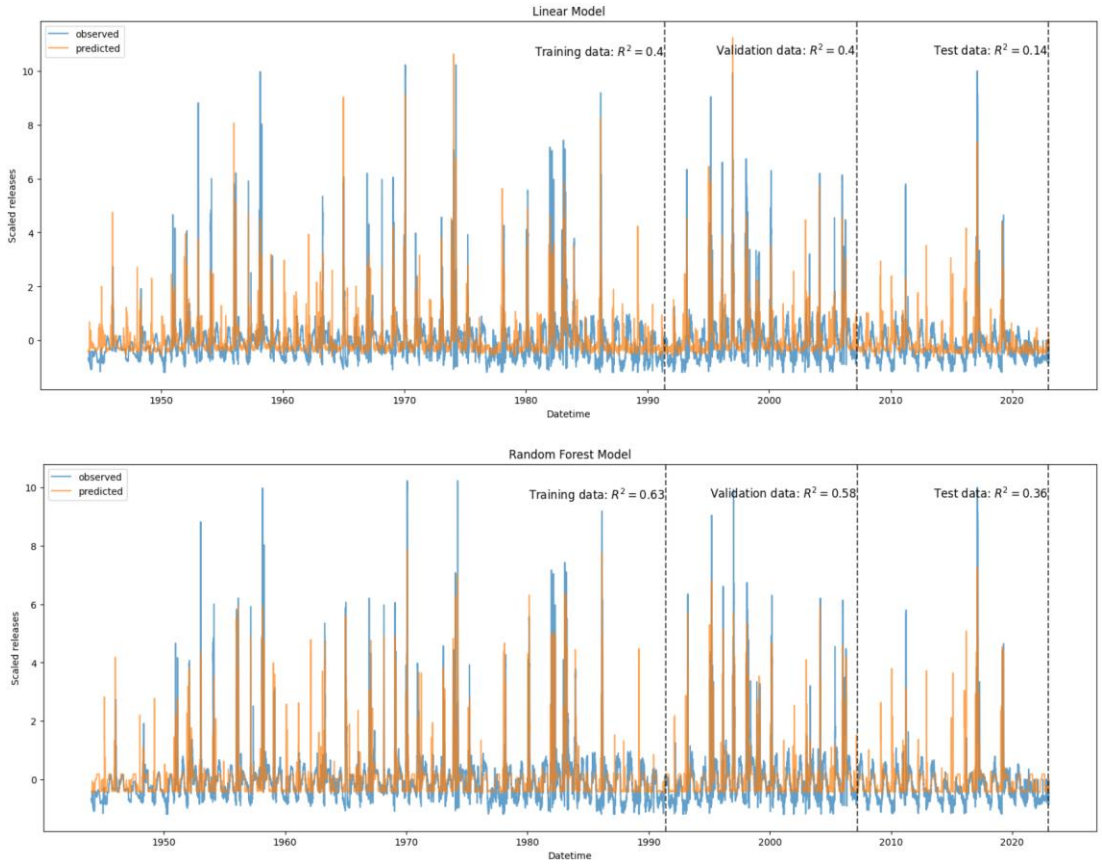
Update Notes 2/7/2024.

