UTRECHT UNIVERSITY

FACULTY OF SCIENCE

Applied Data Science

Final Thesis Project

Using artificial neural networks to improve hydrological streamflow predictions from PCR-GLOBWB

Oriol Pomarol Moyà

—
—

Advisors

Edwin Sutanudjaja (UU), Derek Karssenberg (UU)

July, 2022

Contents

1 Introduction

Streamflow is a most important hydrological variable which can affect agriculture, ecosystems and, ultimately society. Being able to accurately predict long-term streamflow can lead to an improvement in water resource management as well as flood mitigation.

The use of machine learning (ML) models in hydrology has been rapidly spreading in the past few years due to their higher performance and flexibility. C. Shen et al. (2021) showcases many different ML applications in a wide range of hydrology-related topics. Such an increase in popularity is not only due to the increase of computational power but also to the large quantities of data that has become available, which contain significantly more information than hydrologists have been able to capture within theories (Nearing et al., 2021). Whenever ML models have been benchmarked against calibrated conceptual models or process-based models, they have always generally performed better. To try and get the best of both worlds, though, some researchers have implemented coupled models that correct predictions from a theory-based model using ML methods.

Isik et al. (2013), for example, used two fully connected artificial neural networks to obtain baseflow and stormflow from the surface runoff as determined by the SCS-CN model as well as other meteorological variables. A much simpler approach was followed by Noori and Kalin (2016), who created a fully connected artificial neural network to improve the streamflow prediction from the baseflow and stormflow outputted by the physically-based SWAT model. Y. Shen et al. (2022) used a Random Forests approach taking an extensive set of hydrological state variables simulated with the PCR-GLOBWB model as well as precipitation, temperature and evapotranspiration, improving the performance of the model significantly at three streamflow gauging stations in the Rhine basin.

In parallel, the use of ANNs in hydrology has been extensively evaluated in recent years due to their ability to learn non-linear relationships between variables and find relationships between those and the output without prior knowledge of the physical characteristics of the problem. Maheswaran and Khosa (2012) found that the studied streamflow series from two rivers in India showed significant results for the presence of non-linear features. Addi-

tionally, in the study from Duan et al. (2020), the linear model performed the worst among other neural network alternatives when predicting streamflow from meteorological variables in almost all of the studied basins, which they attributed to the non-linearity of the prediction problem. On top of that, in the same study the models which took into account the temporal dependency of the data also performed better than a fully connected network, especially the Temporal Convolutional Neural Networks. Other articles such as Gao et al. (2010) have also used artificial neural networks to predict streamflow from meteorological data in order to study the impact of different climate change scenarios in hydrology.

This project aims to improve the streamflow predictive performance by use of an ANN-based error-correction model, using the input from the PCR-GLOBWB model and meteorological data in a similar setting as Y. Shen et al. (2022). In that paper, the use of a Random Forests model produced a significant increase in performance from the base PCR-GLOBWB model, which raises the question of whether similar or better results can be achieved with already proven ANN methods.

Two ANN architectures have been executed, a fully connected neural network (FCNN) and a Temporal Convolutional Neural Network (TCNN). While the first is a more generic and widely used ANN architecture, the second takes into account the temporal information from the data and has been shown by Duan et al. (2020) to outperform other common time-series approaches such as LSTM or GRU. To check whether such complex models are worth using, a Multiple Linear Regression (MLR) is also fitted to the data as a baseline. The data is fed to the models in two formats; adding their lagged versions up to a certain threshold, or not.

The main research question is then the following: How much can artificial neural networks improve streamflow prediction performance from PCR-GLOBWB in the Rhine basin? This will be achieved through answering four sub-questions grouped in two main topics.

1. Are non-linear ANN models significantly better than a multiple linear regression for streamflow prediction? Are the predictions of such models performing equally well for all streamflow values?

2. Does the introduction of lagged variable information significantly improve the performance of the error-correction models? Can a time-series specific model (TCNN) perform significantly better than other more generic approaches (MLR, ANN)?

In the following sections we will explore the data used in this project, the methods put into practice to respond to these questions, the obtained results, and a brief discussion to conclude.

2 Data

The data from this study belongs to two different locations in the Rhine basin: Basel and Lobith. It was obtained from GitHub (Y. Shen, 2021) and contains daily observations consisting of 3 meteorological features and 18 simulated state variables using the PCR-GLOBWB model for a period of twenty years, from 1981 to 2000. The data was already cleaned and complete.

The meteorological variables are precipitation, temperature and reference potential evapotranspiration, all of which have been commonly adopted for both direct streamflow prediction (Duan et al., 2020; Gao et al., 2010) and error-correction models (Isik et al., 2013; Y. Shen et al., 2022).

As mentioned before, the rest of the predictors fed into the machine learning model are the variables simulated by PCR-GLOBWB, which are listed in Table 1. More information on the PCR-GLOBWB model can be found in Section 3.1.

The predicted variable fed to the ML models is the PCR-GLOBWB residual, obtained by subtracting its streamflow prediction from the actual observations of the two gauging stations for each location. As can be seen in Figure 1, despite the noise, there is a clear seasonal variation of the observed streamflow that differs between locations. In Basel, there is a nival regime characterized by high discharge in summer due to the melting ice, while in Lobith there is a pluvial regime which typically corresponds to higher discharge in winter and spring as a direct consequence of precipitation. The residuals obtained from the uncalibrated PCR-GLOBWB model also show a

certain seasonal variation, generally performing worse for higher runoff values. The figure shows only half of the full extent of the data, to allow for easier visualization of the seasonal variability.

