Lessons in Machine Learning from an Ungauged Basins Study

**Ellie White1, Jay Lund2, Andrew Hill3, Nicole Williamson3, Matthew Holland3, Erin Foresman3, Reza Ghasemizadeh3 (order OK?, affiliations?)**

1 EPA ORISE, Cincinnati OH

2 Center for Watershed Sciences, University of California, Davis

3 California State Water Resources Control Board

# Abstract

In recent years, rapidly growing computing power, improvements in machine learning (ML) algorithms, and more user-friendly programming languages have facilitated the application of statistical learning methods to problems in geosciences and hydrology. This paper first addresses the concerns that many hydrologists have regarding the application of purely statistical techniques to the prediction of physical processes. We also introduce the types of questions that need to be answered before building ML models–such as those relating to model types, loss functions, and resampling strategies. Then, we help answer them in the context of the problem of Predicting Ungauged Basin (PUB) flows. To be precise, this study estimates “unimpaired flows”, for 19 basins in California, in the 2000-2015 years, at a daily time step, using four model types: Linear Multivariate Regression Models (LM), Generalized Linear Models (GLM), Random Forests (RF), and Neural Networks (NN). We hope the experiences and discussions presented help usher in better ML models in hydrology.

# 1 Introduction

Runoff formation processes are highly variable, non-linear, and spatially heterogeneous, which make predicting processes such as streamflow challenging (Dooge, 1986). Hydrologic models that attempt to do so can be classified as causal/process based (PB) or statistical/empirical (Guisan & Zimmermann, 2000) (Figure 1). PB models have been increasing in complexity over time and often lack realistic parameter estimates. Moreover, these models require considerable field input data collection and calibration to obtain basin-specific parameters (Singh & Frevert, 2005). Unfortunately, as mechanistic models increase in complexity, it is unclear if hydrologic predictions improve commensurately (Beven, 2011).

In 2003, the International Association of Hydrological Sciences (IAHS) dubbed the 2003-2012 years the decade on Predictions in Ungauged Basins (PUB) (Sivapalan et al., 2003). The PUB initiative aimed the scientific community, in a coordinated manner, towards achieving major advances in hydrologic predictions for ungauged basins–advances that have yet to come. But why?

Lack of data and uniqueness-of-place *may* be why hydrologist have failed to come to scale-relevant theories of watersheds and in turn build accurate predictive models. However, it is also likely that the failure is due to the type of tools hydrologists have been using–the PB models themselves (Nearing, 2020).

Alongside the field of hydrology, statistics has also been grappling with its own growing pains. In 2001, Brieman argued that two cultures exist in statistical modeling: one where the model form is assumed (parametric modeling as in linear regression) and one where the model form is found (non-parametric modeling as in neural networks). He argued that the statistical community’s commitment to the almost exclusive use of parametric modeling has led to *“irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.”* In the past two decades, the field has moved past this hurdle evidenced by successful ML applications in computer vision, speech recognition, and bioinformatics.

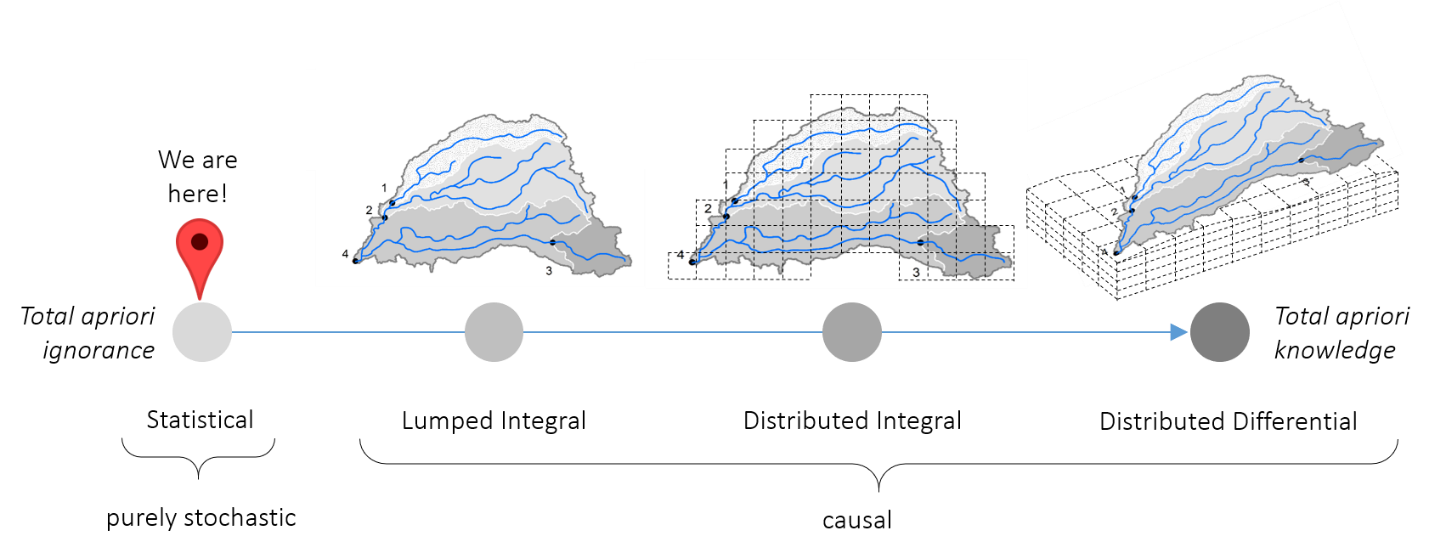


Figure 1. The different classes of hydrologic models. The hydrologic modeling field has been moving from total a priori ignorance to total a priori knowledge of the system. With the increase in computing power and the development of statistical learning methods, hydrologist can now re-visit predicting hydrologic conditions with purely stochastic methods.

Given the recent advances in statistical learning, we now have a good opportunity to improve the tools used in hydrologic modeling. The advantages are that given input data, which is just as challenging to gather for PB models, ML models are quick to build, appropriate for real-time or close to real-time updates, and require no babysitting or calibrating. Moreover, their predictive accuracy has surpassed PB models; first example of such an event was Hsu et al. (1995) and more recently Kratzert et. al. (2019) where ML models built for the ungauged basin outperformed PB models calibrated to specific basins. Quite the accomplishment.

# Concerns with ML modeling

Many hydrologists are skeptical of statistical modeling. Klemes (1982) warns modelers of the general limitations of empirical modeling: being in danger of overfitting, lacking justification outside the range of data supplied to the model, issues with non-stationarity, complexity in model structure, and lastly the arbitrariness of the selection of the form of models. Although, I will not dispute all these criticisms here, I will say that as I see it advances in statistical learning had attempted to remedy most of these problems, except for two: (1) models still perform poorly outside the training data. **Generalizing**, or predicting out-of-sample, is difficult for models that lack any a priori knowledge of the processes they are modeling; (2) models are still not as suited for **inference,** as in *understanding* the processes; it is still difficult to interrogate a complex non-parametric model to find meaning in its trained structure. The field is now beginning to make advances in both these areas.