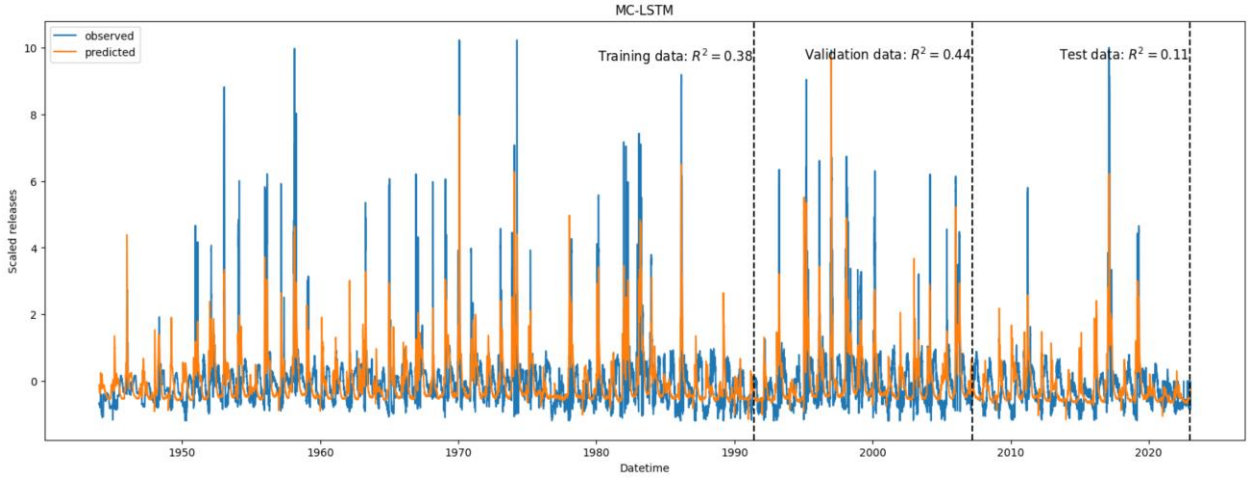
1 Benchmark models

Up until now, all of our work was focused on the LSTM model and some variants. To compare these models to some non-LSTM architectures, I trained a linear and random forest model on 5-autoregressive inflow lags for Shasta reservoir. These models perform notably worse than the LSTM models, and the problem of false peaks is more pronounced. If storage is provided, the R2 scores increase (0.42/0.43/0.21 for the linear model, 0.69/0.64/0.56 for the random forest) but the problem of false peaks persists.



2 MC-LSTM on Shasta Reservoir

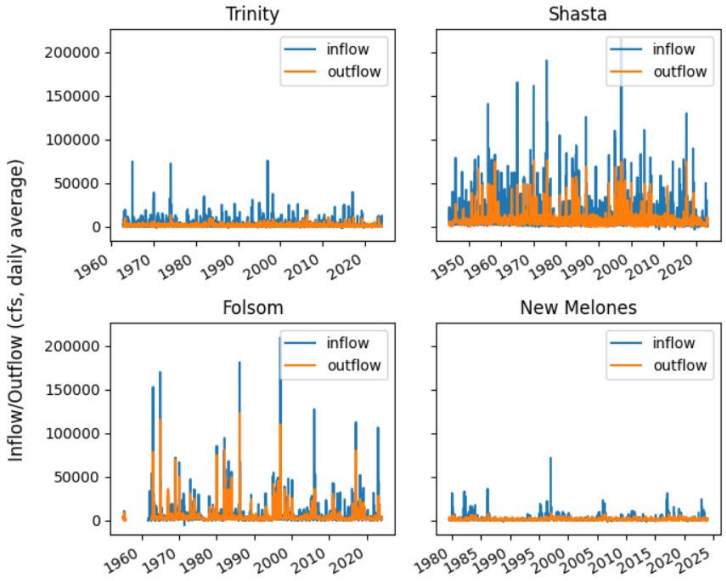
The MC-LSTM is a modified LSTM architecture that enforces mass conservation in its cell state; i.e. applied to reservoir release modeling, the cell states would represent a true storage state (Hoedt, 2021). After training on Shasta Reservoir, the results were surprisingly poor (see below). The model struggles to learn seasonal releases as well as the standard LSTM models, and the problem of false peaks persists. The correlation coefficient between true storage and MC-LSTM cell states were 0.335, 0.470, and 0.455 for the train, validation, and test portions, respectively.



