Stochastic Environmental Research and Risk Assessment (2019) 33:481–514
https://doi.org/10.1007/s00477-018-1638-6

ORIGINAL PAPER

Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes

Georgia Papacharalampous[1] · Hristos Tyralis[2] · Demetris Koutsoyiannis[1]

Published online: 1 January 2019

Abstract
Research within the field of hydrology often focuses on the statistical problem of comparing stochastic to machine learning (ML) forecasting methods. The performed comparisons are based on case studies, while a study providing large-scale results on the subject is missing. Herein, we compare 11 stochastic and 9 ML methods regarding their multi-step ahead forecasting properties by conducting 12 extensive computational experiments based on simulations. Each of these experiments uses 2000 time series generated by linear stationary stochastic processes. We conduct each simulation experiment twice; the first time using time series of 100 values and the second time using time series of 300 values. Additionally, we conduct a real-world experiment using 405 mean annual river discharge time series of 100 values. We quantify the forecasting performance of the methods using 18 metrics. The results indicate that stochastic and ML methods may produce equally useful forecasts.

Keywords No free lunch theorem · Random forests · River discharge · Stochastic hydrology · Support vector machines · Time series

1 Introduction

1.1 Background information

The fundamental problem of statistically producing point forecasts of univariate time series by exploiting information from their past values only (hereafter "forecasting", unless specified differently) is of traditional interest to hydrological scientists (Yevjevich 1987). Right after the introduction of the currently classical Autoregressive Integrated Moving Average (ARIMA) models by Box and Jenkins (1968), Carlson et al. (1970) used several stationary models of this specific family, i.e., Autoregressive Moving Average (ARMA) models, to forecast the evolution of four annual time series of streamflow processes. Today the available models for time series forecasting are numerous and can be classified according to De Gooijer and Hyndman (2006) into eight categories, i.e., (a) exponential smoothing, (b) ARIMA, (c) seasonal models, (d) state space and structural models and the Kalman filter, (e) nonlinear models, (f) long-range dependence models, e.g., the family of Autoregressive Fractionally Integrated Moving Average (ARFIMA) models, (g) Autoregressive Conditional Heteroscedastic/Generalized Autoregressive Conditional Heteroscedastic (ARCH/GARCH) models and (h) count data forecasting. The models from the categories (a)–(g) are of potential interest in hydrology, while they can be implemented for both one- and multi-step ahead forecasting.

The theoretical properties of the models of categories (a)–(d), (f), (g) (hereafter referred to as "stochastic") have been more or less investigated, in contrast to those of the nonlinear models and in particular the Machine Learning

✉ Georgia Papacharalampous
papacharalampous.georgia@gmail.com

Hristos Tyralis
montchrister@gmail.com

Demetris Koutsoyiannis
dk@itia.ntua.gr

[1] Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens, Iroon Polytechniou 5, 157 80 Zografou, Greece

[2] Air Force Support Command, Hellenic Air Force, Elefsina Air Base, 192 00 Elefsina, Greece

(ML) algorithms, also referred to in the literature as "black-box models". These two main categories of models are known to represent two different cultures in statistical modelling, i.e., the data modelling culture and the algorithmic modelling culture (Breiman 2001b). The former assumes that an analytically formulated stochastic model is behind the generation of the data, while the latter that behind this process is something complex and unknown, which does not have to be analytically formulated, as long as a purely algorithmic model can offer high forecast accuracy. In other words, profoundly understanding and properly modelling the (future) behaviour of a process are strongly connected within the data modelling culture, but completely irrelevant within the algorithmic modelling culture. The distinction between causal explanation, prediction and description is acknowledged and clarified in terms of modelling in Shmueli (2010). Still, one could question whether the (rather artificial) separation of models with respect to the "stochastic-ML dipole" actually corresponds to a striking difference in their forecasting performance.

