Investigation of the effect of the hyperparameter optimization and the time lag selection in time series forecasting using machine learning algorithms

Georgia Papacharalampous, Hristos Tyralis, and Demetris Koutsoyiannis

European Geosciences Union General Assembly 2017
Vienna, Austria, 23-28 April 2017
Session IE3.2/NP4.3: Big data and machine learning in geosciences

EGU2017-3072
Available online at: itia.ntua.gr/1693

Department of Water Resources and Environmental Engineering,
School of Civil Engineering, National Technical University of Athens
(papacharalampous.georgia@gmail.com)

1. Abstract

The hyperparameter optimization and the time lag selection are considered to be of great importance in time series forecasting using machine learning (ML) algorithms. To investigate their effect on the ML forecasting performance we conduct several large-scale simulation experiments. Within each of the latter we compare 12 methods on 2 000 simulated time series from the family of Autoregressive Fractionally Integrated Moving Average (ARFIMA) models. The methods are defined by the set {ML algorithm, hyperparameter selection procedure, time lags}. We compare three ML algorithms, i.e. Neural Networks (NN), Random Forests (RF) and Support Vector Machines (SVM), two procedures for hyperparameter selection i.e. predefined hyperparameters or defined after optimization and two regression matrices (using time lag 1 or 1, . . . , 21). After splitting each simulated time series into a fitting and a testing set, we fit the models to the former set and compare their performance on the latter one. We quantify the methods' performance using several metrics proposed in the literature and benchmark methods. Furthermore, we conduct a sensitivity analysis on the length of the fitting set to examine how it affects the robustness of our results.

The findings indicate that the hyperparameter optimization mostly has a small effect on the forecasting performance. This is particularly important, because the hyperparameter optimization is computationally intensive. On the other hand, the time lag selection seems to mostly significantly affect the methods performance when using the NN algorithm, while we observe a similar behaviour for the RF algorithm albeit to a smaller extent.

2. Introduction

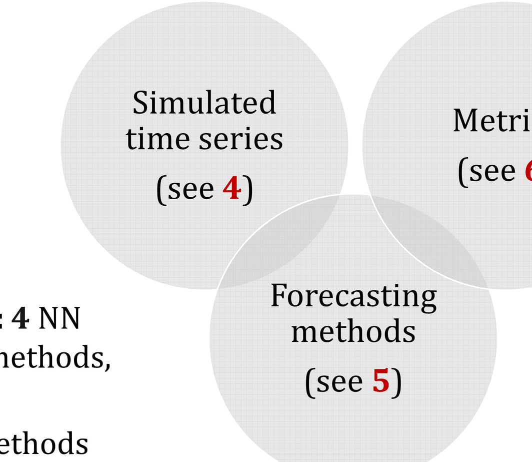
- **Machine learning (ML)** algorithms are widely used in geosciences and beyond as an alternative to stochastic methods. Popular ML algorithms are the:
  - ✓ Neural Networks (NN)
  - ✓ Random Forests (RF)
  - ✓ Support Vector Machines (SVM)
- The **hyperparameter optimization** and the **time lag selection** are considered to be of great importance in time series forecasting using ML algorithms. However, published research focusing on their effect in the ML forecasting performance is rather rare.
- The available studies using ML algorithms within case studies in geosciences (e.g. Koutsoyiannis et al. 2008) inevitably implement ML model structures that have not been prior investigated on a theoretical basis, perhaps after performing a small-scale experimentation on the used data sets. The latter is generally not presented for various reasons.
- Zhang (2001) aimed at the investigation of the capabilities provided by the NN concerning the foreasting of linear processes. The simulation experiments were based on 8 stochastic processes from the ARMA family and 30 simulated time series for each stochastic process.
- Recently, Papacharalampous et al. (2017b) conducted a multiple-case study using 50 time series of precipitation and temperature to detect systematic patterns across the various cases regarding the hyperparameter optimization and time lag selection in forecasting using ML algorithms. The ML algorithms investigated were NN and SVM.
- Papacharalampous (2016), Papacharalampous et al. (2017a), Papacharalampous et al. (2017c) also contributed to a small extent to the subject, despite their different focus. The ML algorithms investigated were NN and SVM in the former study and NN, RF and SVM in the latter study.
- Here we conduct several **large-scale simulation experiments** to broaden the existing knowledge on the subject and hopefully give rise to further research.
- We simulate time series according to stochastic models from the frequently used family of ARFIMA. Although this specific modelling is accompanied by certain problems (Koutsoyiannis 2016), it is considered rather satisfying for the present study and has been widely applied in the literature (e.g. Montanari et al. 1997).