3 Modeling several reservoirs

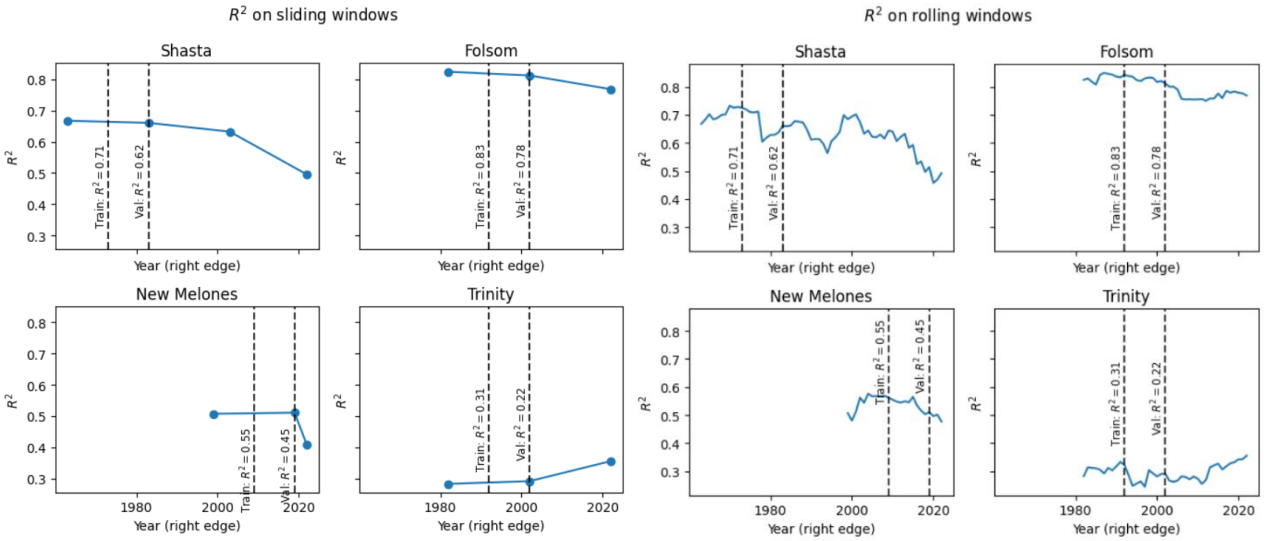
3.1 Operational differences between reservoirs

By plotting releases against inflow for Trinity, Shasta, Folsom, and New Melones reservoirs, it is clear that some reservoirs are more sensitive to inflow peaks than others. Folsom reservoir, for example, shows flood protection release peaks for almost every high inflow peak while Trinity and New Melones are less sensitive to inflow peaks and are more concentrated on seasonal releases. From a machine learning perspective, reservoirs that are more sensitive to inflows will be easier to model than those that are not.