What cannot be questioned, on the other hand, is the popularity that the various ML forecasting methods have gained in many scientific fields, including hydrology. Amongst the most popular ML algorithms are the Neural Networks (NN), Random Forests (RF) and Support Vector Machines (SVM). The latter two algorithms are presented in their current forms in Breiman (2001a), and Cortes and Vapnik (1995; see also Vapnik 1995, 1999) respectively. For the implementation of NN for time series forecasting the reader is referred to Zhang et al. (1998) and Zhang (2001), while a review of SVM forecasting applications can be found in Sapankevych and Sankar (2009). The large number of hydrological studies implementing NN and SVM forecasting methods is imprinted in Maier and Dandy (2000), and Raghavendra and Deka (2014) respectively. Moreover, Abrahart et al. (2012) collectively review the NN streamflow forecasting and rainfall-runoff applications (see, e.g., De Vos 2013). A major difference between these two families of applications is the use of exogenous variables in the latter. In contrast to NN and SVM, RF are barely utilized for hydrological process forecasting.

To explore the related background and facilitate the following discussion, in Table 1 we present some literature information on hydrometeorological time series forecasting emphasizing a few key aspects and concepts. As it is apparent, hydrological research often focuses on ML or hybrid (e.g., combinations of ARMA and ML) forecasting methods and, in particular, on the comparison between stochastic (mainly ARMA and ARFIMA) and ML methods. However, the culture of assessing the performance of forecasting methods on large datasets is not customary in hydrology. Therefore, the assessment is made within case studies. Concerning the testing procedure, while the available forecast quality metrics are a lot, most of the studies use only a few (Krause et al. 2005), understating the importance of the testing process despite relevant suggestions (see, e.g., Armstrong 2001; Abrahart et al. 2008; Humphrey et al. 2017). Likewise, the number of the compared forecasting methods is usually small and simple benchmarks are rarely included in the comparisons, although their use is highly recommended in the (hydrological) forecasting literature (see, e.g., Harvey 1984; Pappenberger et al. 2015; Hyndman and Athanasopoulos 2018, chapter 3.1).

Researchers have long been chasing the most accurate forecast for their data, a "universally best technique". On the other hand, there is an argument that it is the data and the application of interest that determine the proper methodology for each case, rather than vice versa (Hong and Fan 2016). Another argument is that perhaps research should invest more on probabilistic forecasting (e.g., using Bayesian statistics, as in Tyralis and Koutsoyiannis 2014) and less on point forecasting (Krzysztofowicz 2001). In fact, the opinions on forecast evaluation are often diverging, as they tend to depend on the perspective from which the forecasts are examined. An interesting study on this subject can be found in Murphy (1993). The latter identifies three criteria for this specific evaluation, which are adopted as a foundation for further discussion in later studies (e.g., Ramos et al. 2010; Weijs et al. 2010). These criteria are (1) the consistency during the forecasting process, (2) the quality or the correspondence between the forecasts and the target values, and (3) the value or the profit that the forecast provide to the decision makers. Weijs et al. (2010) note that criterion (2) concerns more the pure science, while criterion (3) is closer related to the decisions made within the engineering applications (of science), rather than science itself. Thus, only a few studies are dedicated to criterion (3), such as Ramos et al. (2010) and Ramos et al. (2013), while the greatest part of the literature focuses on criterion (2). The latter likewise largely applies to the present study and to all of its references aiming to deal with the modelling issue (*which model should I use?*) within specific hydrological concepts. Another criterion of practical importance is the computational (running) time required for obtaining the forecasts. This information might be significant depending on the forecasting task, while it could also be decisive, especially when one has to select between methods producing equally useful forecasts. The computational requirements are known to depend on the primary algorithm and its complexity, as well as on its software implementation, while the computational time also depends on the computer.

Regarding the so far conducted comparisons between forecasting methods, their majority in all scientific fields is