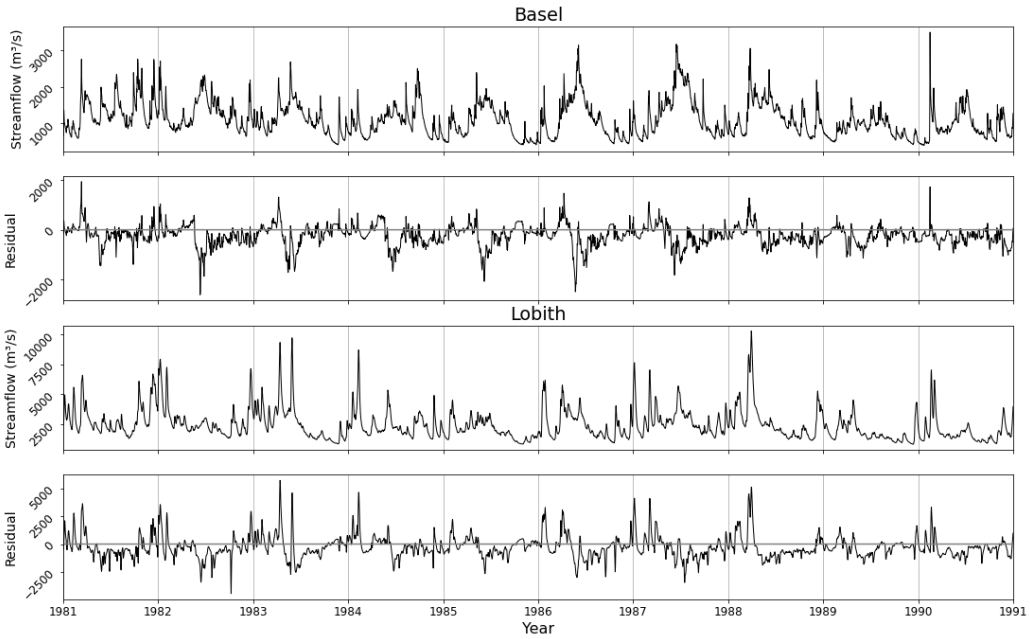

Figure 1: Time series of the observed streamflow and residual from the PCR-GLOBWB prediction for Basel and Lobith between 1981 and 1990.

3 Methods

An error-correction approach has been followed for this project, which works as follows. First, the PCR-GLOBWB model is run for the selected time period and location to obtain its simulated state variables and streamflow prediction. Then, an ML model is fit using both the state variables and some additional meteorological variables in the training set to output the residuals from the PCR-GLOBWB streamflow prediction. To make models aware of seasonality, the day of the year was also included as a predictor. For the test set, the output of the ML model is summed to the one from PCR-GLOBWB to obtain the definitive streamflow prediction.

In the next sub-sections both PCR-GLOBWB and ML models (MLR, FCNN and TCNN), as well as their choice of parameters, will be explained.

3.1 PCR-GLOBWB

The PCR-GLOBWB model (Sutanudjaja, 2017) is a grid-based global hydrology and water resources model which can predict streamflow as well as other state variables described in Table 1.

The model supposes three layers, two soil moisture storage and groundwater storage. Each of those has its water flow (surface runoff, stormflow and baseflow), as well as water exchange between them, with the atmosphere on the top layer and with human interaction through industry, livestock, domestic use and irrigation.

Table 1: Simulated state variables obtained from the PCR-GLOBWB model as described in Y. Shen et al. (2022).

Variable name	Unit	Explanation
baseflow	m/day	baseflow, groundwater discharge
directRunoff	m/day	surface runoff
domesticWaterWithdrawal	m/day	domestic water withdrawal
gwRecharge	m/day	groundwater recharge, fluxes from the lower soil layer to groundwater stores
industryWaterWithdrawal	m/day	industrial water withdrawal
interflowTotal	m/day	interflow, shallow sub-surface flow
irrigationWaterWithdrawal	m/day	water withdrawal allocated for irrigation purposes
livestockWaterWithdrawal	m/day	water withdrawal allocated for livestock demand
nonIrrWaterConsumption	m/day	non-irrigation sectoral (domestic, industry and livestock) water consumption, i.e. non-irrigation sectoral withdrawal minus return flow
snowCoverSWE	m	snow cover/storage in water equivalent thickness (excluding liquid part)
snowFreeWater	m	liquid water/meltwater storage in the snowpack
storGroundwater	m	groundwater storage (renewable part)
storUppTotal	m	S1 actual upper soil water storage
storLowTotal	m	S2 actual lower soil water storage
surfaceWaterStorage	m	surface water storage (lakes, reservoirs, rivers, and inundated water)
totLandSurfaceActuaET	m/day	total evaporation and transpiration from land part
storGroundwater	m/day	total evaporation and transpiration from land and water body parts

The model was run with a daily time step and a spatial resolution of 30 arcmins. For the rest, the default parameter values were used since (Y. Shen et al., 2022) found that, after applying the error-correction model, the performance of calibrated and uncalibrated PCR-GLOBWB runs for different locations were equally good, so this extra step is not necessary.

3.2 Error-correction

As introduced previously, to compare the importance of time-series in streamflow prediction, two ANN architectures were tested: a fully connected neural network (FCNN) and a temporal convolutional neural network (TCNN). Additionally, a Multiple Linear Regression (MLR) was used as a benchmark. All of these models are explained in the following subsections.

3.2.1 MLR

The multiple linear regression is the simplest model of them all since it takes into account only linear dependence between the dependant and independent variables. It was implemented using the *LinearRegression()* method from *sklearn* library using the default setup. Linear regression was also used as a baseline for other ML models in (Duan et al., 2020).

3.2.2 FCNN

A fully connected neural network is a type of ANN, a family of ML models that uses layers of units called neurons, which take a numerical input and output its sum after applying a certain activation function. For FCNN in particular, every neuron takes its input from all the neurons in the previous layer. This is a generic structure that can adapt to many scenarios and has been found to improve streamflow prediction when coupled with physically-based models (Noori and Kalin, 2016).

The model was built using Keras, and the parameters of the network were tuned using validation mean squared error (MSE) to compare them. After

a few test runs, it was observed that *relu* activation function was favored instead of *sigmoid*, and was also used in Duan et al., 2020. The learning rate was also mirrored from that article, setting it at 0.0005 using the Adam optimizer, as well as the number of hidden layers, which was set to 2. The number of neurons in each layer was allowed to vary between 25 and 200 in steps of 25 for each run, using the Hyperband tuner from KerasTuner. Finally, a 0.05 dropout layer was added after every hidden layer to improve generalizability.

3.2.3 TCNN

The temporal convolutional neural network follows the concept of convolutional neural networks, in which only a subset of nearby neurons, defined by a kernel, is used as input for the neurons in the next layer. TCNN uses this structure for time-series data. The main difference is that due to the temporal nature of the data, it must use casual convolutions, which means that a neuron can only receive information coming from past time steps. Additionally, it uses dilated convolutions as depicted in Figure 2, which reduces the number of parameters speeding up its training. Duan et al., 2020 found that TCNN performed better than other deep network architectures such as FCNN, LSTM or GRU when predicting streamflow, and complimented its potential to perform future hydrology projections.

