Roman Chernenko Vitaly Bondar

“Water Supply Forecast Rodeo: Final Prize Stage” competition report by “ck-ua” team

Abstract

In our report, we detail the methods and outcomes of our participation in the "Water Supply Forecast Rodeo: Final Prize Stage" competition, where we aimed to predict water flow levels with constrained data availability. Our approach centered on using a Multi-Layer Perceptron (MLP) neural network with four layers, which proved to be the most effective within the given constraints. The network was constructed to simultaneously predict the 10th, 50th, and 90th percentile targets of water level distribution. We experimented with various network enhancements and dropout regularization but observed no substantial improvement in the model's performance.

The proposed solution is highly efficient and fast.

For our data sources, we relied on the NRCS and RFCs monthly naturalized flow, USGS streamflow, USBR reservoir inflow and NRCS SNOTEL data, all of which were meticulously normalized and encoded to serve as features for our training process. We propose a novel approach for using SNOTEL data by training specialized RANSAC mini-models for each site separately. For each of these mini-models the list of the used SNOTEL stations are selected by heuristic approach.

We also employed data augmentation due to the limited size of our training dataset, which allowed us to artificially expand our sample set.

As a result, we demonstrate that water supply forecasts may be done with the lightweight simple but robust, and powerful single model for all the flow sites used in the competition.

Technical Approach

Algorithm and Architecture Selection

In light of the constrained dataset at our disposal, the decision was made to opt for compact neural networks. The utilization of small convolutional neural networks[6] was ruled out due to the unique characteristics of our data, where certain data points could be overlooked or exhibit varying distances to neighboring points. Although we initially planned to explore the feasibility of employing small transformer models[10], this avenue remained unexplored due to time constraints.

Consequently, our chosen solution revolves around a feedforward Multi-Layer Perceptron (MLP) network[7] comprising four layers with Rectified Linear Unit (ReLU)[1] activations. Our investigation revealed that predicting the residual value for the seasonal water level is a

relatively simpler task for the model, and thus, this approach was incorporated into the final submission (Fig. 1).

The resultant architecture is designed to yield three outputs, enabling predictions for the 10th, 50th, and 90th percentiles of the distribution in a single execution.

Furthermore, we conducted experiments with various enhancements, including the incorporation of residual connections within the MLP, concatenation of all inputs to the last layer, adopting a densenet-like structure, and introducing additional embeddings on top of the input features. Regrettably, none of these approaches yielded significant improvements in the overall score.

Interestingly, incorporating dropout regularization[3, 8] into our solution sometimes reduced the final score. But we use this regularization method for the better generalization.

Additionally our team explored some bayesian based approaches for estimation of the full probability density function of the target value. But we faced many obstacles in the estimation of the posterior probability and we refused this direction.

An important aspect of our approach is fast inference. The model may handle all the Hindcast test set in 80 seconds, including features preparation.

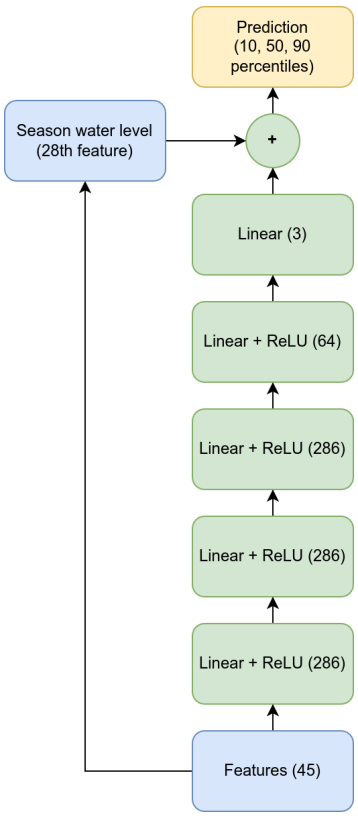

Fig. 1 Model architecture

Data Sources and Feature Engineering

Due to the big difference between the target volumes of different sites, all target volume values from the training dataset were normalized to 99th percentile independently for each site_id. So after normalization “0” values mean no water flow and “1” values mean almost the maximum possible water flow during all available historical data.