Table 1 Case studies presenting forecasts of hydrometeorological processes

S/n	Study	Primary focus	Hydrometeorological process				Data level				Horizon		
			Temperature	Precipitation	Streamflow or river discharge	Other	Hourly	Daily	Monthly	Annual	One-step ahead	Multi-step ahead	Not clear
1	Atiya et al. (1999)	NN methods	×	×	✔	×	×	×	✔	×	×	✔	×
2	Lambrakis et al. (2000)		×	×	✔	×	✔	×	×	×	✔	×	×
3	Kişi (2007)		×	×	✔	×	×	✔	×	×	✔	✔	×
4	Cheng et al. (2008)		×	×	✔	✔	×	×	✔	✔	✔	✔	×
5	Yaseen et al. (2016)		×	×	✔	×	×	✔	✔	×	✔	×	×
6	Sivapragasam et al. (2001)	SVM methods	×	✔	✔	×	×	✔	×	×	✔	×	×
7	Shi and Han (2007)		×	×	✔	✔	×	×	✔	✔	✔	✔	×
8	Lu and Wang (2011)		×	✔	×	×	×	✔	×	×	✔	×	×
9	Hu et al. (2001)	Hybrid methods	×	✔	×	×	×	×	×	✔	✔	×	×
10	Kim and Valdés (2003)		×	×	×	✔	×	×	✔	×	✔	✔	×
11	Pai and Hong (2007)		×	✔	×	×	✔	×	×	×	✔	×	×
12	Hong (2008)		×	✔	×	×	✔	×	×	×	✔	×	×
13	Kişi and Cimen (2011)		×	×	✔	×	×	×	✔	×	✔	×	×
14	Liong and Sivapragasam (2002)	SVM versus NN methods	×	×	×	✔	×	✔	×	×	✔	✔	×
15	Guo et al. (2011)		×	×	✔	×	×	×	✔	×	×	×	✔
16	Kişi and Cimen (2012)		×	✔	×	×	×	✔	×	×	✔	×	×
17	He et al. (2014)		×	×	✔	×	×	✔	×	×	✔	×	×
18	Jain et al. (1999)	Stochastic versus ML methods	×	×	✔	×	×	×	✔	×	✔	×	×
19	Ballini et al. (2001)		×	×	✔	×	×	×	✔	×	✔	✔	×
20	Kişi (2004)		×	×	✔	×	×	×	✔	×	✔	✔	×
21	Khan and Coulibaly (2006)		×	×	×	✔	×	×	✔	×	✔	✔	×
22	Lin et al. (2006)		×	×	✔	×	×	×	✔	×	×	×	✔
23	Mishra et al. (2007)		×	×	×	✔	×	×	✔	×	✔	✔	×
24	Yu and Liong (2007)		×	×	✔	×	×	×	×	×	×	✔	×
25	Koutsoyiannis et al. (2008)		×	×	✔	×	×	×	✔	×	✔	×	×
26	Wang et al. (2009)		×	×	✔	×	×	×	✔	×	×	✔	×
27	Abudu et al. (2010)		×	×	✔	×	×	×	✔	×	✔	×	×
28	Kişi et al. (2012)		×	×	×	✔	×	✔	×	×	✔	✔	×
29	Shabri and Suhartono (2012)		×	×	✔	×	×	×	✔	×	✔	×	×

Table 1 continued

S/n	Study	Primary focus	Hydrometeorological process				Data level				Horizon		
			Temperature	Precipitation	Streamflow or river discharge	Other	Hourly	Daily	Monthly	Annual	One-step ahead	Multi-step ahead	Not clear
30	Valipour et al. (2013)		×	×	✓	×	×	×	✓	×	×	×	✓
31	Patel and Ramachandran (2015)		×	×	✓	×	×	×	✓	×	×	✓	×
32	Papacharalampous et al. (2017c)		✓	✓	×	×	×	×	✓	×	✓	✓	×

based on case studies. Nevertheless, in some few cases beyond hydrology the number of the examined real-world time series is quite large. These time series are realizations of several phenomena, which however are fundamentally different from being hydrological, and their examination includes concepts that are rather inappropriate in hydrological terms (e.g., paying attention to small quantitative differences in the forecasting performance of the methods). Examples of such studies can be found in Makridakis et al. (1987), Makridakis and Hibon (2000), and Ahmed et al. (2010), which examine 1001, 3003 and 1045 time series respectively. Within these studies a statistical analysis is performed and the results are presented accordingly. Furthermore, the literature includes two studies, specifically Zhang (2001) and Thissen et al. (2003), in which the performance of the methods is assessed on simulated time series from linear stochastic processes. The scale of the simulation experiment is small in both cases. Thissen et al. (2003) examine one long time series from the ARMA family, and Zhang (2001) examine eight stochastic processes from the ARMA family and 30 simulated time series for each stochastic process. The forecasting methods are ARMA models, NN and SVM in the former study, and ARMA models and NN in the latter study, while Makridakis et al. (1987), Makridakis and Hibon (2000), and Ahmed et al. (2010) do not focus their comparisons on the stochastic-ML dipole.