3. Methodological framework

- We conduct **6** large-scale simulation experiments (SE_1a, SE_1b, SE_2a, SE_2b, SE_3a, SE_3b), which are determined by the simulated time series.
- We combine the following methodological elements:

✓ 3 x **2 000** time series of **110** values
✓ 3 x **2 000** time series of **310** values

✓ **12** ML methods: **4** NN methods, **4** RF methods, **4** SVM methods
✓ **2** benchmark methods

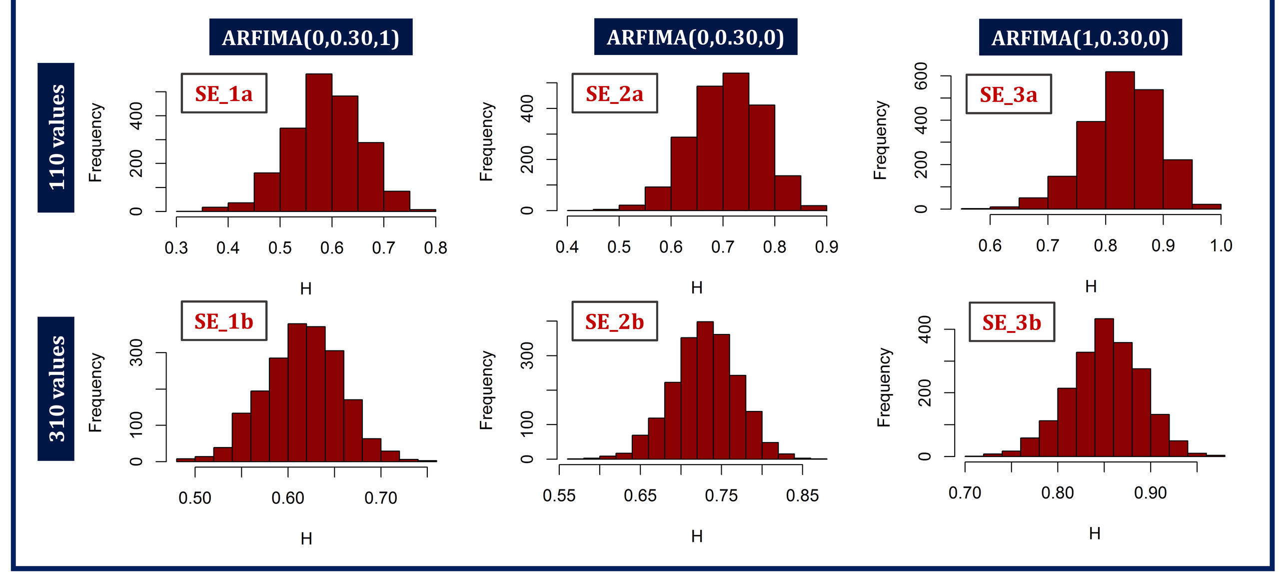


✓ **4** metrics providing assessment in respect to the following criteria: accuracy, capture of the variance, correlation

- Regarding the application of the forecasting methods, we split each simulated time series into a **fitting** and a **testing set** (last 10 values).
- We fit the models to the fitting set and compare their performance on the testing set using the values of the metrics.

4. Simulated time series

- We simulate time series according to the **ARFIMA(*p*,*d*,*q*)** model.
- We use the fracdiff.sim algorithm of the fracdiff R package (Fraley et al. 2012) to simulate the time series.
- To describe the long-term persistence of the simulated time series we estimate their **Hurst parameter *H*** (see Figures below) using the mleHK algorithm of the HKprocess R package (Tyralis 2016), which implements the maximum likelihood method (Tyralis and Koutsoyiannis 2011).
- The parameter *H* takes values in the interval (0,1). The larger it is the larger the long-range dependence of the Hurst - Kolmogorov stochastic process, which is widely used for the modelling of geophysical processes instead of the ARFIMA(0,*d*,0) model.