The site_id value was encoded with the one-hot vector of size 26.

We used the following data sources in the final version of the solution:

- NRCS and RFCs monthly naturalized flow
- USGS streamflow
- NRCS SNOTEL
- USBR reservoir inflow

NRCS and RFCs monthly naturalized flow data preprocessing and data engineering

All monthly naturalized flow data that is calculated before the issued data is loaded. All rows with empty values are dropped. The monthly flow values normalized to the same coefficients as train target volumes.

We used the following features for monthly naturalized flow data:

- sum of monthly flow data from the start of water season to the month before the issue date
- sum of monthly flow data from the start of forecast season (March or April depending on site_id) to the month before issue date
- normalized elapsed time from the start of water season
- normalized elapsed time from the start of forecast season

USGS streamflow data preprocessing and data engineering

All USGS streamflow data that measured before the issued date is loaded. We used only the '00060_Mean' column from .csv files that correspond to discharge in cubic feet per second[9]. All streamflow values were converted to acre*foot/day values and then normalized in the same way as monthly naturalized flow data.

The images below on Fig. 2 represent the correlation between the aggregation of USGS streamflow to the target volume for two different sites. For some sites the correlation is very high and for other correlation is significant too, but noisy. The difference between aggregated USGS streamflow and target volume for some sites can be related to some river regulation as suggested in the data source description.

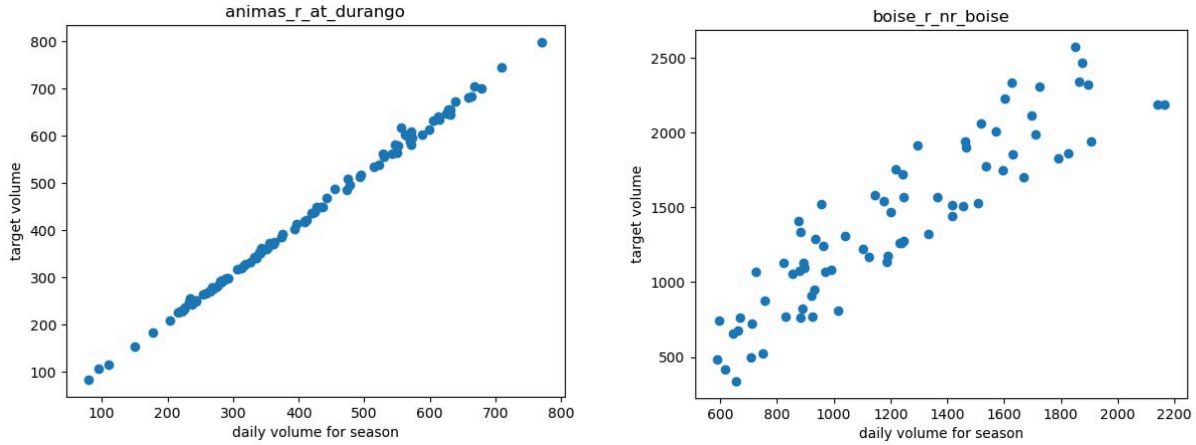

Fig.2 The correlation between aggregated USGS streamflow data and target volume

We used the following features for USGS streamflow data:

- sum of streamflow daily volumes from the start of water season to the day before issue date
- a sum of streamflow daily volumes from the start of the forecast season (March or April depending on site_id) to the day before the issue date
- normalized elapsed time from the start of the water season
- normalized elapsed time from the start of the forecast season

USBR reservoir inflow data preprocessing and data engineering

The preprocessing and data engineering step for the USBR reservoir inflow data is quite similar to the USGS streamflow data. We found the relevant data on the USBR RISE API only for the next 4 sites:

- taylor_park_reservoir_inflow
- fontenelle_reservoir_inflow
- american_river_folsom_lake
- boysen_reservoir_inflow

All USBR reservoirs inflows values were converted to acre*foot/day values and then normalized in the same way as monthly naturalized flow data.