One main concern with ML modeling is that there are no “physics” modelled. So, how can we make sure at least mass balance is obeyed? In ML modeling, the physics should first be embodied in the training data and can also be imposed as constraints. So, if mass balance is of paramount concern, either more data that obey physical laws can be added to the observed data (as in **data augmentation** methods) or **constraints** can be added to the model architecture or as a penalty in the loss functions. These constraints train a model that predicts values within user-specified bounds; a lower bound of 0 and upper bound of a crude estimate of the response (like precipitation\*drainage area) can force the model to give more realistic estimates of flow. Constraints also have the added benefit of imposing realities on the model that imperfect data may fail to do. For example, the unimpaired flows in this study were constructed from unavoidably imperfect accounting of water and therefore, at times, had negative values. To keep the model architecture simple, those values were omitted from the data, but there could have been valuable information in those values nonetheless. A model with constraints would not need to omit those values prior to training. ~~However, adding constraints will most likely degrade model quality as measured by the loss function.~~

All in all, when there is a choice between PB and ML models, the PB models are still recommended when extrapolating out-of-sample (e.g., climate change studies) and ML models are recommended when learning a complex and poorly understood relationship. It gets difficult when you have both a complex and poorly understood process and want to extrapolate out-of-sample. The two models represent two points on a pareto front-a tradeoff between **predictive performance** (what ML models do well) and **functional performance** (what PB models do well). Predictive performance is how well the predictions match the observations, and functional performance is how well the model variables relate to one another. But is there a way to have the best of both the ML and PB worlds?

Hybrid models give us more points on the pareto front. Hybrid models can either: (1) use an ML method to solve the Partial Differential Equations (PDE) in hydrology. This is ML coming to the aid of a PB model; or (2) use simulated PB model data to *augment* ML training observations. Here, the ML learns the process from observation *and* data that obey the PB model physics; (3) use boosting techniques where a PB model learns the process and the ML model learns from the errors of the first model. Points (2) and (3) are explained more in the discussion section as improvement strategies.

# 2 Data

The uniqueness of each basin begs the question of how much data is needed for a model to learn in a sparce predictive space. Usually, sparsity in ML is induced by high dimensionality, but here it is induced by *static* basin characteristics. For example, in this example problem, the model must learn how unimpaired flow changes with drainage area given only 19 different instances of the variable, since that variable is static. Therefore, in this learning space, diversity in basins would provide more valuable information than a longer record at a basin the model has already seen. This was reflected in the drastic decrease in training error when the basins in our study grew from 9 to 19.

A nice incidental feature of neural networks is that they learn a lot from few parameters. Therefore, amassing many predictor variables is not necessary. These models benefit more from being deeper with more internal nodes and hence requiring more computing resources. This shows that the information we want to extract from data is there in the data we have already collected, and merely our “tools” have been lacking. In short, the answer to the question of how much data do we need to build a good ML model can only be answered when the study it taken to its conclusion- as in, the model is built, tested, and benchmarked.

Data for our analysis came from the California Data Exchange Center (CDEC) daily unimpaired flows. Unimpaired flow is the flow produced by the basin in its current state, but without human created or operated water storage, diversion, or return flows (California Department of Water Resources, Bay-Delta Office, 2016). It is constructed from a simple accounting of water. The data spans 19 basins from 01/01/2000 to 01/01/2015 at a daily timestep for a total of 102,334 observations in cubic feet per second (CFS) (Figure X). These basins were chosen because of the availability of daily constructed unimpaired flows (cite CDEC). Other static basin characteristics were assembled from various publicly available sources (Table X).

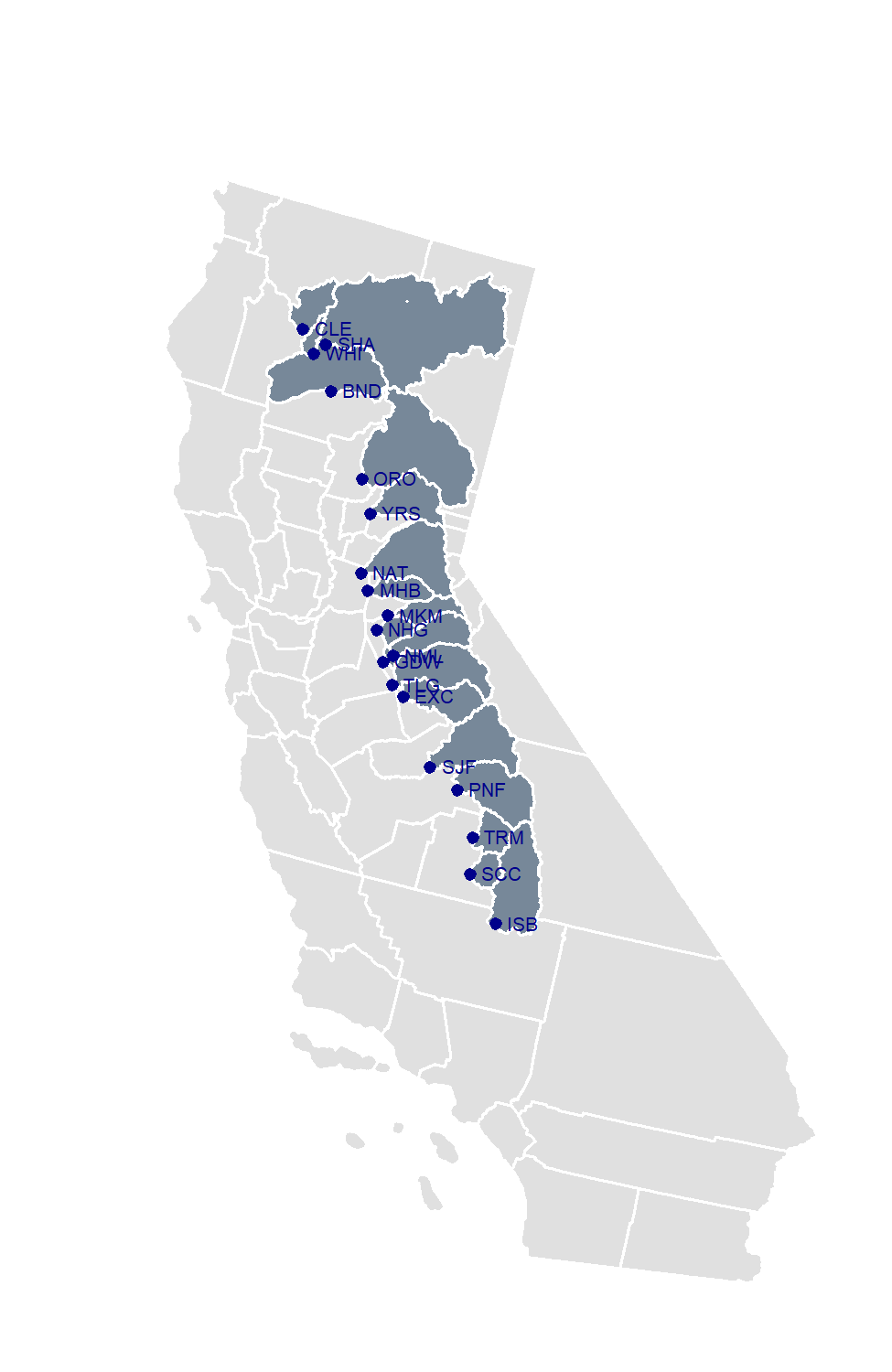


Figure 2. The 19 California basins under study are the California Data Exchange Center (CDEC) unimpaired flow basins. Gauge data are “unimpaired” by adding back in diversions, subtracting imports, and accounting for change in storage and evaporation caused by reservoirs.