5. Forecasting methods

- We compare 12 ML methods. We also use 2 benchmarks in the comparisons.
- We apply the ML methods using the R package rminer (Cortez 2010, 2016), as well as several built in R algorithms (R Core Team 2017) and the benchmark methods using the R package forecast (Hyndman and Khandakar 2008, Hyndman et al. 2017).
- The R package rminer uses the nnet algorithm of the nnet R package (Venables and Ripley 2002), the randomForest algorithm of the randomForest R package (Liaw and Wiener 2002) and the ksvm algorithm of the kernlab R package (Karatzoglou et al. 2004) for the application of the NN, the RF and the SVM methods respectively.

hyperparameter optimized

	hyperparameter optimized
NN methods	number of hidden nodes
RF methods	number of variables randomly sampled as candidates at each split
SVM methods	sigma inverse kernel width

benchmark methods

Naive *last observation*	auto_ARFIMA

ML methods

	NN methods				RF methods				SVM methods			
	NN_1	NN_2	NN_3	NN_4	RF_1	RF_2	RF_3	RF_4	SVM_1	SVM_2	SVM_3	SVM_4
hyperparameter optimization	×	✓	×	✓	×	✓	×	✓	×	✓	×	✓
time lags used	1		1 to21		1		1 to 21		1		1 to 21	

6. Metrics

- Subsequently, we define the metrics used for the comparative assessment of the forecasting methods.
- For the definitions we consider a time series of N values. Let us also consider a model fitted to the first N - n values of this specific time series and subsequently used to make predictions corresponding to the last n values. Let $x_1, x_2, \ldots, x_n$ represent the last n values and $f_1, f_2, \ldots, f_n$ represent the forecasts.
- The **root mean square error (RMSE)** metric is defined by

$$\mathrm{RMSE} := \sqrt{(1/n) \sum_{i=1}^{n} (f_i - x_i)^2}$$

- Let $\bar{x}$ be the mean of $x_1, x_2, \ldots, x_n$ and s_x be the standard deviation of $x_1, x_2, \ldots, x_n$, which are defined by the following equations:

$$\bar{x} := (1/n) \sum_{i=1}^{n} x_i$$

$$s_x := \sqrt{(1/(n-1)) \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

- Let $\bar{f}$ be the mean of $f_1, f_2, \ldots, f_n$ and s_f be the standard deviation of $f_1, f_2, \ldots, f_n$, which are defined by the following equations:

$$\bar{f} := (1/n) \sum_{i=1}^{n} f_i$$

$$s_f := \sqrt{(1/(n-1)) \sum_{i=1}^{n} (f_i - \bar{f})^2}$$

- Then the **ratio of standard deviations (rSD)** metric is defined by

$$\mathrm{rSD} := s_f / s_x$$

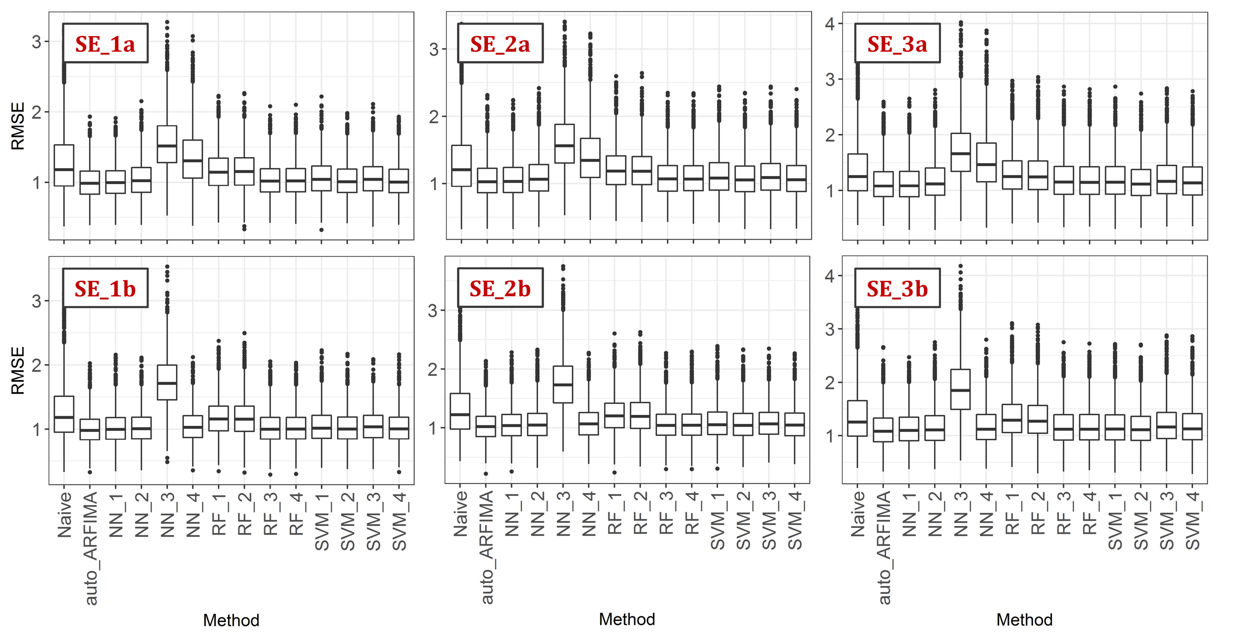
- The **Pearson's correlation coefficient (Pr)** metric is defined by