We used the same features as for USGS streamflow data.

NRCS SNOTEL data preprocessing and data engineering

The main challenge of using SNOTEL data as a feature for the model is related to the different amounts of stations for each basin. The first attempt at using SNOTEL data is to choose the SNOTEL station that has the best correlation from aggregated precipitation to the target volume. But in this case we lost a lot of data from other SNOTEL stations.

The final solution uses a linear regression from several SNOTEL aggregated precipitation to target volume as normalization functions. The special greedy heuristic algorithm is implemented to choose the best SNOTEL subset of stations for each site independently. The same normalization function was also applied to snow-water equivalent values of SNOTEL data. The idea behind using several SNOTEL stations that we can make better spatial coverage of the basin and surrounding land and get information from points with different weather conditions.

The RANSAC regression model implementation from scikit-learn package is used for better outlier filtering.

Using the combination of SNOTEL stations makes a significant boost for the score on the local cross-validation and on the Hindcast leaderboard.

There is a significant correlation between the maximum aggregated precipitation value and the target volumes as shown in the charts below (Fig. 3).

We used only 'PREC_DAILY' and 'WTEQ_DAILY' features from SNOTEL data that correspond to Precipitation data and Snow water equivalent data respectively.

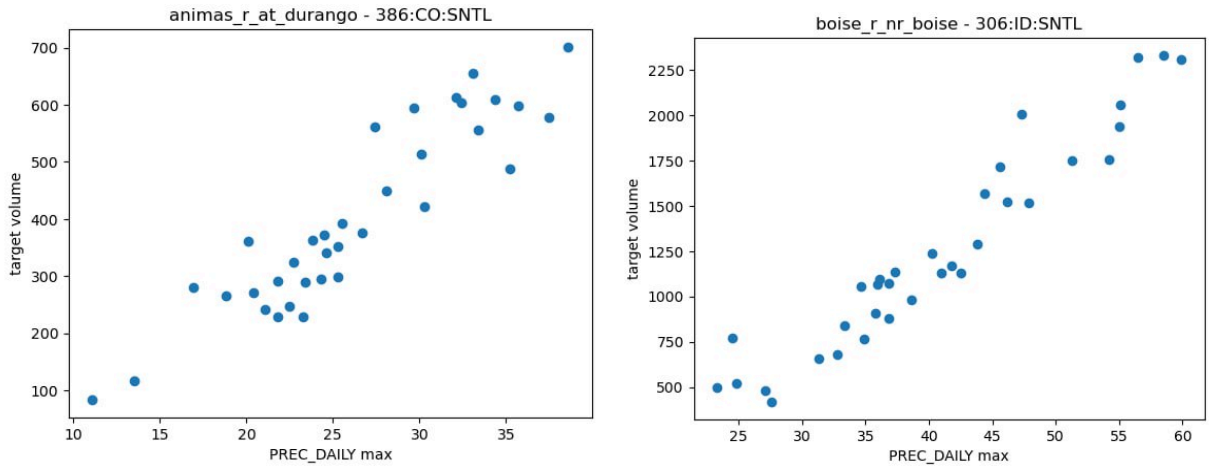

Fig. 3 The correlation between SNOTEL precipitation values and target volumes

We used the next preprocessed features for NRCS SNOTEL data:

- aggregated normalized precipitation for the day before the issue date
- latest normalized snow-water-equvalent value for the date before the issue date
- maximum of snow-water-equivalent values from the start of the water season to the day before the issue date
- normalized elapsed time from the start of the water season
- normalized elapsed time from the start of the forecast season

Other data sources not included in the final solution

We also tried the next data sources too, but did not find any significant improvement and therefore not included them in the final solution:

- CDEC Snow Sensor Network
- UA/SWANN
- CPC Seasonal Outlooks
- Palmer Drought Severity Index (PDSI) from gridMET
- SNODAS
- Seasonal meteorological forecasts from Copernicus
- NLDAS-2 forcing data
- Oceanic Niño Index (ONI)

The UA/SWANN, SNODAS, NLDAS-2 and PDSI datasets are available as raster georeferencing maps in different geographical projections and therefore require preprocessing to use in our model. The main idea of preprocessing is considered in the min/mean/max aggregation of raster data inside each basin polygon separately.