Table 1. Summary of the variables used in the implementation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Description | Source |
| Response | Unimpaired Flows | monthly estimated unimpaired flows, in *CFS* | CDEC (Beaudette, 2016) |
| Time | Ordinal Month  Water Year | numerical distance till October  numeric year starting from the October of previous Gregorian year | - |
| Climate | Temperature, Lag 1, 2, and 3 days  Precipitation  Lag 1, 2, and 3 days  Snow | temperature and lagged daily temperature, in ℃  precipitation and lagged daily precipitation, in *mm*  cumulative precipitation of the same water year for temperatures below 2 *℃,* in *mm* | PRISM (Hart & Bell, 2015) |
| Hypsometric | Relief Ratio  Mean Elevation | (max(elev) - min(elev))/ basin length in, *m/m*  mean basin elevation, in *m* | SRTM90 (Jarvis, Reuter, Nelson, Guevara, et al., 2008) |
| Basin Boundaries | Area  Shape  Compactness | basin drainage area, in *miles2*  basin length/basin width, in *m/m*  basin area/(basin perimeter)2, in *m2/m2* | NHD2PLUS (McKay et al., 2012) |
| Soil | % Clay  % Silt  % Sand  Saturated Hydraulic Conductivity  Lambda  N  Available Water Content | percent clay in surface layer, in *%*  percent silt in surface layer, in *%*  percent sand in surface layer, in *%*  hydrologic conductivity of surface layer, in *cm/hr*  pore size distribution index (brooks-corey)  measure of the pore size distribution (van genuchten)  available water content, in m^3/m^3 | POLARIS (Chaney et al., 2016) |
| Land Cover | Vegetated | Percent of area in the basin vegetated in *%* | CALVEG  (Forest Service, USDA, Pacific South-west Region,  2006) |
| Groundwater | Depth to Restricted Layer | depth to aquitard, in *cm* | POLARIS (Chaney et al., 2016) |

# 3 Methods

As mentioned before, a suitable ML method has (1) good predictive performance or “gets the right answers”, and (2) good functional performance or “gets the right answers for the right reasons.” So, the first task is to train the models with parameters that give good model measures of fit, then look at the way the parameters relate to one another in the best model. We do this by plotting the variable importance and partial dependence.

Schmidt (2020), recommends using many different modeling methods to eliminate the bias we as researchers introduce when picking a particular model to train. In this study, we used four model types: Linear Multivariate Regression Models (LM), Generalized Linear Models (GLM), Random Forests (RF), and Neural Networks (NN) (Table1). Currently, LSTMs, a special case of Recurrent Neural Networks, are the state-of-the-art for PUB. Since we never optimized the LSTM parameters, we have excluded them from our analysis, but have discussed their merits and initial lessons learned from training a handful of LSTMs in the discussion section.

Table 2. A review of ML methods and their conceptual basis, weaknesses, and strengths.

|  |  |  |  |
| --- | --- | --- | --- |
| ML Methods | Conceptual Basis | Weakness | Strengths |
| Linear Multivariate Regression (LM) and variants like Kriging | random:  Y ~ N (µ; σ2)  systematic:  µ = Xβ | Assumes the relationship between the inputs is *linear* and *additive*, and that the errors are from a *Normal* or Gaussian distribution.  Difficulties when extrapolating beyond the range in the data (e.g., negative flow predictions) | Simple to understand.  Given standardized data the relative effects of the inputs can be discerned from the trained weights. |
| Generalized Multi-variate Regression (GLM)  with compound exponential dispersion distributions (e.g., Tweedie) | random:  Y ~ P (µ; θ)  systematic:  **g**(µ) = Xβ | Assumes the relationship between the inputs is linear and additive, and that the errors are from a *known* distribution other than Normal.  Difficulties when extrapolating beyond the range in the data (e.g., large flood predictions) | Gives better fit than LMs.  Negative flow predictions are no longer an issue because an appropriate link function adds that information to the model. |
| Tree based methods and variants like Random Forests (RF) | Recursive binary splitting of the predictor space | Difficulties predicting out-of-sample.  *Averaging* in Random Forests dampens the prediction densities compared to observations. In other words, a simple RF will never predict a flood higher than it has already seen. | Automatically recognizes lower and upper bounds should they exist in the data (e.g., stream flows are never negative). |
| Neural Networks (NN) and variants like Long Short-Term Memory (LSTM) networks | Multiple sequential non-linear transformations of inputs to outputs | Larger/deeper networks require good processing capabilities.  Hard to *interpret* or understand the weights inside the network. | Gives better fits to data in sample and out-of-sample.  Can recognize constraints with regularized penalty functions or the addition of a constraint layer. |
| Support Vector Machines (SVM) | Maximal margin classifiers (modified for regression) | Peculiar behavior where *lighter* rainfall would generate unrealistic hydrographs.  Very new in regression applications | Performance is coming to be on par with NNs. |

Before training models, there are two fundamental questions that need to be answered: (1) what should the loss function be? Typical loss functions in hydrology are symmetric meaning they weight overestimation the same as underestimation of the same magnitude. In this study we show results for a range of symmetric and asymmetric loss functions that fit the different parts of the flow distribution differently (Table 3); (2) given the loss function, how well is the model doing? In the absence of a test set, one needs to be constructed by following one rule: *resample like sample*. The most appropriate resampling technique for PUB is one where one or more basins are held out to be the test set. The procedure is then repeated so each basin has a chance of being in the test set (Table 4).

Table 3. Loss functions

|  |  |  |
| --- | --- | --- |
| Type | Loss Functions | General Performance |
| Symmetric | Mean Squared Error (MSE) | Regresses to the mean. |
| Symmetric | Mean Absolute Error (MAE) | Regresses to the median. |
| Symmetric | Log Hyperbolic Cosine (LOGCOSH) | In small errors acts like the MSE, and in large errors like the MAE. |
| Asymmetric | Weighted Least Squared Error (WLSE) | Fits the tails of the distribution at the expense of the center depending on specified weights. |
| Asymmetric | Linear Exponential Error (LINEXE) | Fits the tails of the distribution at the expense of the center depending on specified weights. |

Table 4. Resampling strategies

|  |  |  |
| --- | --- | --- |
| Type | Cross-Validation Resampling | Description |
| Random | Resubstitution (RESUB) \* | The test set is the training set. Here, the model is evaluated against the same data it has already seen. |
| Random | 2, 5, and 10 Fold (K FOLD) | The data is split into k folds and in each iteration one fold is held out as test set. The process is repeated for each fold. |
| Blocked | Leave One Group Out (LOGO) † | In each iteration, one basin's data is held out as a whole and the rest of the basins become the training set. The process is repeated for each basin. |
| Blocked Randomly | Leave Multiple Groups Out (LMGO) ‡ | In each iteration, 1/5th of the basins (3 or 4) are held out and the other basins become the training set. The process is repeated for each fold. |

\* least appropriate for any model because we may be over fitting.

† most appropriate for PUB.

‡ makes for a harder problem than PUB. Other resampling strategies like Leave Hierarchies Out can *systematically* block gauges on the same river system. Because, of the limitations in size of the data set we never attempted this.

# 4 Results: Predictive Performance

Model results presented here primarily show the bR2 statistic. The R2, coefficient of determination, is a single-parameter indicator of the discrepancy between observed and predicted values (Equation 1). However, R2 is insensitive to additive and proportional difference between observations and model predictions. One can simply show that for a non-zero value of and , if the predictions follow a linear form, , the R2 equals one (Legates & McCabe Jr, 1999).

Therefore, for a proper model assessment, it is recommended that the slope of the line (with 0 intercept) fitted on the predicted vs. observed graph be reported or systematically included as shown in Equation 2.

Equation 1

Equation 2

Given a large dataset, non-parametric methods like the RF and NN models outperform parametric ones like the LM and GLM because they make none, or at least fewer assumptions, about the underlying process. The results for the PUB problem, in the LOGO resampling strategy, show this with the bR2 values of x, y, z, d, for the models in order of increasing complexity: LM, GLM, RF, and NN. In all models, the test set error is lower for smaller block/fold sizes; in other words, random resampling methods (e.g., 10 fold) underestimate model error as compared to blocked methods (e.g., LOGO).