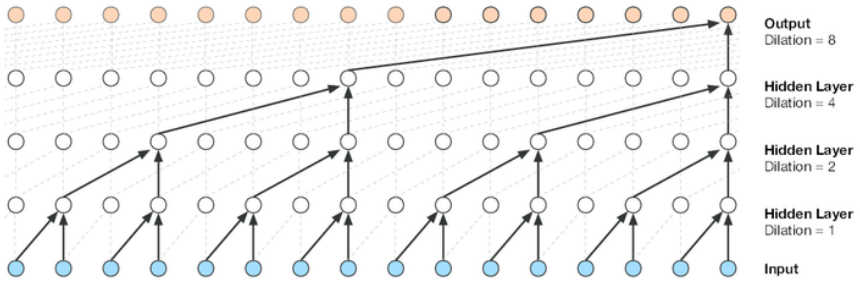

Figure 2: Visualization of a stack of dilated causal convolutional layers (Oord et al., 2016)

The *keras-tcn* (Rémy, 2022) library was used to implement the TCN layer with a kernel size of 3 and dilation of (1,2,4,8), producing a receptive field size of 61, meaning how far can the model see in terms of time-steps. This value

must not be confused with the actual time window inputted to the model, which must be smaller so that all of the information provided can be "seen" by the model. Additionally, a dropout rate of 0.05 (as recommended by the author) and layer normalization were applied. The *nb_filters* parameter which increases the complexity of the model was tuned optimized with *keras-tuner* similarly to the FCNN between 25 and 200 in steps of 25.

3.3 Model setup and evaluation

For the input variables, two different settings were tested. For the *no_lag* models, all the variables described in Section 2 were fed directly to the MLR and FCNN models. The *lag* models, on the other hand, include lagged versions of all variables to introduce the time-series information to them. See Table 2 to check all of the analyzed models.

The lagged versions of the data are fed as new variables, generating as many as the number of old variables times the desired lag parameter. For TCNN the data must be formatted to follow a different structure that uses time windows instead, but the information available to the model is the same. Each variable was standardized based on training data only, not to overestimate the performance, and the lagged versions of the variables were standardized using the parameters from their non-lagged counterparts. The number of past time steps added for all of the variables was 60 since a parallel thesis project based on the same data found that there was still a significant correlation between the past values from some variables up to this number and the streamflow observations. On the other hand, adding more lag would have increased the running time of the algorithms to unattainable levels with the time and resources available for this project.

To evaluate the models, cross-validation was employed taking 80% of the data each run as training, leaving the remaining 20% unused during model parameter tuning so that the performance on unseen data could be tested. In total, this generates five possible train-test splits comprised pf compact blocks of adjacent test set data, that can provide a better idea about the performance stability of the models. Moreover, to study the intrinsic variability of the ANN models, they were also run five extra times on a fixed test set. The review of all of the model runs can be seen in Table 2.

Table 2: Type of model, inclusion or not of lagged variables, the given name and number of runs for each error-correction mode with different (1-5) or same (5) test set in each location.

			Runs	
Model	Lagged vars.	Name	Test set 1-5	Test set 5
MLR	No	mlr_no_lag	5x	-
	Yes	mlr_lag	5x	-
FCNN	No	fcn_no_lag	5x	5x
	Yes	fcn_lag	5x	5x
TCNN	Yes	tcn_lag	5x	5x

All of the code was run on Google Colab. MLR was the fastest model by far, taking less than 10s for each run, even when including the lagged variables. FCNN runs took 3-4 min when not using lagged variables, and 4-5 min including them. When TCNN was run with the same hardware settings, it was roughly estimated to take up to 4h for each run which was not viable, but using GPU boosting greatly reduced it to 6-12 min.

The metric chosen to assess model performance is the Kling-Gupta efficiency (KGE), which is gaining dominance in recent hydrology literature and is preferable to other popular metrics such as Nash-Sutcliffe Efficiency for streamflow prediction since it can better capture the data seasonality (Y. Shen et al., 2022). To achieve that, it combines the three components of Nash-Sutcliffe efficiency, correlation r, bias α and ratio of variances β, in a more balanced way. Equation 1 shows how KGE is obtained from these components.

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{1}$$

4 Results

The performance obtained from running the models are displayed in Figure 3. The box plots help visualize the results obtained by running the models with different train and test samples (see Section 3.3).

It is clear that all of the error-correction models show a great improvement over the performance of PCR-GLOBWB in both analysed locations. In Basel, FCNN showcases a slightly better average performance and less variability between test sets compared to MLR. In Lobith, the non-lagged version of FCNN also performs better than MLR on average but this time the model variation is higher.

Including lagged variables slightly decreases the performance of the FCNN models in both locations, as can be seen in the second column of Figure 3. While the PCR-GLOBWB model data is the same, it is displayed again for two reasons; it makes it easier to compare against the lagged error-correction models, and also the test regions change slightly due to the introduction of lagged variables. TCNN performs better than FCNN in both locations, but both models fail to match the increased performance of MLR.

