Project Description: Improving NWM Forecasts Using Deep Learning Post-processing

AI Course Project

March 13, 2025

1. Introduction and Context

Accurate runoff forecasting is crucial for flood prediction, water resources management, and various hydrologic analyses. The United States National Water Model (NWM), developed by the National Oceanic and Atmospheric Administration (NOAA), provides short-range forecasts of runoff across the continental US. However, the NWM forecasts can contain systematic and time-dependent errors, particularly for longer lead times.

Recent advances in Deep Learning (DL) show promise for improving the performance of physically based hydrologic models by using DL approaches to post-process the forecasts. In particular, sequence models such as Long Short-Term Memory (LSTM) networks or other architectures can learn and predict the errors (residuals) of the NWM, thereby producing a corrected runoff forecast that better matches observed data. The attached paper [1] illustrates the concept of combining NWM outputs with observed runoff and precipitation inputs in a deep learning framework for improved forecasts.

2. Project Goal

In this project, you will apply a Deep Learning technique of your choice (e.g., LSTM, GRU, Transformers, or any other neural network architecture) to **improve the performance of the NWM** for short-range runoff forecasting. Specifically, you will:

(a) Preprocess NWM and United States Geological Survey (USGS) observed runoff data for two stations in the US.

(b) Train, validate, and test your DL-based post-processing model to correct NWM forecasts for lead times of $1-18$ hours.

(c) Evaluate the performance of your corrected forecasts using standard hydrologic metrics.

[1] Han, H. & Morrison, R. R. (2022). *Improved runoff forecasting performance through error predictions using a deep-learning approach.* Journal of Hydrology, 608, 127653.

3. Data Description and Usage Rules

- **Data Sources:**
  - **NWM forecasts:** Hourly short-range forecasts (lead times 1 – 18 h) for two different stations in the US, from April 2021 to April 2023.
  - **USGS observations:** Hourly observed runoff data (same time span, April 2021 to April 2023).
  - **Other inputs (optional):** Precipitation data or other meteorological forcings (e.g., from NOAA or other public datasets), if desired for your DL model.
- **Train/Validation/Test Split:**
  - **Training/Validation:** April 2021 – September 2022
  - **Testing:** October 2022 – April 2023

  Important: It is strictly forbidden to use any test-set information (October 2022 – April 2023) in the training or validation process.

4. Tasks

4.1 Data Preprocessing

- Perform any necessary cleaning (e.g., handling missing values, unit conversions, time alignment, etc.).
- Split your data into training/validation/testing subsets, respecting the time frames indicated.

4.2 Model Development

- Select any Deep Learning model architecture (e.g., LSTM, GRU, CNN-LSTM, Transformers, or a simple feed-forward network with temporal features) that can learn from past forecast errors and precipitation (if used).
- Train the model to predict *the error* (or *the residual*) between the NWM forecast and the observed runoff at each lead time. Alternatively, you may directly learn the corrected runoff if you prefer.
- Ensure that no future/test data is used during training or validation.

4.3 Model Testing and Analysis

- Use the **test set** (October 2022 – April 2023) to evaluate the performance of your final trained model.

- Generate a **corrected forecast** for each hour from $1-18$ h lead time.
- Compare the corrected forecast against:
  (a) The *original* NWM forecast,
  (b) The *observed* USGS runoff.

5. Required Results and Plots

(1) **Box-plot of Observed, NWM, and Corrected Runoff:** Produce a *box-plot diagram* of:

- Observed (USGS) runoff
- Forecasted (NWM) runoff
- Corrected (Your model) runoff

for **each lead time from $\mathbf{1-18}$ h**, similar in concept to Figure 4 in the reference paper but in box-plot form. Each lead time should have three box-plots (Observed, NWM, Corrected).

(2) **Box-plots of Evaluation Metrics:** Compute and display the following four metrics, for each lead time from $1-18$ h, in a *box-plot* format:

- Coefficient of correlation (**CC**)
- Root mean square error (**RMSE**)
- Percent bias (**PBIAS**)
- Nash-Sutcliffe Efficiency (**NSE**)

Present these metrics in a manner that shows the distribution or variation across the test period. One way is to have four separate box-plots (one for each metric), each containing 18 boxes (one for each lead time). Make sure you include a direct comparison between the NWM forecasts and your *corrected* model forecasts.

6. Deliverables

Each group of two or three students will submit the following:

(a) **Technical Report:**

- In the style of a scientific paper (e.g., IEEE or typical hydrology journal format).
- Should detail:
  (a) **Methodology:** Data sources, preprocessing steps, and a thorough description of your deep learning model and training approach.

(b) **Results:** Plots and tables, including the required box-plots, discussion of results, error metrics, and insights gained.
(c) **Implications:** How well does your approach improve forecasts compared to the raw NWM forecasts? Discuss any challenges or limitations.

- Include a link to a *private GitHub repository* with all the code needed to reproduce your results. Ensure it is clearly documented (e.g., `README.md` describing how to run your code).

(b) **Class Presentation:**

- A short (10–15 minute) presentation covering:
  (a) Your approach and workflow
  (b) Key results and findings
  (c) Future directions or improvements
- All group members should participate in presenting.

7. Important Notes

- **Data leakage is strictly forbidden.** You must not use the test set (October 2022 – April 2023) for any training or hyperparameter tuning.
- Collaboration among group members is mandatory; each member should contribute significantly to both the technical work and final deliverables.
- Feel free to explore different deep learning architectures, provided you justify your design choices.
- You may use any open-source frameworks (e.g., TensorFlow, PyTorch, Keras, etc.).
- Cite any external packages, papers, or references that you use.

8. Tentative Grading Scheme

(a) **Correctness and completeness of the technical report (40%)**

(b) **Quality of analysis, discussion, and interpretation of results (30%)**

(c) **Code quality, reproducibility, and documentation (20%)**

(d) **Presentation (10%)**

9. Timeline

- **Model development and preliminary results:** Approximately 3–4 weeks.
- **Report and presentation preparation:** 1–2 weeks after the model results are finalized.

10. Conclusion

This project will give you hands-on experience in combining physical modeling (the National Water Model) with data-driven post-processing via modern deep learning methods. You will gain insights into data handling, model design, and the critical importance of properly evaluating model forecasts with an independent test set. By improving upon a real-world forecasting model, you will also learn how data-driven error correction can benefit hydrologic prediction in operational contexts.

Good luck, and we look forward to seeing your innovative approaches and results!