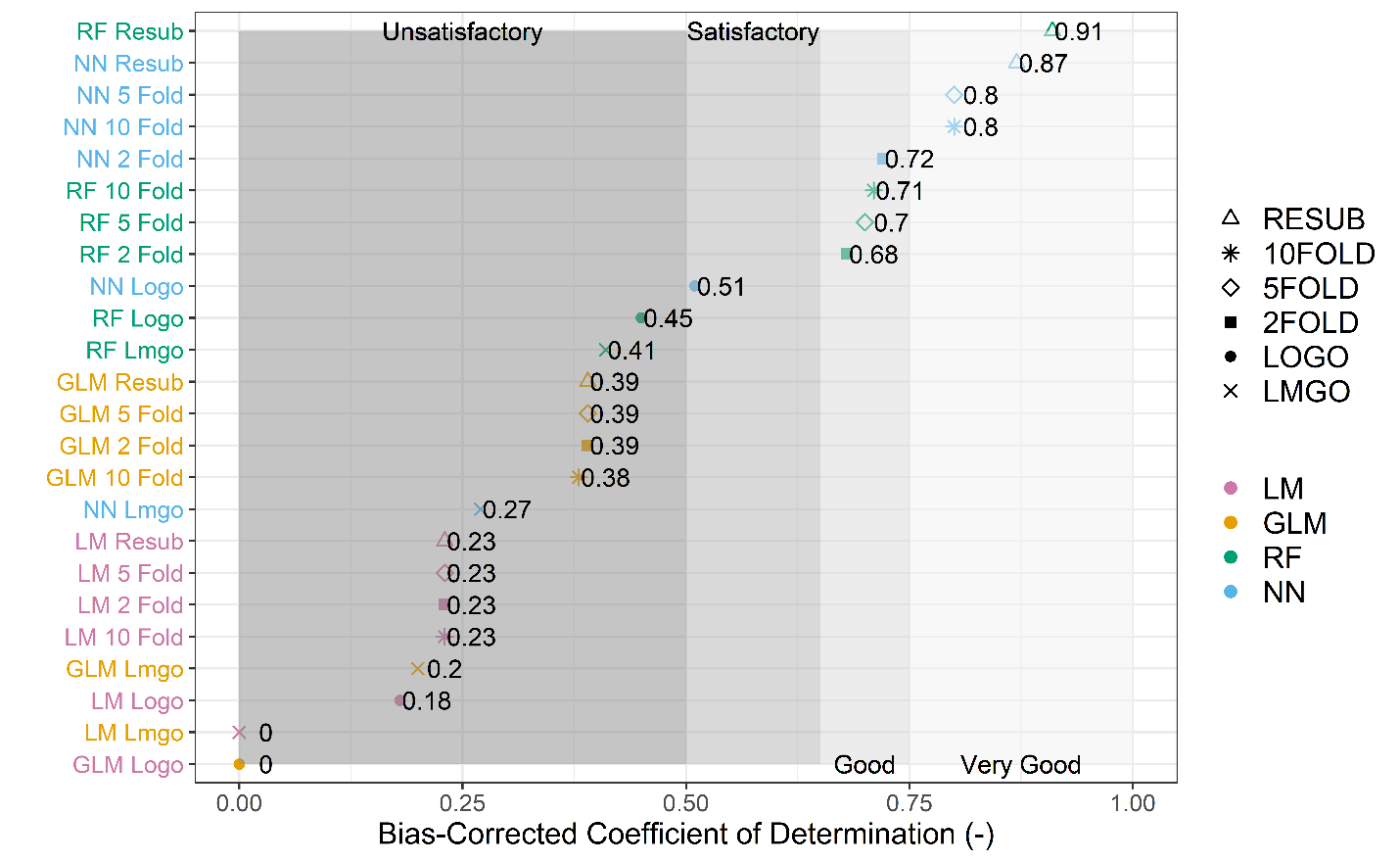


Figure 3. Predictive accuracy given by the bR2 for different resampling strategies in the test set. LOGO and LMGO give the more appropriate measures of accuracy for the PUB problem. The NN model outperforms other model types in LOGO resampling.

The predicted vs. observed plot confirms the goodness-of-fit results in the best model—the NN. The predictions of a perfect model would fall along the 1:1 line; however, in the LOGO resampling strategy, the NN model is slightly biased to underpredicting large values () and overpredicting small ones (). The plot also shows some potential *high leverage* points in the BND, NAT, and ORO basins that should be inspected for accuracy. Afterall, the flows are all constructed by an imperfect method.

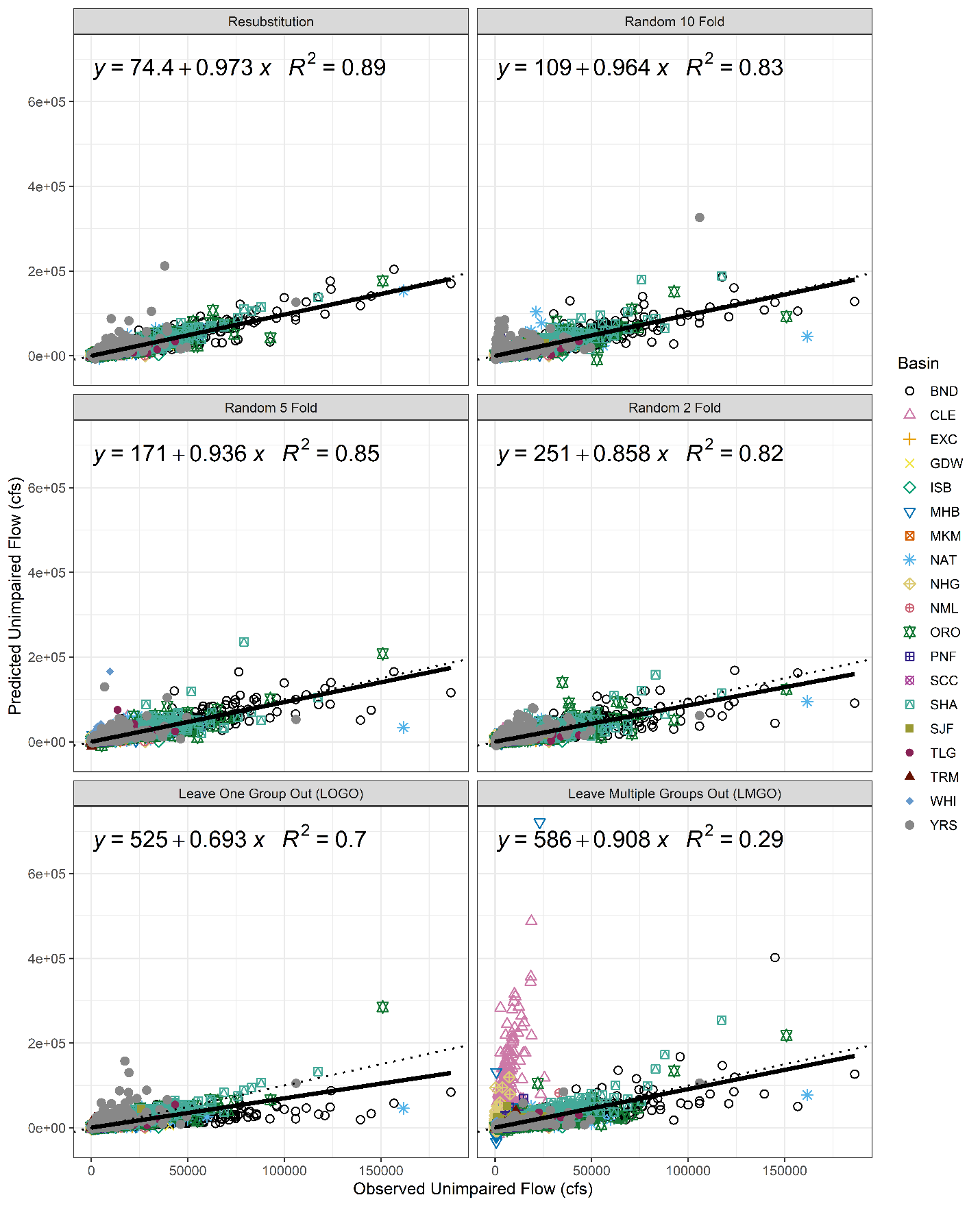


Figure 4. Predicted vs. observed for NN models trained on MSE loss with various resampling strategies. Points on the plot come from the test sets only. The LOGO method is the most appropriate resampling strategy for the PUB problem and reports a R2 of 0.7 with a small bias.