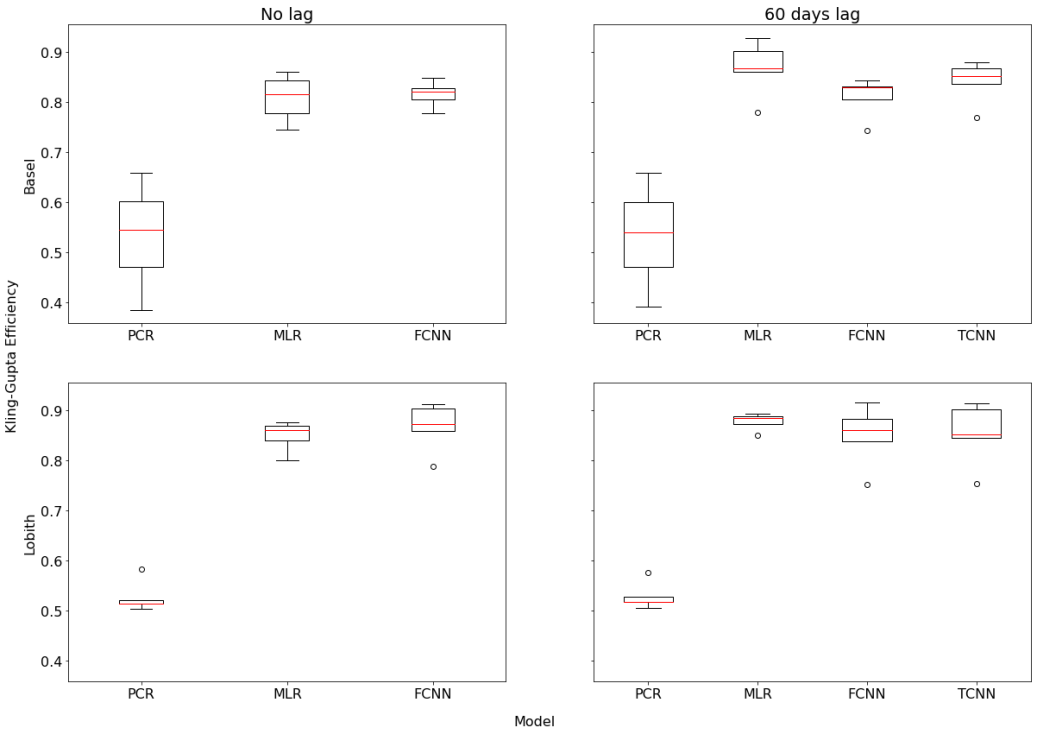

Figure 3: Distribution of KGE values for PCR-GLOBWB and the error-correction models with 60 days lagged variables (or time window for TCNN) and without it for Basel and Lobith.

The points that the box plots represent as outliers only coincide for FCNN

and TCNN in a specific location, but do not seem to indicate any relevant feature of the data or the models, other than its variability.

According to the Welch's t-test, the only significant differences in performance are between PCR-GLOBWB and all of the error-correction models, so the difference between MLR, FCNN and TCNN with or without the use of lagged variables can not be considered big enough to extract any statistically significant conclusions.

The results in this project exceed the KGE obtained by Y. Shen et al. (2022) using the same data, although in that occasion the models were trained only with half the total extent of the data and they used a lag of 10. In Basel, their best performing RF-based error-correction model only achieved a KGE of 0.75, while the models showcased in this project show average results ranging from 0.80 to 0.87. In Lobith, the improvement is not so much, 0.85 for RF compared to the newly obtained 0.85-0.88. As a reference, a sibling project that used the same data and an ANN approach, this time predicting directly the streamflow from meteorological variables, obtained a KGE of 0.72 in Basel and 0.84 in Lobith.

The results from all of the models run on test set 5 data are shown in Figures 4 and 5 for Basel and Lobith respectively. The observed values are displayed in black, the PCR-GLOBWB model prediction in blue and, for each plot, the different error-correction model predictions are in blue.

The predictions from all of the models show a similar shape; they inherit a struggle to accurately predict high values of streamflow from PCR-GLOBWB, while, for low-flow, they fit the observations much better. In general, PCR-GLOBWB tends to overestimate the streamflow predictions, a tendency which is fixed by the error-correction models that even underestimate a bit the results, especially in Lobith. It can be seen that there is an unusual streamflow peak in early spring of 1999 in both locations that all models fail to properly predict, even though there is an improvement compared to PCR-GLOBWB. It can also be noted that *mlr_lag*, the best performing model, is able to better capture the variability in observations whereas the others only modify the PCR-GLOBWB values in a more generic way missing the higher frequency patterns of the observations. Examples of this behaviour can be observed in autumn of 1999 for Basel, and late spring of 2000 for Lobith.