Work [2] showed that limited direct soil moisture observations could improve statistical forecast accuracy. We tried to use the PDSI dataset as the source of soil moisture data, but we could not achieve noticeable results with it.

We also tried to use the CDEC stations network as additional information for precipitation and snow-water-equivalent data for the basins in California where there is a lack of SNOTEL stations. However using CDEC stations does not improve the score on the cross-validation.

We also tried to use the sequence of three-month CPS seasonal outlooks and 6-months forecasts from Seasonal meteorological forecasts from Copernicus. We calculated aggregated features from the start of the water season to the end of the season as a long-term weather forecast, but it did not change the cross-validation score either.

Several basins at north of the USA also extend to Canada and will also be useful to try to include SNOTEL-equivalent data about precipitation from Canada to improve the quality of forecasting. Seems this kind of data is available on www.nrcs.usda.gov website, but we found it too late to request it as an available data source.

Both SNODAS and NLDAS-2 models have coverage in Canada too, but data usage from these sources has not changed the overall situation.

Also we had an idea to use MODIS Snow Cover product to detect snow melting date outside of SNOTEL stations, but not implemented it due to lack of time.

Features summary

#	Data source	Feature description
0-25		One-hot encoding of site_id values
26	monthly naturalized flow	The sum of monthly flow data from the start of the water season to the month before the issue date
27		Normalized elapsed time from the start of the water season
28		The sum of monthly flow data from the start of the forecast season to the month before the issue date
29		Normalized elapsed time from the start of the forecast season
30	USGS streamflow	The sum of streamflow daily volumes from the start of the water season to the day before the issue date
31		Normalized elapsed time from the start of the water season
32		The sum of streamflow daily volumes from the start of the forecast season to the day before the issue date
33		Normalized elapsed time from the start of the forecast season
34	SNOTEL	Boolean flag that SNOTEL data is available for this site_id and year
35		Aggregated normalized precipitation for the day before the issue date
36		The latest normalized snow-water-equvalent value for a date before the issue date

37		Maximum of snow-water-equivalent values from the start of the water season to the day before the issue date
38		Normalized elapsed time from the start of the water season
39		Normalized elapsed time from the start of the forecast season
40	USBR reservoir inflow	Boolean flag that USBR reservoir inflow data is available for this site_id and year
41		The sum of streamflow daily volumes from the start of the water season to the day before the issue date
42		Normalized elapsed time from the start of the water season
43		The sum of monthly flow data from the start of the forecast season to the month before the issue date
44		Normalized elapsed time from the start of the forecast season

Table 1. Features details used for the final model

Uncertainty Quantification

Our first attempt to handle uncertainty was to predict the full probability density function (in the discretized version), but we were unable to correctly estimate posterior probability for each situation.

So we decided to estimate 3 target percentiles directly by prediction of these 3 values as independent predicted values.

To force model to correctly predict needed percentiles we used the next loss function:

$L_{total} = \frac{L_{10} + L_{50} + L_{90}}{3}$.

Each component of the loss is the quantile loss[4] $L_x = \frac{2}{N}\sum_{i}^{N}(x * max(0,\ t - p_x) + (1 - x) * max(0,\ p_x - t))$, where p_x - prediction for the x percentile, t - target value.

Training and Evaluation Process

We used a train/validation split strategy based on years to prevent overfitting and make the model more robust. We also tried to use a repeated k-fold cross-validation strategy based on years [5], but we did not achieve significant improvements with it. In the final solution, we use a single model for each test cross-validation fold.