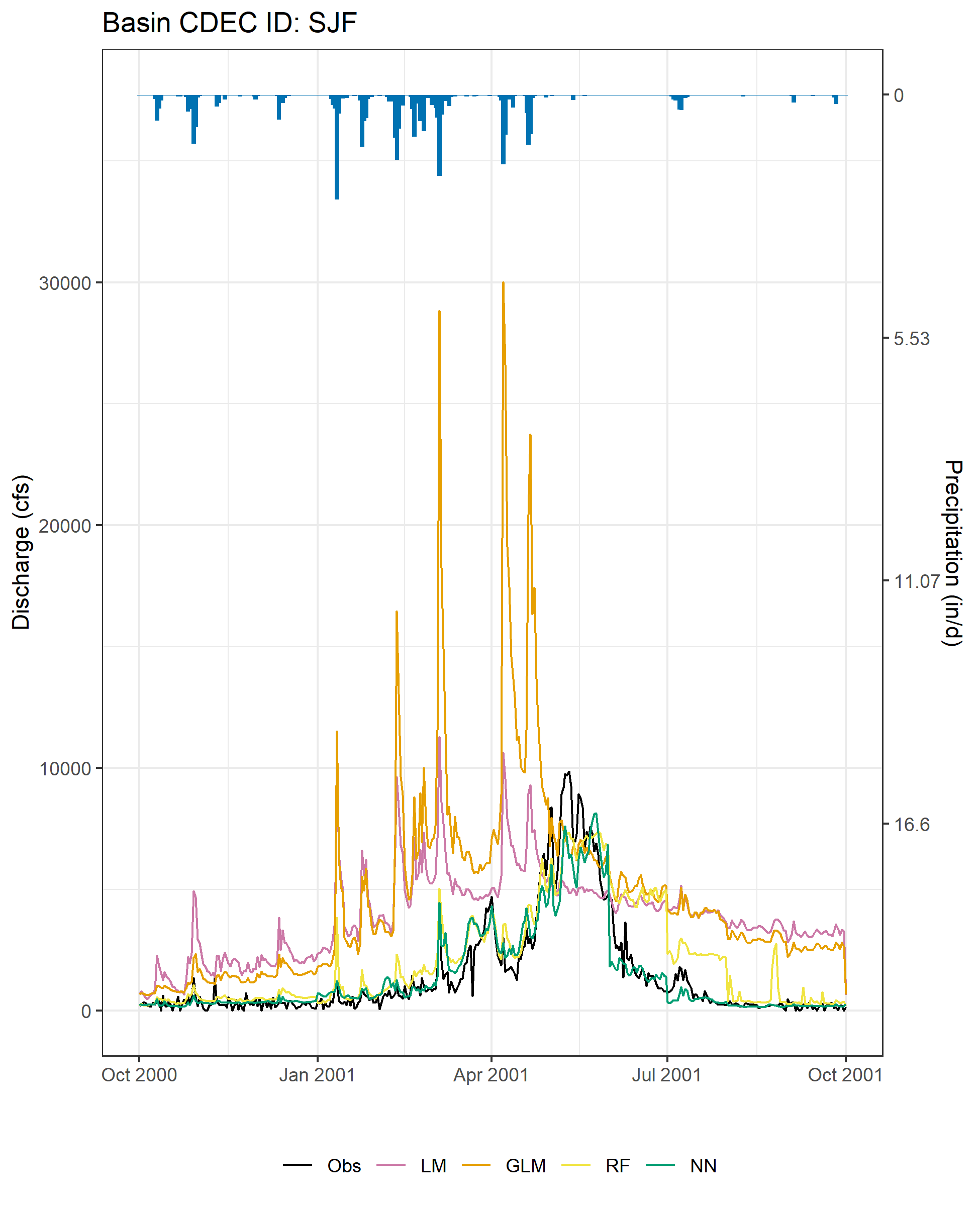


Figure 9. Time series results for the four model types zoomed in to first year for SJF.

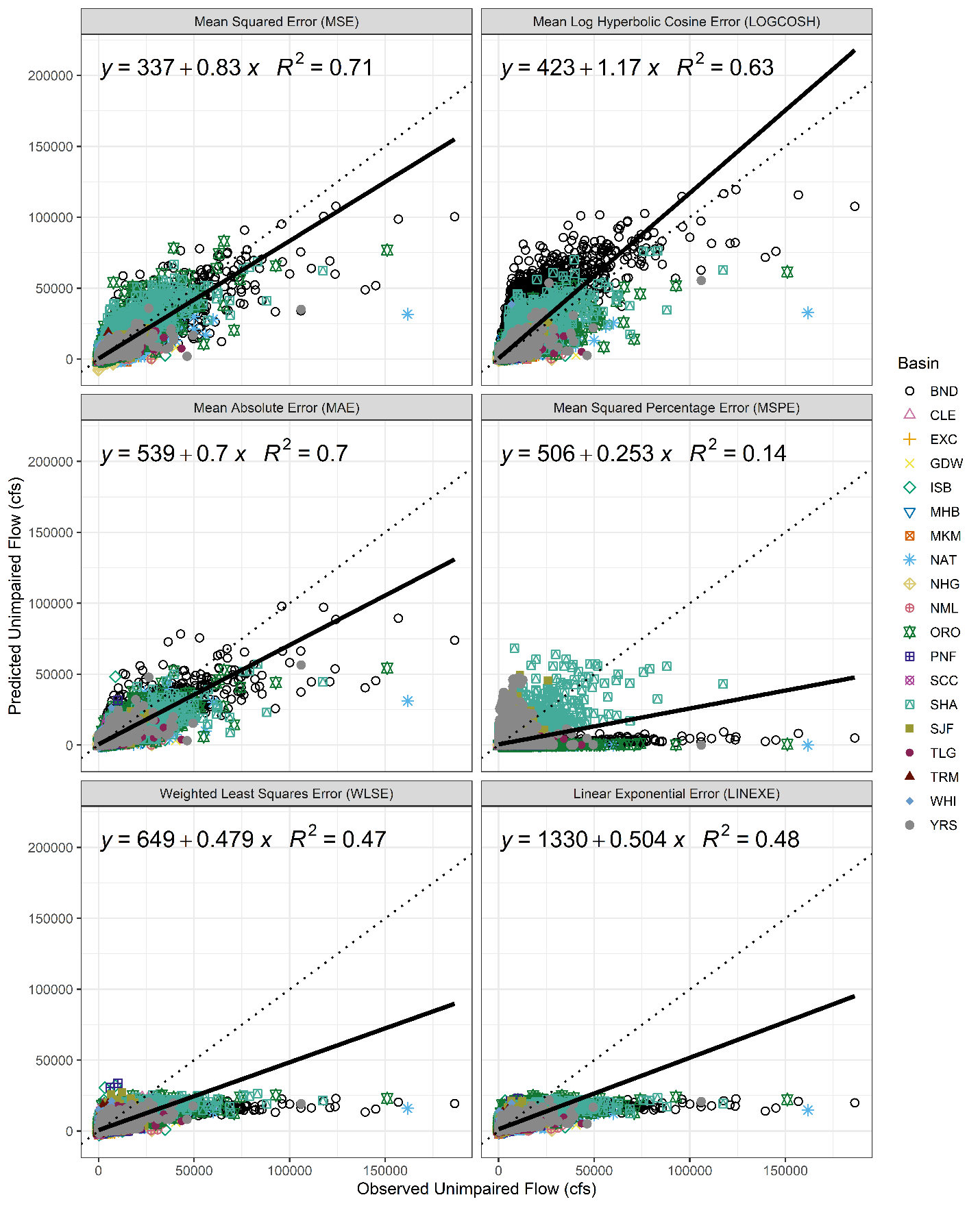


Figure 5. Predicted vs observed plot for NN models trained with the LOGO cross validation sampling with various loss functions. There is very little difference observed between the MSE, LOGCOSH and MAE methods. The MSPE greatly underpredicts observations. The WLSE and LINEXE loss functions generally underpredict the largest floods but overpredict lower flood values.