$$\mathrm{Pr} := \left(\sum_{i=1}^{n} (x_i - \bar{x})(f_i - \bar{f})\right) / \left(\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (f_i - \bar{f})^2\right)^{0.5}$$

- The **index of agreement (d)** metric is defined by

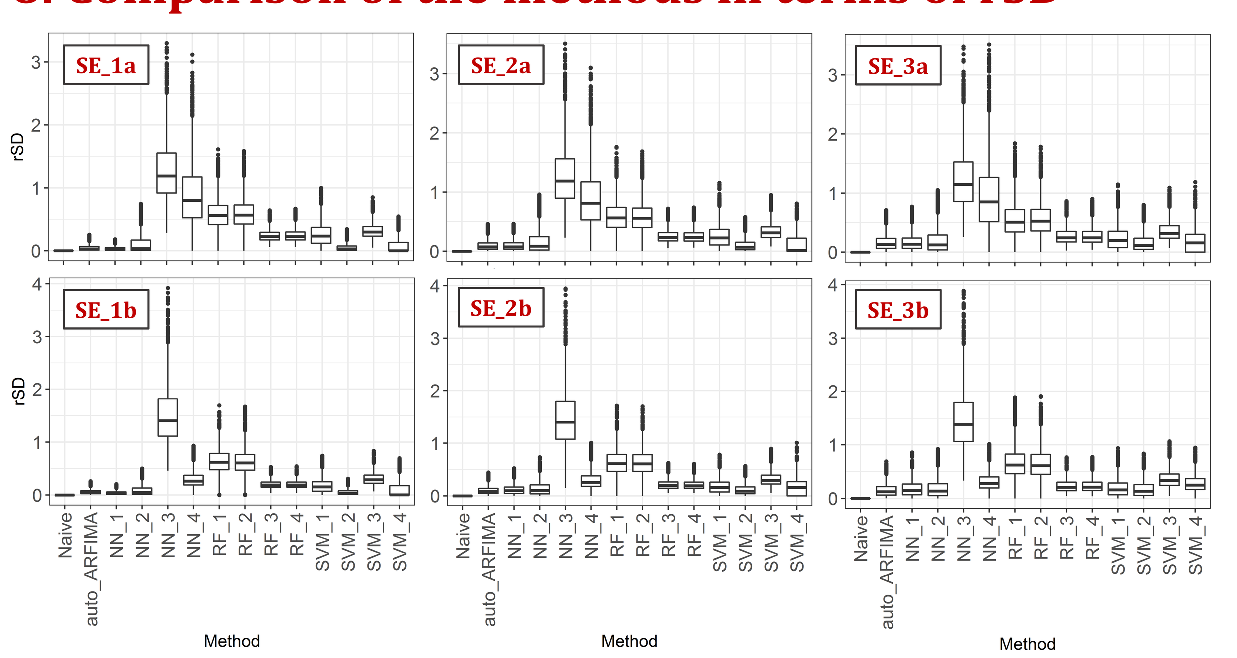
$$\mathrm{d} := 1 - \left(\sum_{i=1}^{n} (f_i - x_i)^2 / \sum_{i=1}^{n} (|f_i - \bar{x}| + |x_i - \bar{x}|)^2\right)$$

