 31.44%, 34.68%, and 29.67% across the datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Metric</th>
<th>Ross All</th>
<th>Ross High</th>
<th>Ross Low</th>
<th>Saratoga All</th>
<th>Saratoga High</th>
<th>Saratoga Low</th>
<th>Upperpen All</th>
<th>Upperpen High</th>
<th>Upperpen Low</th>
<th>SFC All</th>
<th>SFC High</th>
<th>SFC Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEDformer</td>
<td>RMSE</td>
<td>6.49</td>
<td>30.82</td>
<td>2.51</td>
<td>6.85</td>
<td>11.95</td>
<td>4.59</td>
<td>2.38</td>
<td>17.68</td>
<td>1.07</td>
<td>24.15</td>
<td>94.28</td>
<td>6.68</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>2.49</td>
<td>5.27</td>
<td>2.04</td>
<td>2.26</td>
<td>1.50</td>
<td>2.60</td>
<td>1.02</td>
<td>2.37</td>
<td>0.90</td>
<td>2.81</td>
<td>1.70</td>
<td>3.09</td>
</tr>
<tr>
<td>Informer</td>
<td>RMSE</td>
<td>9.14</td>
<td>31.00</td>
<td>5.56</td>
<td>4.89</td>
<td>13.64</td>
<td>1.01</td>
<td>5.33</td>
<td>16.26</td>
<td>4.40</td>
<td>19.00</td>
<td>85.40</td>
<td>2.46</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>5.45</td>
<td>5.80</td>
<td>3.59</td>
<td>0.73</td>
<td>1.40</td>
<td>0.43</td>
<td>4.21</td>
<td>2.70</td>
<td>4.34</td>
<td>0.54</td>
<td>0.71</td>
<td>0.49</td>
</tr>
<tr>
<td>Nlinear</td>
<td>RMSE</td>
<td>5.84</td>
<td>32.12</td>
<td>1.54</td>
<td>4.98</td>
<td>14.61</td>
<td>0.70</td>
<td>1.74</td>
<td>15.07</td>
<td>0.61</td>
<td>18.43</td>
<td>83.31</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>1.62</td>
<td>4.89</td>
<td>1.09</td>
<td>0.75</td>
<td>1.74</td>
<td>0.31</td>
<td>0.57</td>
<td>1.69</td>
<td>0.47</td>
<td>0.87</td>
<td>1.03</td>
<td>0.83</td>
</tr>
<tr>
<td>Dlinear</td>
<td>RMSE</td>
<td>6.90</td>
<td>30.96</td>
<td>2.97</td>
<td>4.06</td>
<td>7.63</td>
<td>2.48</td>
<td>3.25</td>
<td>14.01</td>
<td>2.33</td>
<td>23.64</td>
<td>79.76</td>
<td>9.65</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>2.79</td>
<td>4.03</td>
<td>2.58</td>
<td>1.31</td>
<td>0.85</td>
<td>1.51</td>
<td>2.04</td>
<td>1.68</td>
<td>2.07</td>
<td>4.02</td>
<td>1.04</td>
<td>4.76</td>
</tr>
<tr>
<td>NEC+</td>
<td>RMSE</td>
<td>9.33</td>
<td>38.34</td>
<td>4.58</td>
<td>1.95</td>
<td>5.55</td>
<td>0.35</td>
<td>1.94</td>
<td>13.92</td>
<td>0.92</td>
<td>16.39</td>
<td>76.63</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>4.53</td>
<td>8.33</td>
<td>3.91</td>
<td>0.21</td>
<td>0.30</td>
<td>0.17</td>
<td>0.80</td>
<td>0.84</td>
<td>0.80</td>
<td>0.55</td>
<td>0.61</td>
<td>0.54</td>
</tr>
<tr>
<td>NBeats</td>
<td>RMSE</td>
<td>5.16</td>
<td>30.09</td>
<td>1.08</td>
<td>3.60</td>
<td>9.44</td>
<td>1.01</td>
<td>1.23</td>
<td>13.20</td>
<td>0.21</td>
<td>31.47</td>
<td>95.33</td>
<td>15.55</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>1.25</td>
<td>3.17</td>
<td>0.94</td>
<td>0.70</td>
<td>1.21</td>
<td>0.47</td>
<td>0.25</td>
<td>0.78</td>
<td>0.20</td>
<td>3.24</td>
<td>0.88</td>
<td>3.83</td>
</tr>
<tr>
<td>EnDecoder</td>
<td>RMSE</td>
<td>5.58</td>
<td>30.81</td>
<td>1.45</td>
<td>1.93</td>
<td>5.69</td>
<td>0.26</td>
<td>2.95</td>
<td>16.33</td>
<td>1.80</td>
<td>17.46</td>
<td>79.04</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>1.62</td>
<td>3.72</td>
<td>1.28</td>
<td>0.16</td>
<td>0.29</td>
<td>0.11</td>
<td>1.81</td>
<td>2.34</td>
<td>1.76</td>
<td>0.84</td>
<td>0.96</td>
<td>0.80</td>
</tr>
<tr>
<td>SEED</td>
<td>RMSE</td>
<td>4.23</td>
<td>29.74</td>
<td>0.05</td>
<td>1.67</td>
<td>5.14</td>
<td>0.12</td>
<td>1.07</td>
<td>12.83</td>
<td>0.07</td>
<td>14.44</td>
<td>70.04</td>
<td>0.59</td>
</tr>
</tbody>
</table>
|          | MAPE   | 0.11     | 0.53      | 0.04     | 0.09         | 0.19          | 0.05         | 0.10         | 0.57          | 0.06         | 0.20     | 0.42    | 0.14    

- Univariate Long-Term (h = 288) Series Forecasting Results.
- Over 1600 test points in the test set were inferenced on all datasets.
- The best results are in bold and the second best results are underlined.
- “All” represents the average RMSE of all test samples compared with the ground truth. “High” means larger than the mean value; “Low” includes test samples lower than the mean value.
Example comparisons with the second best baselines:
Effect of the Importance-Enhanced Oversampling Policy:

- To evaluate the impact of this policy, we increased the threshold $T$ while simultaneously decreasing the step size $S$.
- Increasing the threshold $T$ and decreasing the step size $S$ had a positive impact on the results.
- There is an optimal threshold $T$ value beyond which the policy’s effectiveness plateaus.
Effect of the KL Regularization Terms & Segment Expanding:

We just use $T=4, S=8$ as an example:

- $T4S8$: 5-layer without regularization loss terms.
- $T4S8-3L$: 3-layer SEED with regularization loss terms.
- $T4S8-Regu$: 5-layer SEED with regularization loss terms, which gives the best result.
Q & A