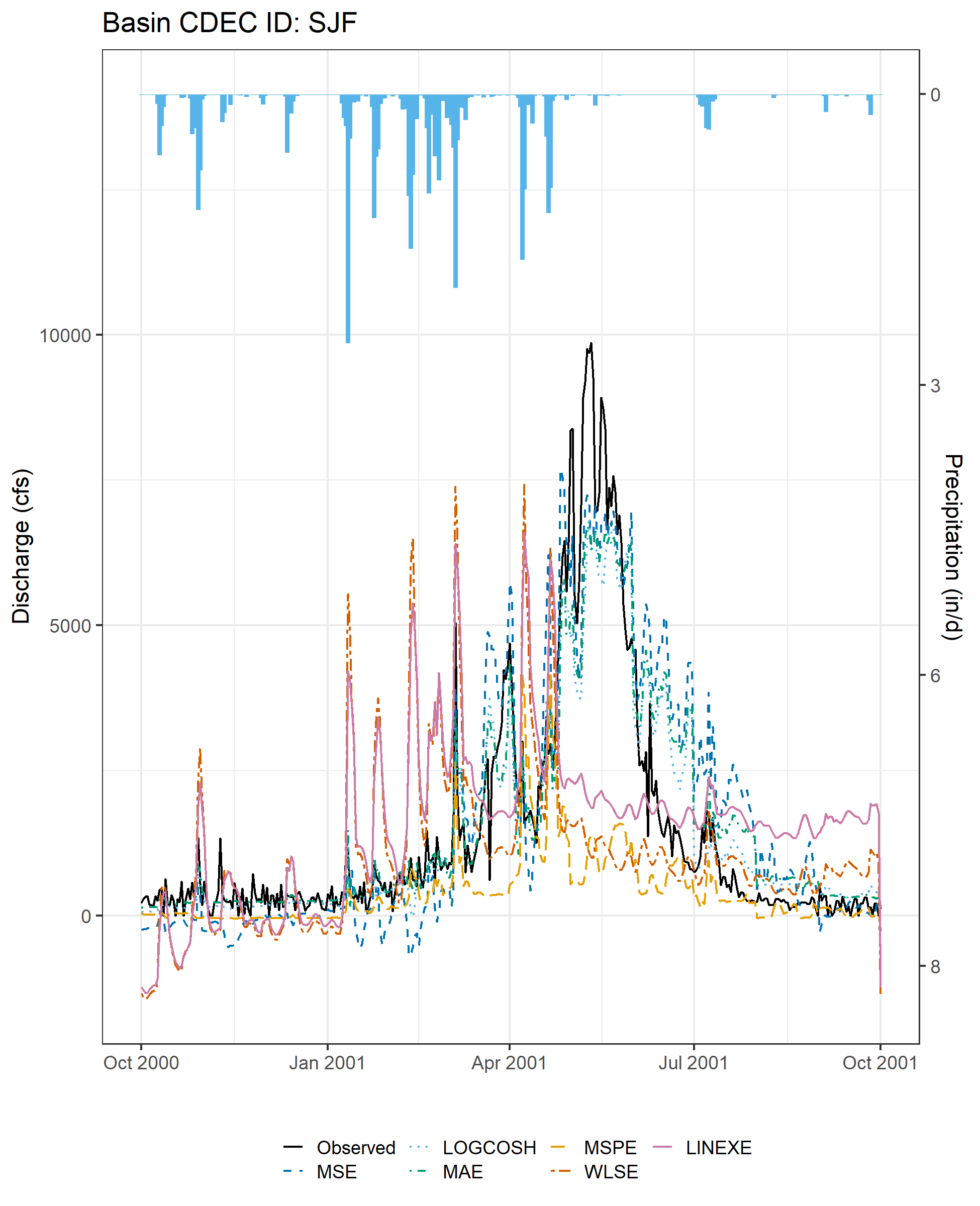


Figure 10. Time series results for the various loss functions for SJF zoomed into first year.

Results for the various loss functions show that the MSE, LOGCOSH and MAE (symmetric loss functions) try to fit to a measure of center (either mean, median, or somewhere in between). WLSE and LINEXE (asymmetric losses) try to fit the tails of the distribution to the degree dictated by its parameters. As expected, the MSE loss perform the best in the bR2, since both the MSE and bR2 metrics are calculating similar squared errors. The spatial distribution of the bR2 appears random, which otherwise would have indicated important variables missing from the model (Figure 6).