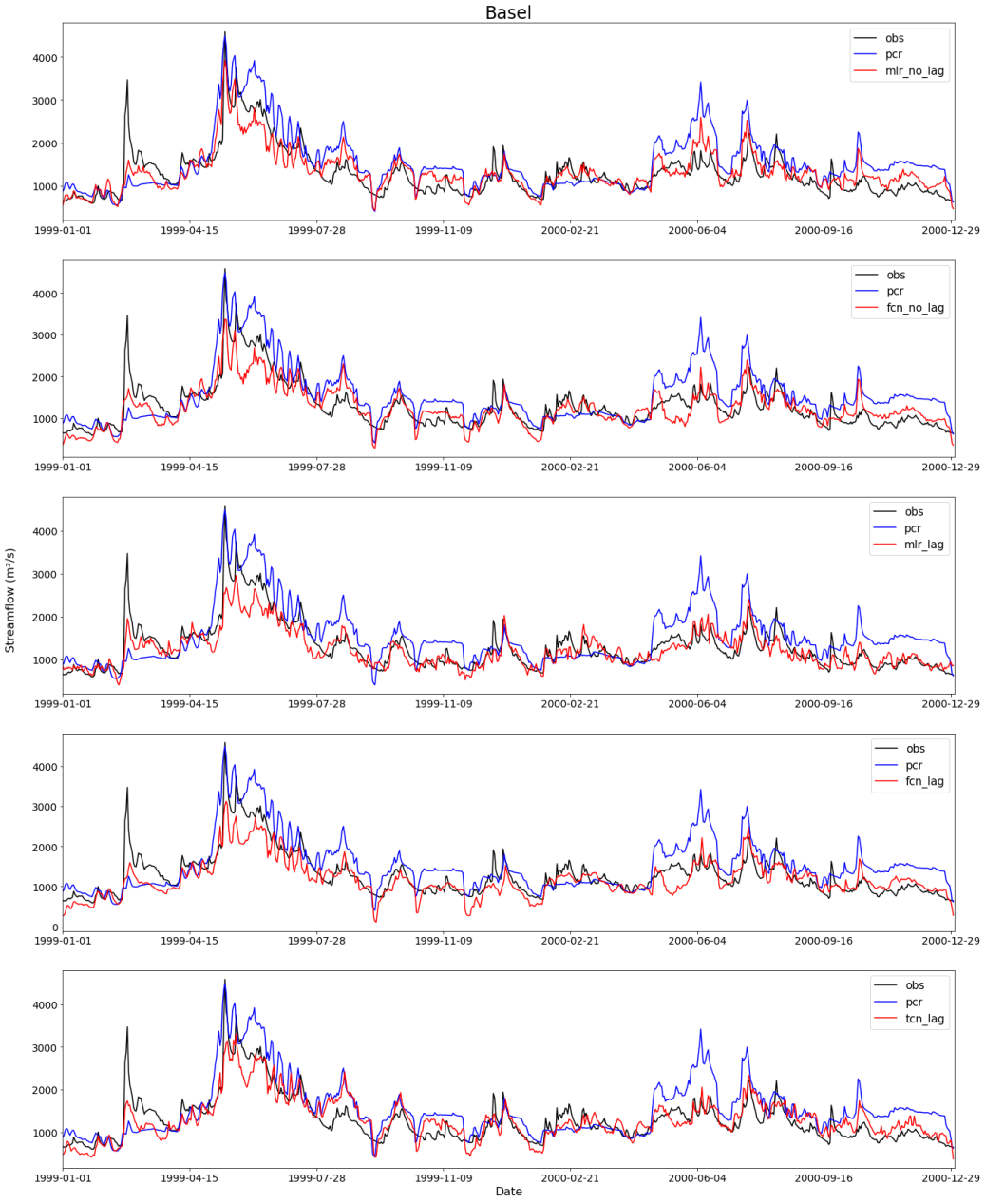

Figure 4: Streamflow observations and predictions from PCR-GLOBWB and the different error-correction models in Basel from years 1999 and 2000.

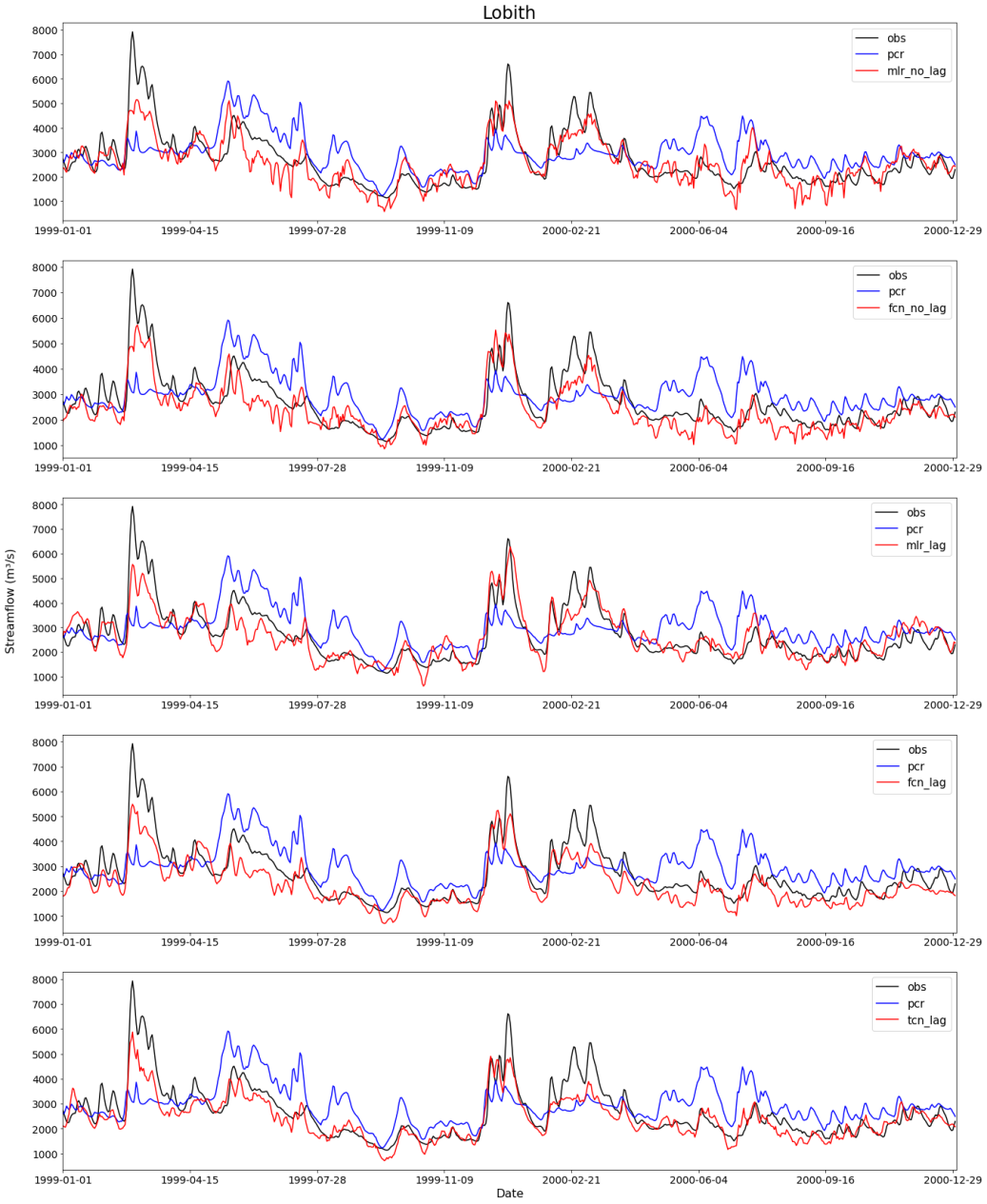

Figure 5: Streamflow observations and predictions from PCR-GLOBWB and the different error-correction models in Lobith from years 1999 and 2000.

4.1 Model variability

Contrary to MLR, there is some variability in the performance of the ANN models when run on the same train and test set, mainly due to the parameter tuning, which relies on a small validation set, but also to their stochastic nature. The results, though, showed very similar performance within runs.

For FCNN, the standard deviations of KGE values obtained by repeatedly running it on test set 5 were 0.011 and 0.025 (without and with lagged variables) for Basel, and 0.018 and 0.029 for Lobith. This is half the amount obtained when running the model on the different test sets, 0.026 and 0.049 for Basel, and 0.040 and 0.062 for Lobith. For TCNN, this ratio was more than six times less, from 0.043 to 0.006 in Basel and from 0.063 to 0.010 on Lobith. Therefore, it can be concluded that the biggest part of the variance comes from using slightly different train set (3/4 of the train set is shared between any pair of runs) and especially different test sets.