7. Comparison of the methods in terms of RMSE



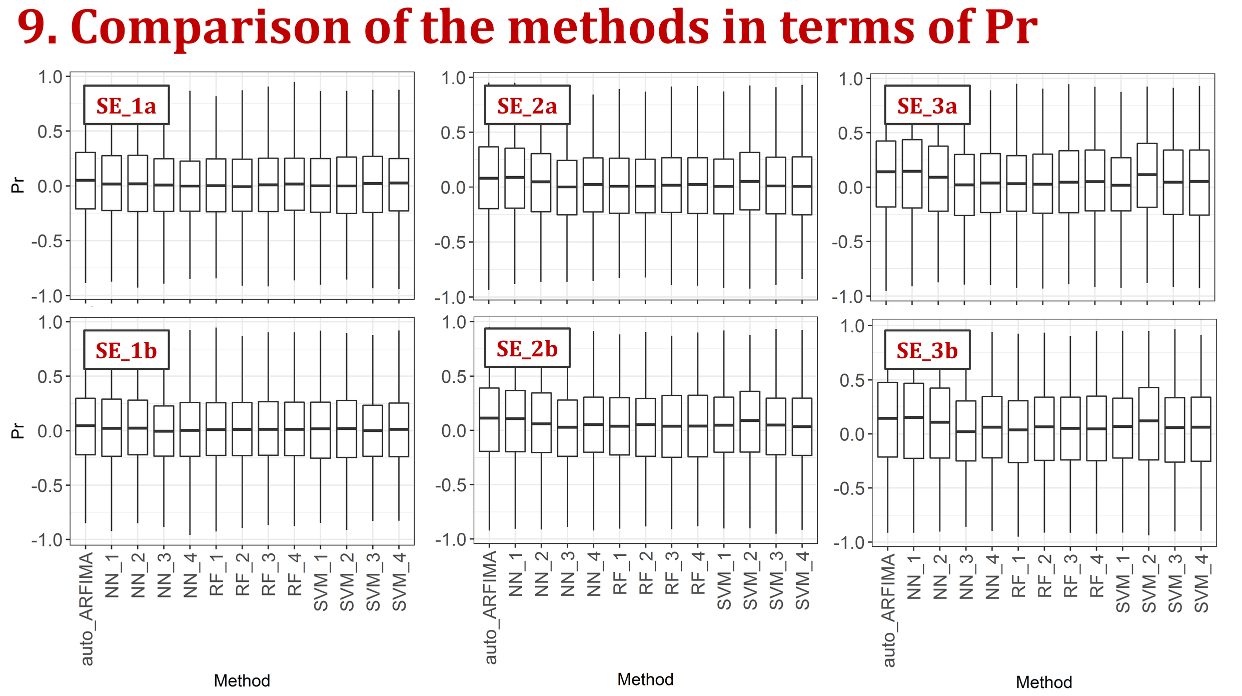
- The far outliers have been removed from the above figures, so that the latter are readable. In fact, the NN methods share a form of instability, which expresses itself by producing outliers, far above the median to an extent greater than the observed average.
- The simulated process does not affect the relative performance of the ML methods.
- The length of the time series mostly affect the performance of the NN_4 method.
- The hyperparameter optimization has mostly a small effect on the ML forecasting performance.
- The time lag selection seems to signifficantly affect the results when using the NN algorithm. We observe a similar behaviour for the RF algorithm albeit to a smaller extent.
- Naive can be better than NN_3 and NN_4, while most of the ML methods are as competent as auto_ARFIMA in terms of RMSE.