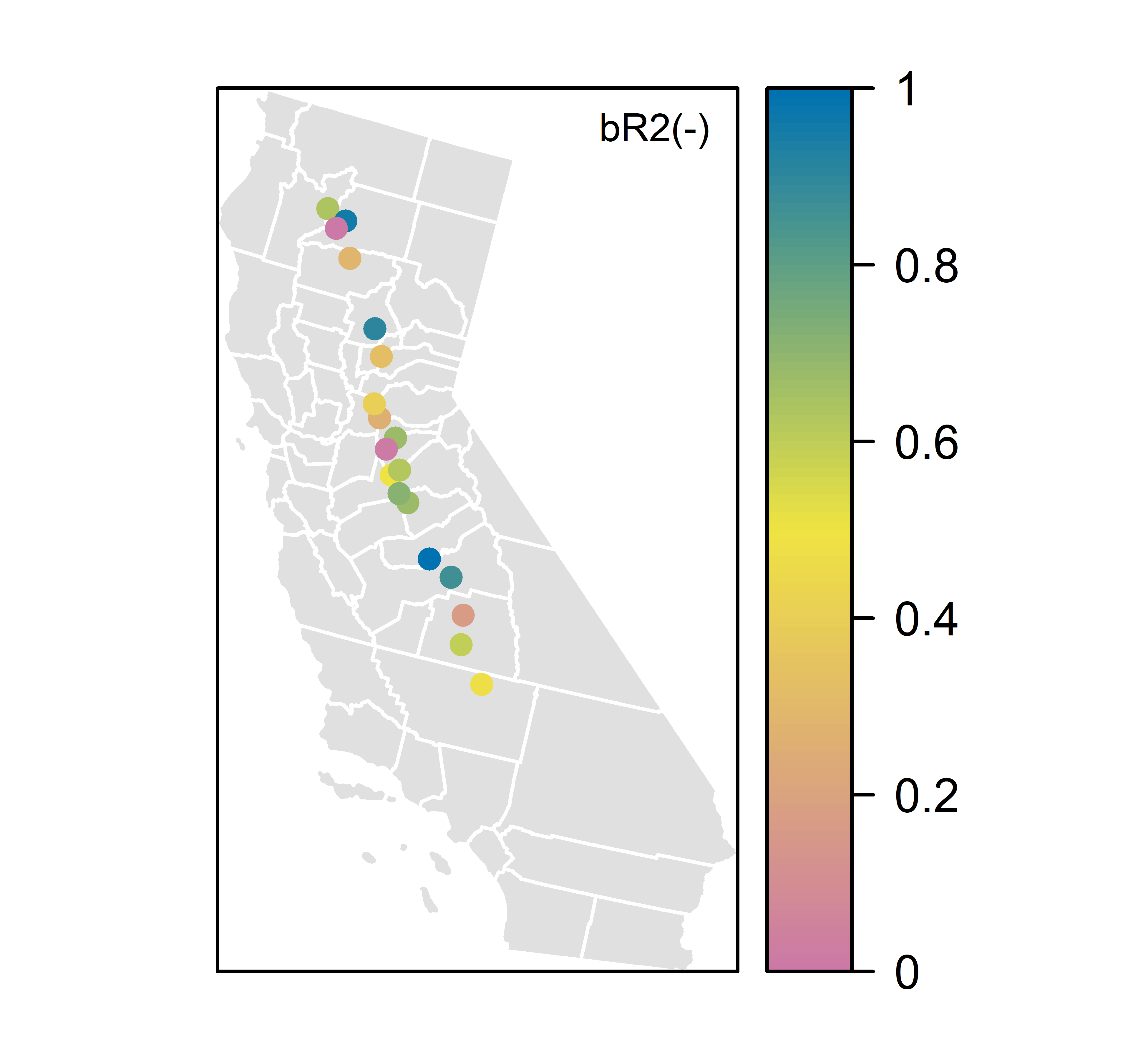


Figure 6. Spatial plot of bR2 for the NN model trained with MSE loss and LOGO resampling. Goodness-of-fit appears random.

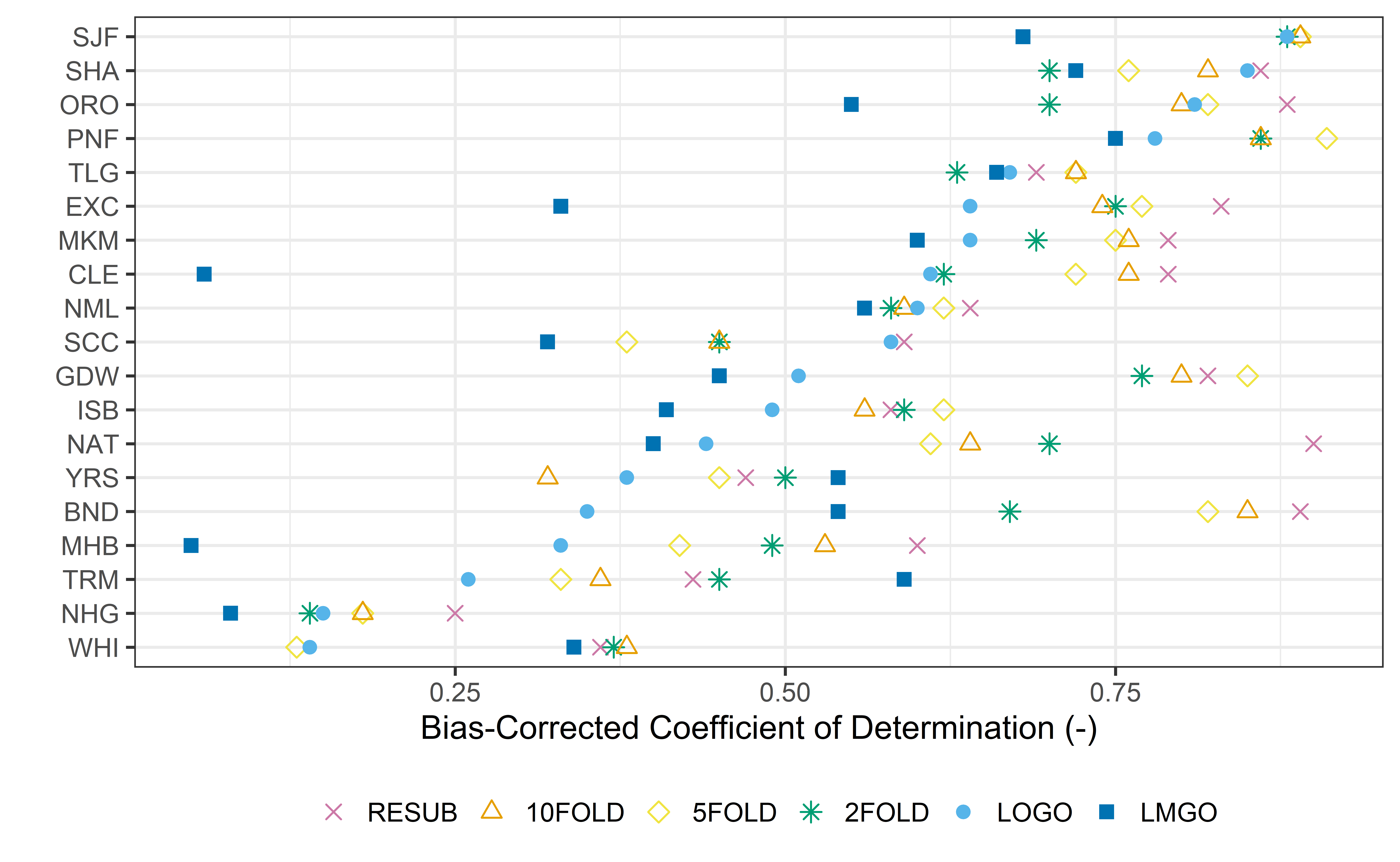


Figure 7. bR2 by basins for the NN model trained with MSE loss and LOGO resampling. Top 3 and bottom 3. discuss

# 4 Results: Functional Performance

SJF: SAN JOAQUIN RIVER BELOW FRIANT best performance, most accurate inferential results.