It is also interesting to analyze how the tuned parameters varied between runs. For the non-lagged version of FCNN, the number of units for both layers oscillates between 200 and 175 consistently in the two locations. The lagged FCNN model still features 200 and 175 as the most common number of units, but in Basel there is more variability and the number of units drops to much lower values in some cases even when using the same test set. For the TCNN model, the *nb_filters* parameter also tends to reach the upper bound, either 200 or 175, showing complete coherence between runs within the same test set for both locations.

4.2 Performance consistency

In order to analyze if the models perform constantly well for different observed values of streamflow, their cumulative frequency curves are plotted against the one obtained from the observations in Figures 6 and 7. In this plot, the data belongs to the full range of data made possible by cross-validation, providing a more general picture of how the models perform.

A common behaviour for all models is to overestimate the amount of extremely low streamflow values, up to approximately 500 m^3/s in Basel and

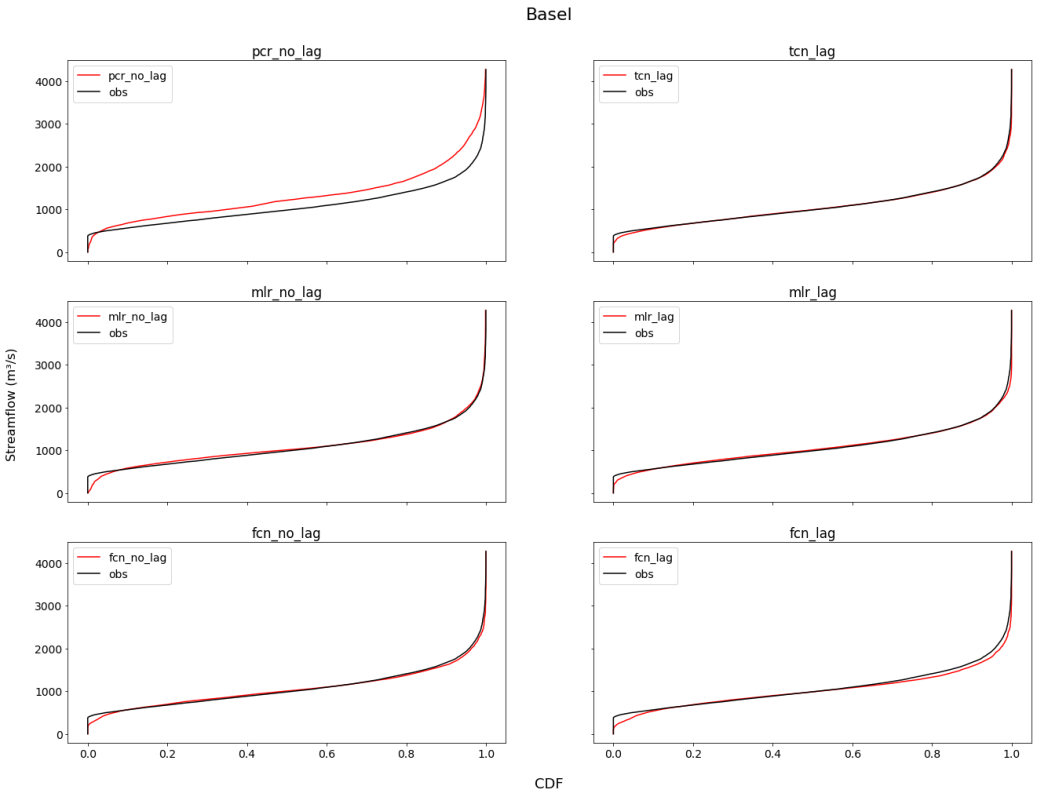

Figure 6: Cumulative frequency curve of the observations compared to PCR-GLOBWB and the five error-correction models in Basel.

1000 m^3/s in Lobith, which are extremely rare in real life. Surprisingly, all models seem to perform worse than PCR-GLOBWB in that aspect, and only TCNN shows a distribution that closely resembles the observations reliably for both locations.

The bulk of the data, comprised between 500-1500 m^3/s in Basel and between 1000-3000 m^3/s in Lobith is not properly captured by the PCR-GLOBWB model, which clearly overestimates predictions for this range of values, which is fixed for the rest of the models, as exemplified in the streamflow vs date plot. The MLR still suffers from this overestimation, although the lagged version fits the observations much better. For the FCNN with and without lag, this only happens towards the highest section and, for TCNN, the curve even shows almost a perfect fit in this extent.