8. Comparison of the methods in terms of rSD



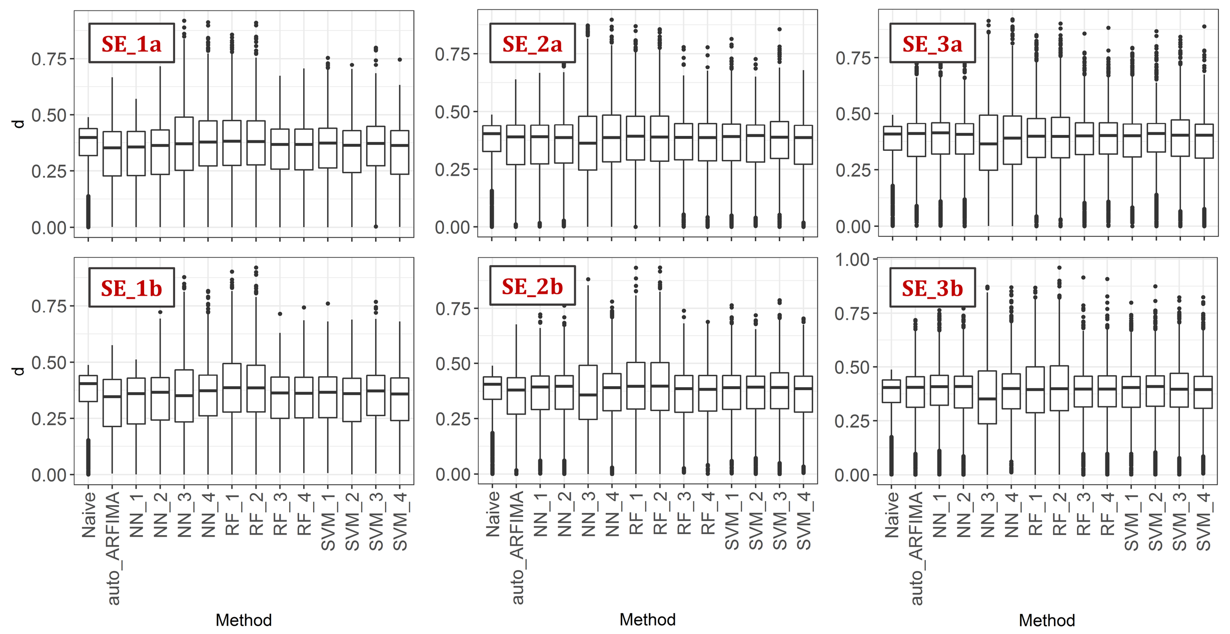
- The optimum rSD value is 1.
- The first five notes of **7** also apply to **8**.
- The image formed in terms of rSD is rather the opposite than the one formed in terms of RMSE for most of the ML methods compared, a fact we expected (see Papacharalampous 2016, Papacharalampous et al. 2017a).
- Most of the ML methods exhibit better performance than the auto_ARFIMA benchmark in terms of rSD.
- The performance of the ML methods is graded to a greater extent in terms of rSD than it does in terms of RMSE.

9. Comparison of the methods in terms of Pr



- The optimum Pr value is 1.
- The picture formed in terms of Pr is rather smoother than those presented in **7** and **8**. In other words, the forecasting methods are less graded in respect to this specific metric.
- The relative performance of the ML methods is slightly different across the various simulation experiments.
- The medians are all close to zero, a fact indicating that **the forecasting performance of all the methods is poor**.
- The auto_ARFIMA is mostly better than the ML methods, but to a small extent.

10. Comparison of the methods in terms of d



- The optimum d value is 1.
- The relative performance of the forecasting methods can differ to a rather small extent across the various simulation experiments.
- The Naive benchmark has the best average case performance in terms of d. This is quite remarkable.
- The auto_ARFIMA benchmark can exhibit either better or worse performance than the ML methods depending on the simulation experiment and the ML method.

11. Contribution of the present study

- The findings indicate that the hyperparameter optimization mostly has a small effect on the forecasting performance when using the machine learning algorithms of the present study. This is particularly important, because the hyperparameter optimization is computationally intensive.
- On the contrary, the time lag selection seems to mostly affect the machine learning methods' performance when using the Neural Networks algorithm, while we observe a similar behaviour for the Random Forests algorithm albeit to a smaller extent.
- Of course, the present study does not fill the existing gap in the literature and further research is essentially required. Nevertheless, it provides some evidence that these two factors do not necessarily affect the machine learning performance or that their effect is strongly dependent on the machine learning algorithm and the criterion of interest.
- In fact, the metrics employed in the evaluation process provide measurements which lead to different aspects of the same information to an extent that we cannot decide on a general ranking of the machine learning methods.
- Furthermore, the findings suggest that the machine learning methods do not differ dramatically from the stochastic benchmarks, while particularly noteworthy is the fact that the last observation benchmark is also competent.
- Admittedly, the results of the present study are entirely consistent with Papacharalampous (2016) and Papacharalampous et al. (2017a), while the limitations accompanying time series forecasting emphasized by the latter studies, as well as by Koutsoyiannis et al. (2008) and Papacharalampous et al. (2017b), are evident here as well.
- Perhaps research should invest more on probabilistic forecasting (e.g. Tyralis and Koutsoyiannis 2014) and its effective exploitation by the users (e.g. Ramos et al. 2013) rather than on point forecasting.