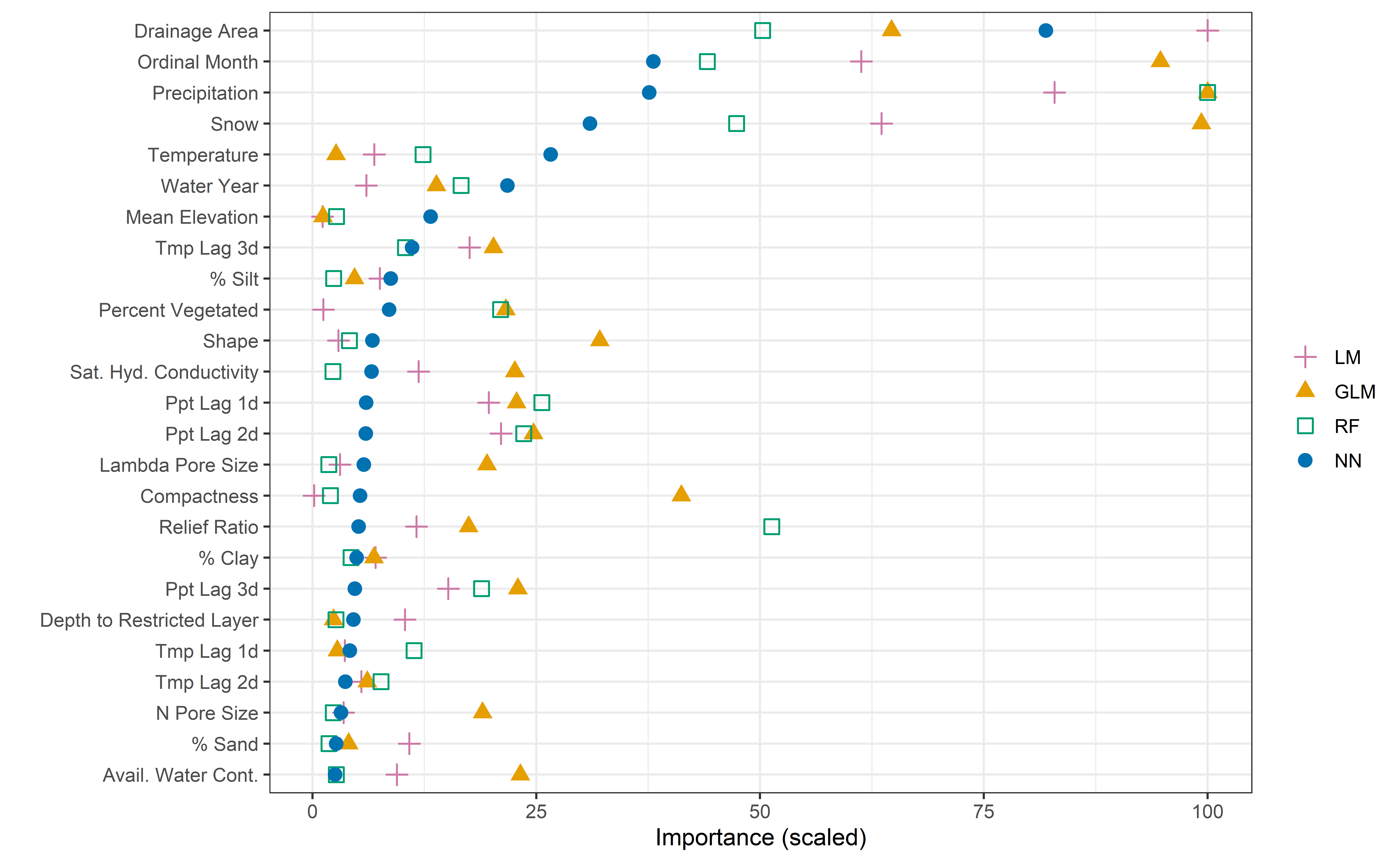


Figure 11. Variable importance plot for NN logo, SJF test set MSE loss

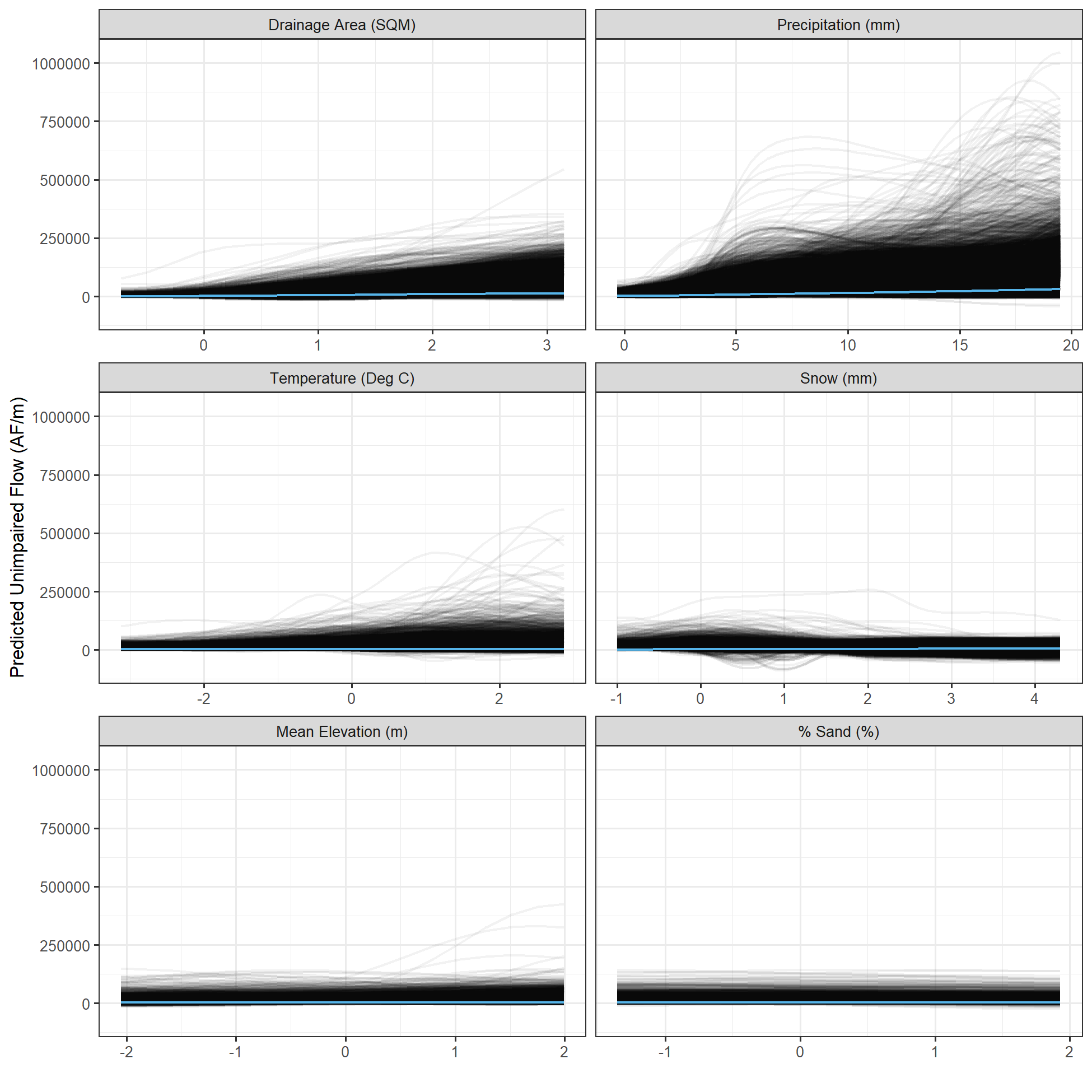


Figure 10. Partial Dependence plot for NN LOGO SJF test set MSE

# 5 Conclusions and Next Steps

Moriasi, we are in the satisfactory modeling place (predictive performance). Pretty good given what I know of PB models. To accurately benchmark we need to run a similar study with similar data with a PB model to compare. talk about functional performance.

Dropout did not help, maybe because the network is already small (the number of units). Future studies should try to optimize the network size, and systematically make it larger when adding dropout to see if it helps. Reduce\_lr\_on\_plateau didn’t help because with the ADAM optimizer the patience was never reached. Maybe this is a sign that longer training epochs would be helpful. Optimizer RMSProp showed large fluctuations especially on the validation data, making learning harder. Longer epochs here were needed compared to ADAM. ADAM was much better/smoother.

RELU RELU activations were adequate for initial runs and had the advantage of not needing standardized predictor variables. The NN with different losses were built using this activation because we needed to preserve the distribution information if we are to try to fit the different areas. Standardizing is effectively removing the center and spread information from data. So, it may not be a good idea anyway.

but in the various experiments, tanh-exponential with standardized predictor variables showed slight improvements. These activations need standardized variables. And standardizing will make for a better-conditioned problem, better initialization.

Batch\_size controls the number of gradients that are estimated before weights get updated. A full batch uses all observations to estimate the gradient and is most computationally expensive but gives the most accurate estimate of the gradient. We used different batch\_sizes to optimize for run time rather than accuracy because I was limited by computational ability. This could be another parameter that could be tweaked.

LSTM runs were made with a moving window. It was never set with optimized parameters so it wasn’t included in this report. It should be a easy next step. Ncells larger than 5-10 degrade quality. Larger units improved quality in more than half the basins.

# 6 Discussion

Q. What are the limitations of this study?

1. data quality [insert negative flow graph or discuss]
2. static variables
3. models aren’t properly optimized. Just a handful of experiments, using intuition on which and by how much to change parameters.
4. non-stationarity
5. out-of-sample or “extrapolation”

Q. How do we remedy each limitation?

1. check the data and the modeling of UF. Negative flows could be encoded differently with NNs so we don’t have to outright omit them from the study.
2. Data Augmentation with PB
3. Boosted learning with PB
4. Semi-supervised learning or filtering/weighting later parts of the record to emphasize time.
5. Decide where on the pareto-front you would like to be, for predictive performance go with ML, for functional performance, go with PB, for somewhere in the middle use a hybrid approach.

Insert entropy slide

# Acknowledgements

SWRCB

Link to Github

# References