3.2 Moving performance for other reservoirs

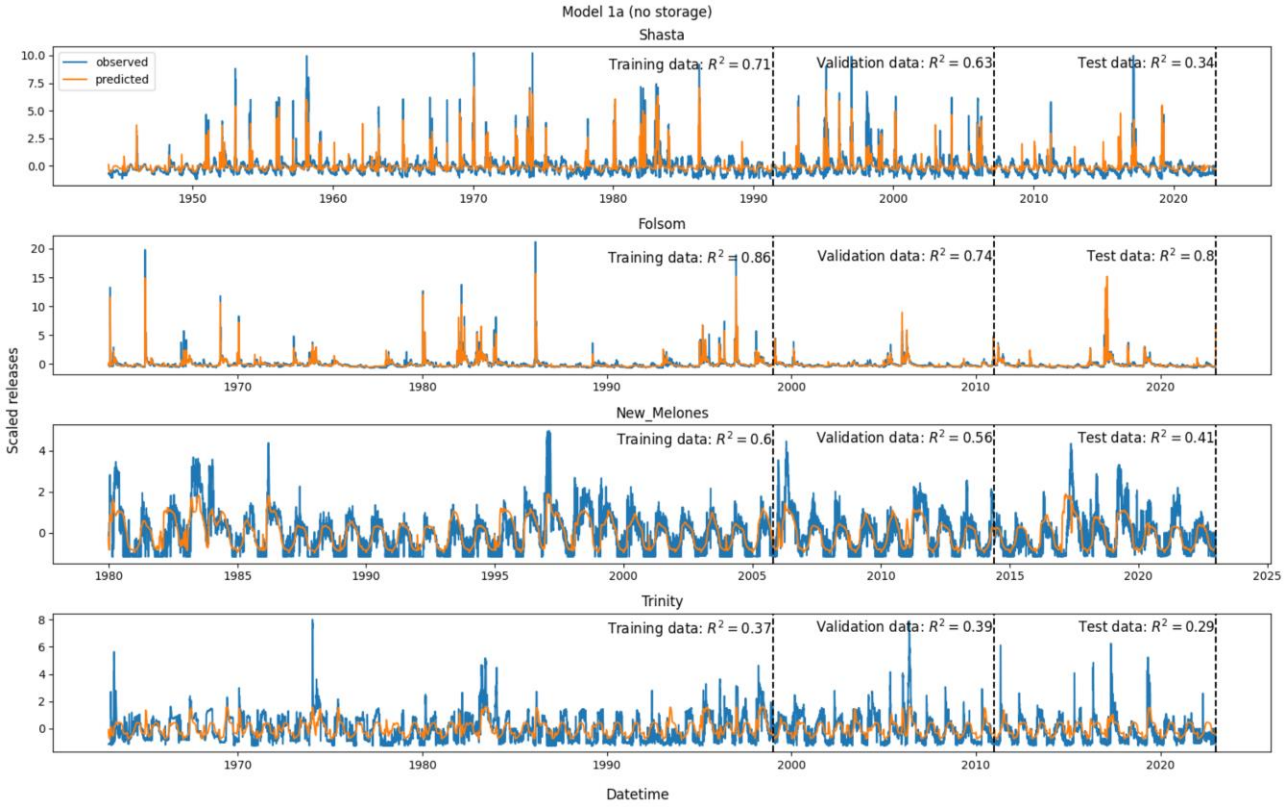
The moving performance experiments for other reservoirs (besides Shasta) are difficult because of limited out-of-sample data to test performance outside the training window. Decrease the training window, we may see a model that is severely underfit (see Trinity). Increase the training window, there may be barely enough out of sample data to make a reasonable judgement (see New Melones). Decrease the window size, there may be too much noise to decern long term trends.