The data from the test year in each cross-validation split was removed from the dataset and not used during the train/evaluation process in any way.

Due to the small size of the training dataset we used an augmentation approach to synthetically increase the number of samples. For each sample we randomly generated the issue date in the range from the 1st of January to the end of the current season. Also we additionally randomly dropped the data from daily streamflows and SNOTEL during training.

The recipe of our training procedure for the single network is next.

We used a CAME optimizer with an initial learning rate of 0.0001 and default beta values and default weight decay [11].

But we also used the ReduceLROnPlateau scheduler with a patience period of 3 epochs and a reduced factor 0.1, which means that we reduce the learning rate ten times after 4 epochs without improvements in the validation loss.

All the training lasted for a maximum of 120 epochs, but additionally stopped after 12 epochs without validation loss improvements.

After the training, we saved the epoch checkpoint with the best validation loss as the training result.

All the hyperparameters were just slightly tuned by the average validation loss on the cross-validation.

Discussion of Performance

Figure 4 shows the average score for each year during the cross-validation period. Most years have approx. score close to 80 or lower, except for years 2011, 2012, 2006, and 2015. The maximum peak in 2012 is mostly related to the anomaly high score in the “libby_reservoir_inflow” site for this year.

We hypothesize that our model struggles with the lack of high-quality rain data and in the case of significant rains influencing the water supply the model underestimates the values.

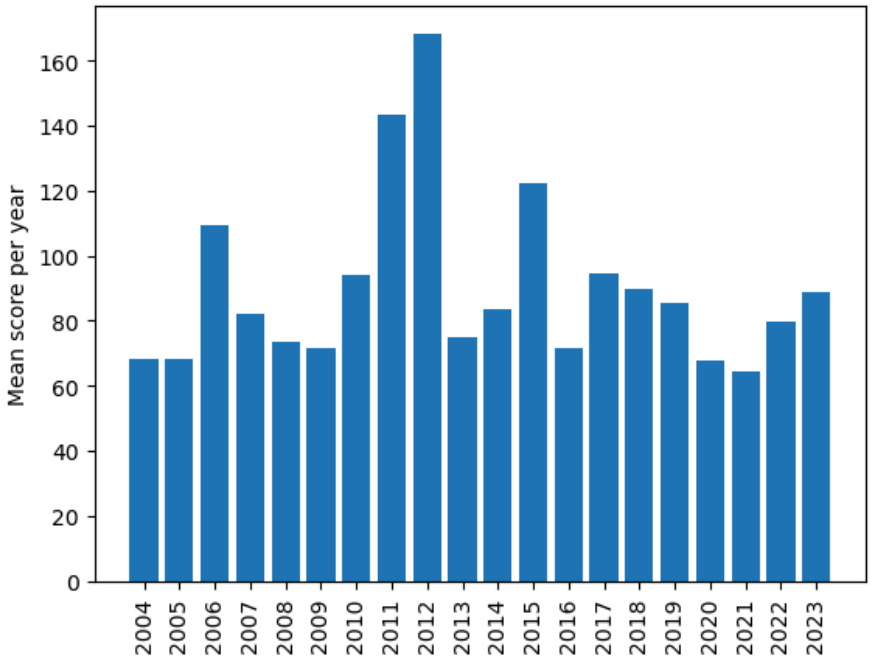

Fig. 4 Mean score for all sites per year

Figure 5 shows the average score for each site during the cross-validation period.