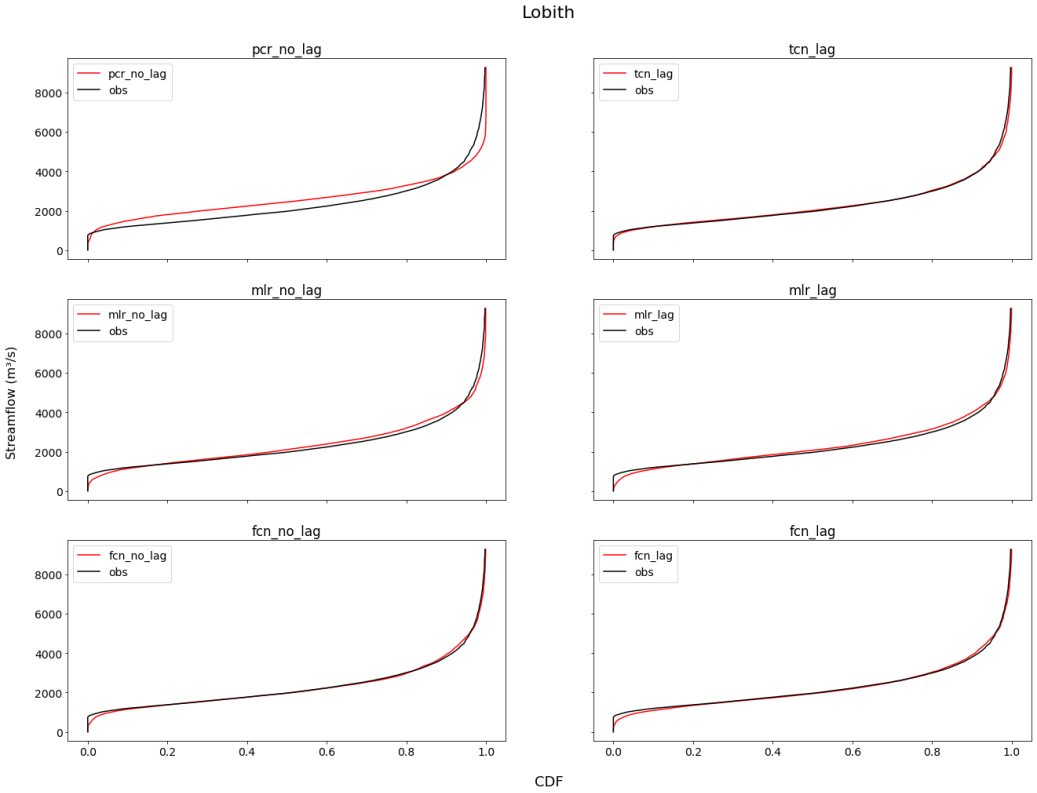

Figure 7: Cumulative frequency curve of the observations compared to PCR-GLOBWB and the five error-correction models in Lobith.

Finally, for the highest observed streamflow values above 1500 m^3/s in Basel and above 3000 m^3/s in Lobith, there is a different behaviour between the two locations. In Basel, PCR-GLOBWB overestimates the presence of high streamflow values, while the rest of the models, except TCNN that fits the curve very well in this range, showcase a slightly lower curve at first but is very similar to the observations for the highest values. In Lobith the opposite happens, as the prediction curve is much steeper than the one from the observations in the higher part of the spectrum indicating that the models fail to predict the largest values. This is more noticeable in PCR-GLOBWB but can be observed to an extent in the other models. In this occasion both of FCNN models have the better fit.

To further support this evidence through another point of view, plots of the observed against predicted streamflow values have been obtained and

showcased in Figures 8 and 9 in Appendix A. A few more characteristics of the predictions are also discussed there.

5 Conclusion and Discussion

The results clearly show a remarkable increase in the performance of the error-correction models over PCR-GLOBWB in both locations, confirming the validity of this approach. On the other hand, the use of ANNs does not provide a significant change in performance from linear regression. On the other hand, assigning more data to the training set improved the performance of all models compared to Y. Shen et al. (2022) even with the simplest model, so it would be interesting to analyze if it is possible to increase the performance further by using a bigger volume of data, which may in turn favor the ANNs.

The addition of lagged variables does not increase the performance significantly, although the best performing model is MLR with the addition of lagged variables up to 60 days. The TCNN model performs better than FCNN, which even decreases its performance slightly with the addition of lag, but the gain is not significant enough to draw any solid conclusions. On the other hand, TCNN showcases the most similar cumulative distribution of streamflow values to the observed one, of all of the models in both locations. Finally, the results show that the use of error-correction models compared to direct prediction using lagged meteorological variables improved the performance, and was especially remarkable in Basel, possibly due to the nival regime present in the area.

Even though in this study ANNs did not show a big enough improvement compared to linear regression that can justify their use, given their much higher complexity and computational expense, this should not be taken as proof to disregard the non-linearity of the problem. ANNs have many hyperparameters, the most important ones described in Section 3, which can affect their performance greatly (Claesen & De Moor, 2015). Due to the scope of this project and the computational power available, hyperparameter tuning had to be severely limited. Both units per layer in FCNN and *nb_filters* in TCNN took values close to the upper bound for all of the model runs, which

could indicate that higher values may provide better results. Additionally, a more in-depth analysis of the effect of the other hyperparameters on the performance of the models would be desirable.

Finally, the effect of using a subset of the input variables or the impact of choosing different lag values could be interesting for further research. Some preliminary results also suggest that the performance could fluctuate substantially between seasons.

References

Claesen, M., & De Moor, B. (2015). Hyperparameter Search in Machine Learning [Number: arXiv:1502.02127 arXiv:1502.02127 [cs, stat]]. https://doi.org/10.48550/arXiv.1502.02127

Duan, S., Ullrich, P., & Shu, L. (2020). Using Convolutional Neural Networks for Streamflow Projection in California. *Frontiers in Water*, *2*, 28. https://doi.org/10.3389/frwa.2020.00028

Gao, C., Gemmer, M., Zeng, X., Liu, B., Su, B., & Wen, Y. (2010). Projected streamflow in the Huaihe River Basin (2010–2100) using artificial neural network. *Stochastic Environmental Research and Risk Assessment*, *24*(5), 685–697. https://doi.org/10.1007/s00477-009-0355-6

Isik, S., Kalin, L., Schoonover, J. E., Srivastava, P., & Graeme Lockaby, B. (2013). Modeling effects of changing land use/cover on daily streamflow: An Artificial Neural Network and curve number based hybrid approach. *Journal of Hydrology*, *485*, 103–112. https://doi.org/10.1016/j.jhydrol.2012.08.032

Maheswaran, R., & Khosa, R. (2012). Wavelet–Volterra coupled model for monthly stream flow forecasting. *Journal of Hydrology*, *450-451*, 320–335. https://doi.org/10.1016/j.jhydrol.2012.04.017

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., & Gupta, H. V. (2021). What Role Does Hydrological Science Play in the Age of Machine Learning? [_eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2020WR028091]. *Water Resources Research*, *57*(3), e2020WR028091. https://doi.org/10.1029/2020WR028091

Noori, N., & Kalin, L. (2016). Coupling SWAT and ANN models for enhanced daily streamflow prediction. *Journal of Hydrology*, *533*, 141–151. https://doi.org/10.1016/j.jhydrol.2015.11.050

Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio.