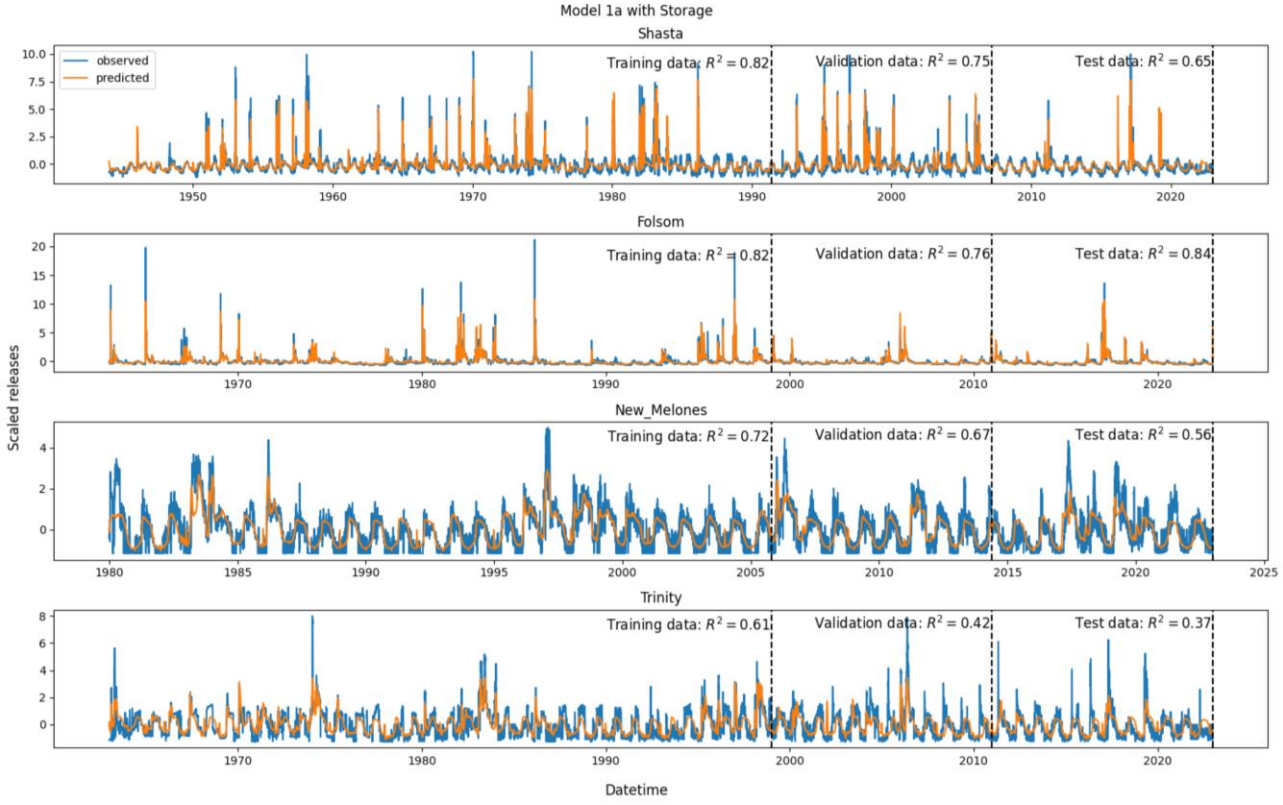


As before, we note the sharp decline in performance for Shasta beyond the training window. If it were overfitting, we would suspect the performance to stabilize at some point in contrast to a continual decrease. For example, in the rolling window performance figure below, performance for Folsom clearly stabilizes yet Shasta continues to decline. We suspect that the Folsom model suffered slightly from overfitting, while the Shasta model may also be affected by long term changes in hydrology or operating policy. It remains an open question about whether or not declines in performance is due to overfitting, changes in hydrology, or changes in operational policy. *Is there potential for machine learning models to detect these changes?*

3.3 Training with and without explicit storage data

A recurring theme of this study was to examine whether or not LSTM reservoir models learn to conserve mass and learn representations of storage in their cell states without guidance. Previously, we found that an LSTM model trained on Shasta reservoir learned the *seasonality* of storage but not storage values themselves in the cell states. This does not preclude the possibility that the models learn to maintain storage states through a more complex representation, though this case would not be as interpretable as we hope. Nonetheless, it makes sense to compare the behavior and performances of LSTM models trained with and without storage (shown below).


Model 1a (no storage)
Shasta
observed
predicted
Training data: R^2 = 0.71
Validation data: R^2 = 0.63
Test data: R^2 = 0.34
Folsom
Training data: R^2 = 0.86
Validation data: R^2 = 0.74
Test data: R^2 = 0.8
New_Melones
Training data: R^2 = 0.6
Validation data: R^2 = 0.56
Test data: R^2 = 0.41
Trinity
Training data: R^2 = 0.37
Validation data: R^2 = 0.39
Test data: R^2 = 0.29
Scaled releases
Datetime


Model 1a with Storage
Shasta
observed
predicted
Training data: R^2 = 0.82
Validation data: R^2 = 0.75
Test data: R^2 = 0.65
Folsom
Training data: R^2 = 0.82
Validation data: R^2 = 0.76
Test data: R^2 = 0.84
New_Melones
Training data: R^2 = 0.72
Validation data: R^2 = 0.67
Test data: R^2 = 0.56
Trinity
Training data: R^2 = 0.61
Validation data: R^2 = 0.42
Test data: R^2 = 0.37
Scaled releases
Datetime

Note that for simplicity and computational efficiency, we use the same set of hyperparameters used by tuning the Shasta model for each reservoir. The table below summarizes the training, validation, and test R^2 for each reservoir with and without storage.