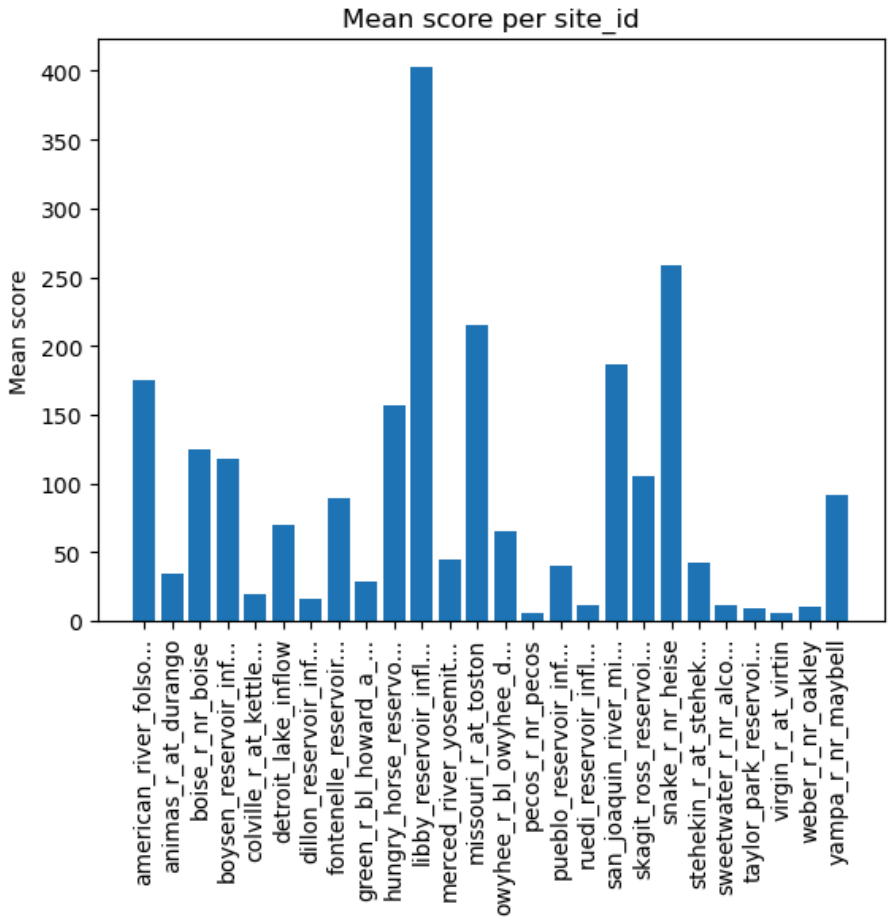

Fig. 5 Mean score per site for cross-validation years

Figure 6 shows the average score depending on the number of weeks from starting of the water season. There is a clear trend of improving prediction with the later predictions, which is expected due to the better input data and smaller residual to predict.

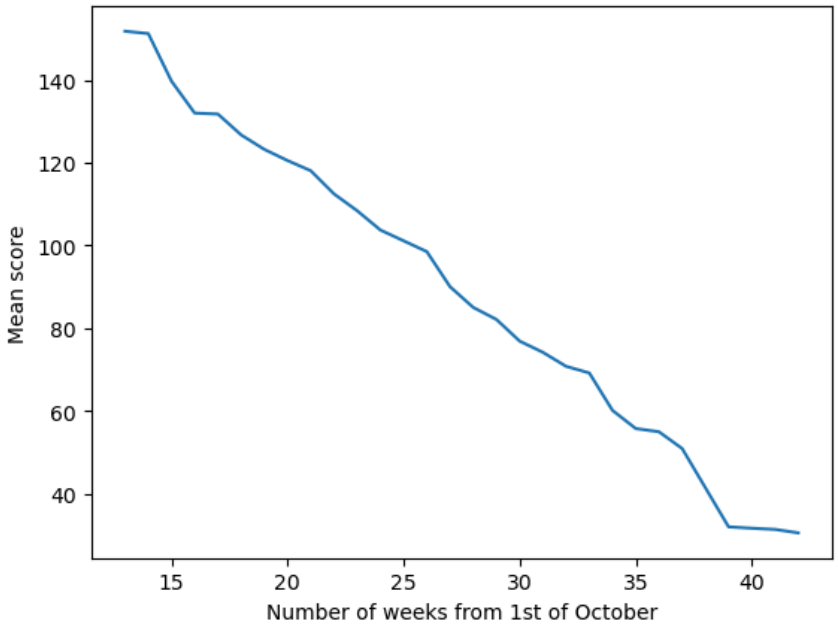

Fig. 6 The distribution of cross-validation scores depends on elapsed weeks from the start of the water season