References

Cortez, P., 2010. Data Mining with Neural Networks and Support Vector Machines Using the R/rminer Tool. In: P. Perner, eds. Advances in Data Mining. Applications and Theoretical Aspects. Springer Berlin Heidelberg, pp 572-583. doi:10.1007/978-3-642-14400-4_44

Cortez, P., 2016. rminer: Data Mining Classification and Regression Methods. R package version 1.4.2.

Fraley, C., Leisch, F., Maechler, M., Reisen, V., and Lemonte, A., 2012. fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models. R package version 1.4-2.

Hyndman, R.J., and Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. Journal of Statistical Software, 27 (3), 1-22. doi:10.18637/jss.v027.i03

Hyndman, R.J., O'Hara-Wild, M., Bergmeir, C., Razbash, S., and Wang, E., 2017. forecast: Forecasting functions for time series and linear models. R package version 8.0.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A., 2004. kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software, 11 (9), 1-20.

Koutsoyiannis, D., 2016. Generic and parsimonious stochastic modelling for hydrology and beyond. Hydrological Sciences Journal, 61 (2), 225-244. doi:10.1080/02626667.2015.1016950

Koutsoyiannis, D., Yao, H., and Georgakakos, A., 2008. Medium-range flow prediction for the Nile: a comparison of stochastic and deterministic methods. Hydrological Sciences Journal, 53 (1), 142-164. doi:10.1623/hysj.53.1.142

Liaw, A., and Wiener, M., 2002. Classification and regression by randomForest. R News, 2 (3), 18-22.

Montanari, A., Rosso, R., and Taqqu, M.S., 1997. Fractionally differenced ARIMA models applied to hydrologic time series: Identification, estimation, and simulation. Water Resources Research, 33 (5), 1035-1044. doi:10.1029/97WR00043

Papacharalampous, G.A., 2016. Theoretical and empirical comparison of stochastic and machine learning methods for hydrological processes forecasting. MSc thesis.

Papacharalampous, G.A., Tyralis, H., and Koutsoyiannis, D., 2017a. Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes. Journal of Hydrology. In review.

Papacharalampous, G.A., Tyralis, H., and Koutsoyiannis, D., 2017b. Forecasting of geophysical processes using stochastic and machine learning algorithms. Proceedings of the 10th World Congress of EWRA on Water Resources and Environment.

Papacharalampous, G.A., Tyralis, H., and Koutsoyiannis, D., 2017c. Multi-step ahead streamflow forecasting for the operation of hydropower reservoirs. European Geosciences Union General Assembly 2017, Geophysical Research Abstracts, 19, Vienna.

R Core Team, 2017. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Ramos, M.H., Van Andel, S.J., Pappenberger, F., 2013. Do probabilistic forecasts lead to better decisions?. Hydrology and Earth System Sciences, 17, 2219-2232. doi:10.5194/hess-17-2219-2013

Tyralis, H., 2016. HKprocess: Hurst-Kolmogorov Process. R package version 0.0-2.

Tyralis, H., and Koutsoyiannis, D., 2011. Simultaneous estimation of the parameters of the Hurst–Kolmogorov stochastic process. Stochastic Environmental Research and Risk Assessment, 25 (1), 21-33. doi:10.1007/s00477-010-0408-x

Tyralis, H., and Koutsoyiannis, D., 2014. A Bayesian statistical model for deriving the predictive distribution of hydroclimatic variables. Climate Dynamics, 42 (11), 2867-2883. doi:10.1007/s00382-013-1804-y

Venables, W.N., and Ripley, B.D., 2002. Modern Applied Statistics with S, fourth edition. New-York: Springer-Verlag. doi:10.1007/978-0-387-21706-2

Zhang, G.P., 2001. An investigation of neural networks for linear time-series forecasting. Computers & Operations Research, 28 (12), 1183-1202. doi:10.1016/S0305-0548(00)00033-2