No Storage				
	Shasta	Folsom	New_Melones	Trinity
Train	0.714	0.861	0.600	0.365
Validation	0.635	0.743	0.555	0.388
Test	0.340	0.802	0.410	0.291

Storage Included				
	Shasta	Folsom	New_Melones	Trinity
Train	0.816	0.820	0.721	0.611
Validation	0.753	0.762	0.674	0.417
Test	0.651	0.838	0.556	0.372

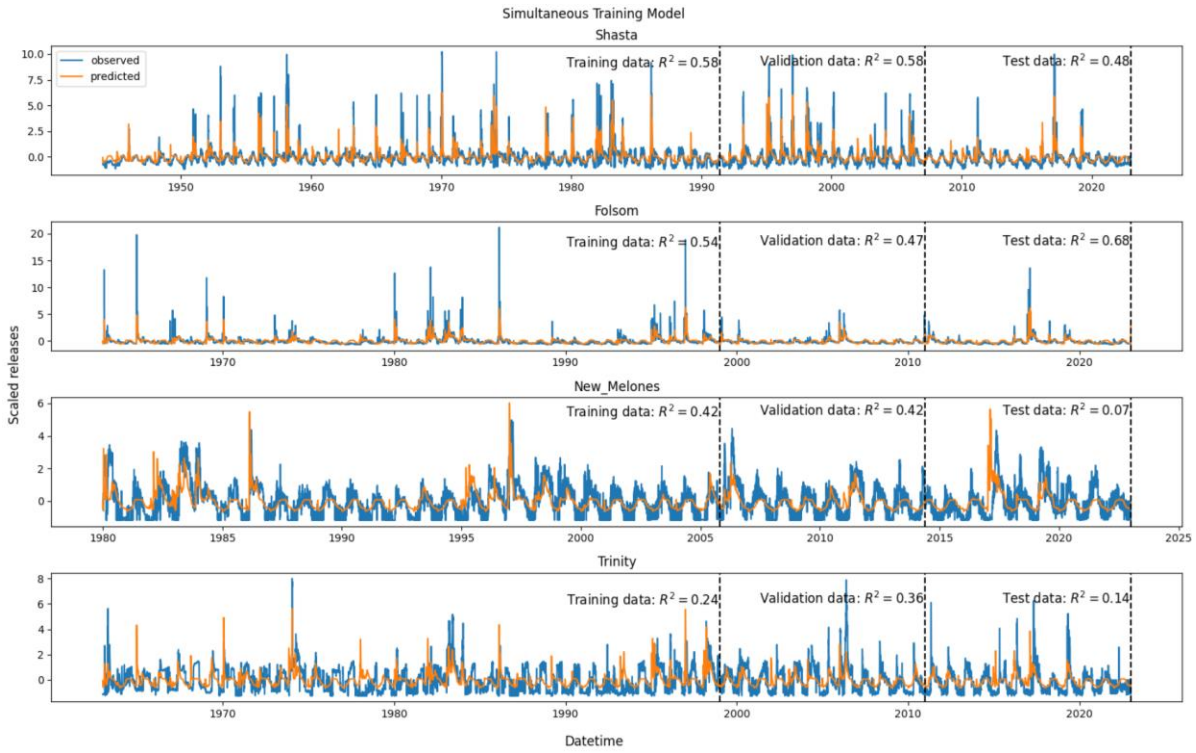
We notice a few observations. One, an early difficulty of modeling with only inflow for Shasta reservoir was the appearance of false peaks, but the Shasta model that was trained with storage data appears to have resolved this problem. *This is evidence that suggests the no-storage Shasta model was unable to learn storage states on its own, or if it did, utilized that information suboptimally*. Model 3 comes to mind for the latter observation, where implied storages were maintained externally and fed to the model, yet the problem of false peaks persists even though we showed that the implied storages matched true storages reasonably well. The model exhibits fewer and less severe false peaks than the benchmark models, though this may be due to the learning of some other long-term interaction. *Nonetheless, we cannot assume that an LSTM will learn to conserve mass on its own.*

Second, adding storage significantly improves the performance of the models for Shasta and New Melones, but not Folsom or Trinity. Additionally, we do not notice false peaks for either the no-storage model or the storage-included model for Folsom. *Does the strong performance of the Folsom model without storage indicate that it has successfully learned storage on its own, or is storage not an important additional predictor for this reservoir?*