We don't have any direct information about dry/wet years in the dataset. So, to estimate the score depending on the weather condition and amount of cumulative streamflow volume we normalized the streamflow value to the median volume for each site independently. Also we moved to normalized scores for each site/year pair to avoid the influence of different absolute values for different sites. The relationship of the normalized score depends on the normalized streamflow shown in figure 7. It's possible to see that for low normalized volumes the relative errors are much higher.

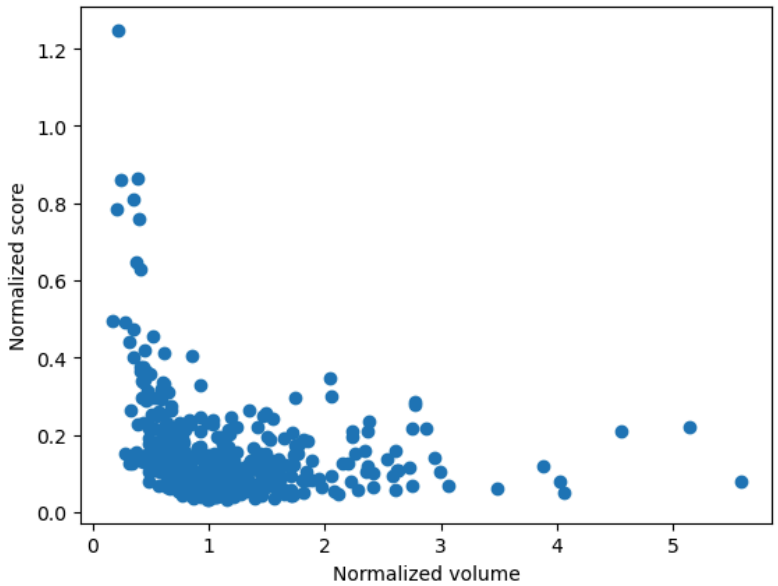

Fig. 7 The distribution of normalized scores depends on normalized target flow values

Figure 8 shows the average score related to the geographic position. The site “libby_reservoir_inflow” has the highest score across all sites. It can be related to its geographical position, because the basin is mostly placed in Canada and we don’t have SNOTEL-related data from Canadian territory.