Rémy, P. (2022). Keras TCN [original-date: 2018-03-22T02:40:06Z]. Retrieved May 26, 2022, from https://github.com/philipperemy/keras-tcn

Shen, C., Chen, X., & Laloy, E. (2021). Editorial: Broadening the Use of Machine Learning in Hydrology. *Frontiers in Water*, *3*, 681023. https://doi.org/10.3389/frwa.2021.681023

Shen, Y. (2021). Co822ee/PCR-GLOBWB_error-correction: V2. https://doi.org/10.5281/zenodo.5068517

Shen, Y., Ruijsch, J., Lu, M., Sutanudjaja, E. H., & Karssenberg, D. (2022). Random forests-based error-correction of streamflow from a large-scale hydrological model: Using model state variables to estimate error terms. *Computers & Geosciences*, *159*, 105019. https://doi.org/10.1016/j.cageo.2021.105019

Sutanudjaja, E. (2017). PCR-GLOBWB_model: PCR-GLOBWB version v2.1.0_beta_1. https://doi.org/10.5281/zenodo.247139

A Appendix

In Figures 8 and 9, it is reinforced that all models but TCNN tend to predict unrealistically low values of streamflow. Also and the limitations of the PCR-GLOBWB model to adapt to the different regions becomes clear, overestimating in Basel and failing to predict high streamflow values in Lobith, both correctly addressed by the error-correction models. A general increase in prediction variability can be seen as the streamflow values increase.

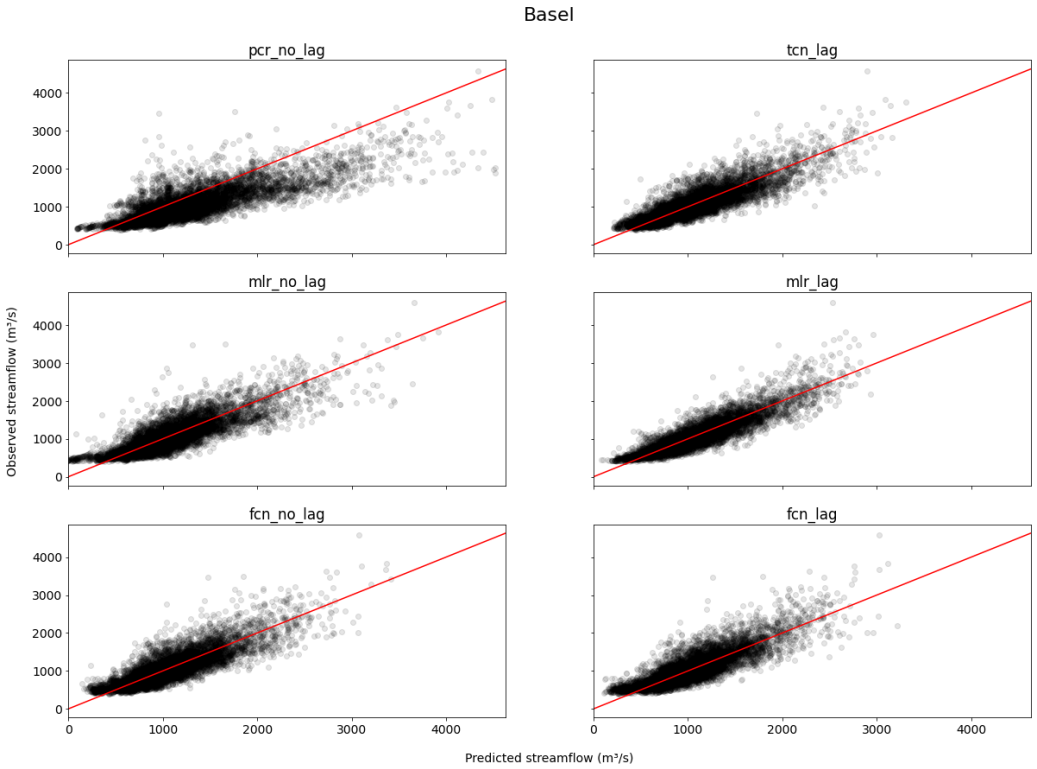

Figure 8: Plot of observed against predicted streamflow for the four models using lagged variables in Basel.

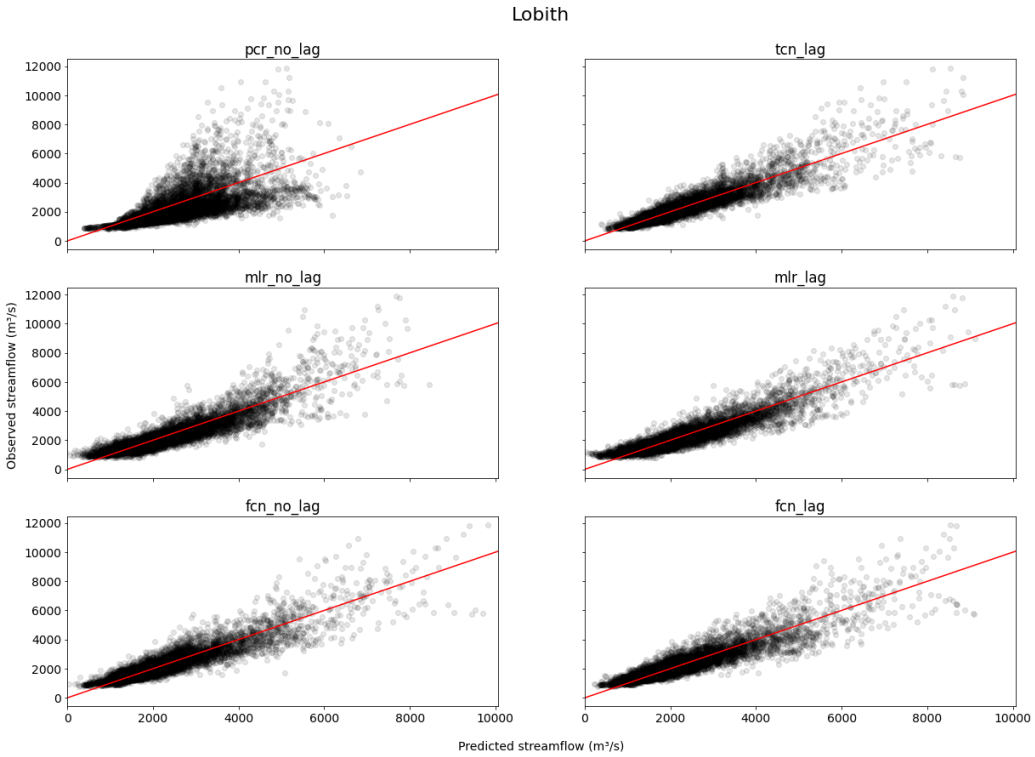

Figure 9: Plot of observed against predicted streamflow for the four models using lagged variables in Lobith.