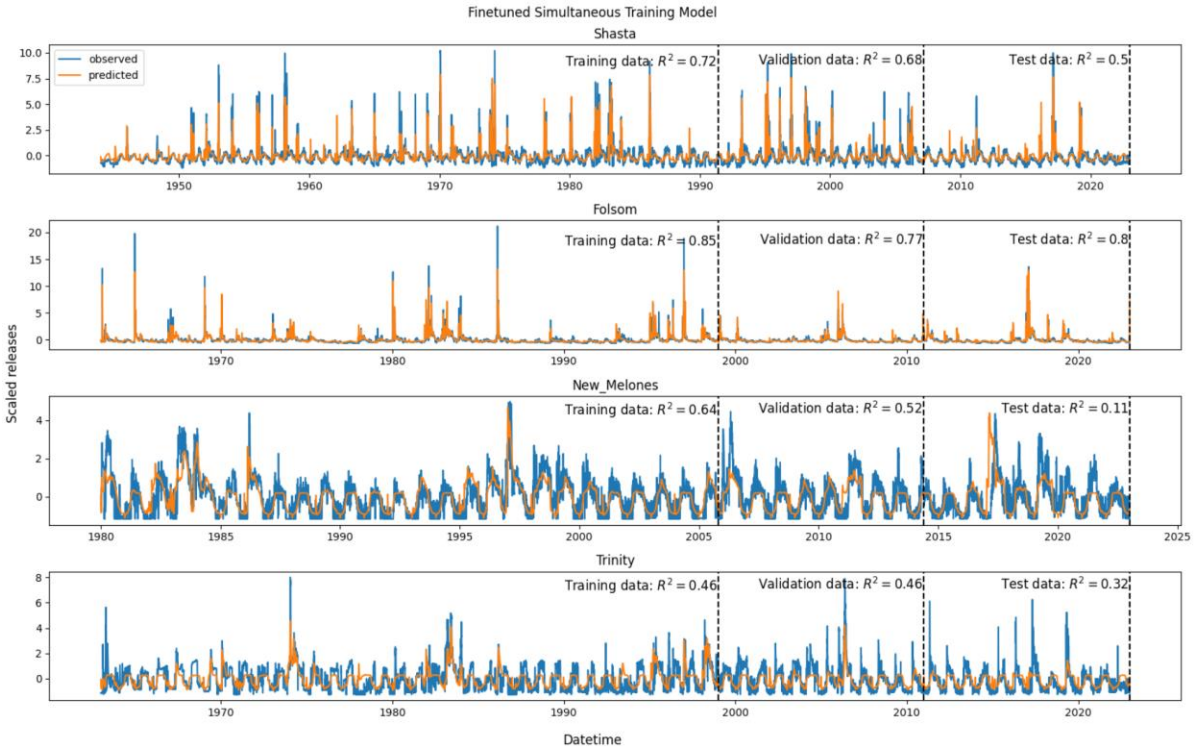
Finally, we note surprisingly that the performance in test for Folsom is not worse than validation, which is what we see for the other reservoirs. This suggests several possibilities. First, the operating policy or hydrology has not changed appreciably with time. Second, the policy for Folsom has not changed appreciably, and the learned policy is insensitive to changes in hydrology. We previously showed that the hydrology for Shasta is drier in test than training or validation, and it is reasonable to expect the same for Folsom.

3.4 Simultaneous training of reservoirs

In rainfall-runoff modelling, training simultaneously on multiple basins lead to improved results. However, the same may not be true for modeling reservoir releases since each reservoir may be operated differently enough so that they cannot be treated as iid samples for the machine learning model. In contrast, the physics of rainfall-runoff can generalize between basins. We demonstrate this by simultaneously training on Shasta, Folsom, Trinity, and New Melones, where the performance for each reservoir is worse compared to training on each reservoir individually.



We can also finetune the simultaneous model to calibrate it to each reservoir. However, the results after finetuning are not better than training individually, which indicates that there is limited additional knowledge learned by first simultaneously training.



4 Future work idea

The question of detecting policy or hydrology changes with machine learning (or other parameterized models) remains an interesting question, despite challenges from real data (limited data history, difficult to decompose policy and hydrology changes). It may be interesting to pursue these questions using synthetic hydrology and a synthetic reservoir operating policy. Then we have the ability to create artificial changes to hydrology or operating policy so we can study how the models respond to these changes.