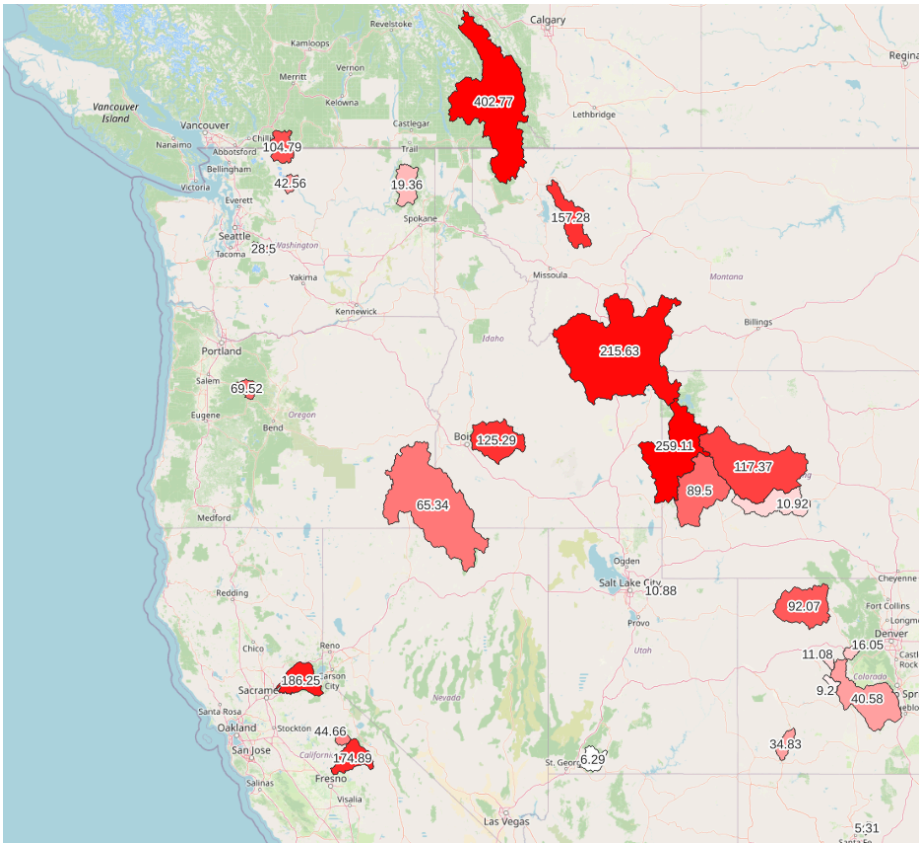

Fig. 8 The Mean score for each site depends on the geographical location

The most significant conditions for accurate predictions are the availability of monthly naturalized volume values and good SNOTEL stations coverage of the basin.

Because we don't use weather forecast data in the proposed approach, the anomaly amount of rain precipitation during spring/summer that is not related to snowmelt can significantly decrease the accuracy of the streamflow forecast.

Changes Between Stages

Compared to the Hindcast and Forecast stages, we added USBR reservoir inflow values to the features. Also, we slightly changed the augmentation process during training. The structure of the neural network was changed a little too, but the overall idea remained the same. Also, we experimented with the new optimizer CAME which turns to be a little more stable and fast.

We also modified the neural network model optimization process, described in the training and evaluation process section.

It is worth noticing that we switched from the repeated k-fold validation strategy with model ensemble to the single train/validation data split which significantly reduced the total training time for all test years.

Machine Specifications

We used two different machines for training the model. The first machine has the following specifications:

- CPU: AMD Ryzen 7 5700G
- RAM: 32 GB
- OS: Ubuntu 23.10

The second machine has the following specifications:

- CPU: AMD Ryzen 5 2600G
- RAM: 26 GB
- OS: Ubuntu 22.04
- NVIDIA GeForce RTX 2080 (8Gb)

Due to using the lightweight model the GPU usage is not mandatory for the training and inference process.

The solution is implemented using the Python language and several open-source libraries (PyTorch, NumPy, pandas, scikit-learn, pytorch-lightning, tqdm, pytorch-optimizer)

References

1. Agarap A., Deep learning using rectified linear units (relu). arXiv preprint arXiv:180308375. 2018.
2. Harpold, Adrian & Sutcliffe, Kent & Clayton, Jordan & Goodbody, Angus & Vazquez, Shareily. (2016). Does Including Soil Moisture Observations Improve Operational Streamflow Forecasts in Snow-Dominated Watersheds?. JAWRA Journal of the American Water Resources Association. 53. 10.1111/1752-1688.12490.
3. Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R.R., Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, 2012.
4. Koenker R. Galton, Edgeworth, Frisch, and prospects for quantile regression in economics, 1998.
5. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence, 1995.
6. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D., Back-propagation applied to handwritten zip code recognition. Neural Computation, 1989, 1(4):541–551.
7. Rumelhart D., Hinton G.E., Williams R.J., Learning Internal Representations by Error Propagation, Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986.
8. Srivastava N., Hinton G.E., Krizhevsky A., Sutskever I., Salakhutdinov R., Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014), 1929–1958.
9. U.S. Department of the Interior | U.S. Geological Survey. Title: USGS Water Data for the Nation Help. URL: https://help.waterdata.usgs.gov/
10. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I., Attention is All you Need, Advances in Neural Information Processing Systems 30, 2017.
11. Yang Luo et al. CAME: Confidence-guided Adaptive Memory Efficient Optimization. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023.