Admittedly, the studies mentioned in the previous paragraph pursue generalized results to greater extent than most of the available studies. However, the gap still remains. What specifically needs to be addressed is whether the stochastic-ML dipole actually corresponds to a clear difference in the forecasting performance of the methods, especially in the light of published studies, which claim that they found a technique better than others. Given the fact that each forecasting case is indisputably unique, this task would necessarily require the examination of a sufficiently large and representative sample of forecasting cases within the same (properly designed) methodological framework. Extensive simulations combined with statistical analysis and benchmarking (i.e., evaluation in comparison to standard approaches and/or theoretically expected outcomes) can constitute, nevertheless, a highly effective approach to solving the problem under discussion. In more detail, for the generalized comparison of stochastic and ML forecasting methods, a sufficient number of different and representative of the underlying phenomena time series could be used for the estimation of the expected performance of forecasting methods regarding several criteria of interest. The need of using simulated time series to assess the performance of forecasting methods is emphasized by forecasting experts (Bontempi 2013). The analytical approach in assessing the performance of ML

algorithms is not possible; therefore, the only alternative approach is using simulations. Apparently, the larger the scale of the simulation experiments, the more general would be the results. Real-world experiments of large scale could be used to complement the results of the simulation experiments in alignment with specific applications. Some suggestions for the design of large-scale comparisons and the incorporation of benchmarking into methodological frameworks are available in Alpaydin (2010) and Hothorn et al. (2005) respectively.

1.2 The present study

In the context described in the above section, we perform an extensive comparison between several stochastic and ML methods for the forecasting of hydrological processes by conducting large-scale computational experiments based on simulations. The comparison refers to the multi-step ahead forecasting properties of the methods. The simulated time series are 48,000 in total, while they are generated by linear stationary stochastic processes. The latter are commonly used for modelling hydrological processes. In fact, the linearity assumption starts to become reasonable when modelling hydrological variables at large time scales (e.g., annual or monthly), while at fine time scales (e.g., daily or hourly) non-linear modelling approaches start to prevail (e.g., due to intermittency). Moreover, stationary models, in contrast to the non-stationary ones, are established as the appropriate modelling choice when dealing with natural processes, unless tangible and quantitative information that can fully support a deterministic description (not based on data but on physical laws) of change in time is available (Koutsoyiannis 2011; Koutsoyiannis and Montanari 2015). Additionally to the simulation experiments, we examine 405 real-world time series. Our aim is to fill the gap detected in the literature by providing large-scale results and useful insights on the comparison of stochastic and ML forecasting methods for the case of hydrological time series forecasting at large time scales, with an emphasis on annual river discharge processes. A strength (and limitation) of the present study (implied by its aim) is the adopted approach to the problem, i.e., the algorithmic or data-driven approach.

The present study was first presented by Papacharalampous et al. (2017a), while a preliminary research on the subject was conducted for the Postgraduate Thesis of the first author (Papacharalampous 2016). Subsequently, we provide some basic information about the large-scale companion studies of this paper. Papacharalampous et al. (2017b) examine the problem of error evolution in hydrological multi-step ahead forecasting, while Tyralis and Papacharalampous (2017) improve the performance of RF in one-step ahead forecasting of geophysical processes. Papacharalampous et al. (2018a) also focuses on the problem of one-step ahead forecasting with the aim to provide large-scale results on the latter in geoscience. These three studies examine simulated, as well as real-world datasets. In detail, they examine 12,000 simulated and 92 monthly streamflow time series, 16,000 simulated and 135 annual temperature time series, and 24,000 simulated, 185 annual temperature and 112 annual precipitation time series respectively. Finally, Papacharalampous et al. (2018b) produce multi-step ahead forecasts for 985 monthly temperature and 1552 monthly precipitation time series aiming at the investigation of the predictability of these processes. All the time series examined by the present study and its companions are short, as it is expected for the hydrometeorological time series.

Makridakis et al. (2018) focus on a similar investigation to the present paper, although in a different field and under a different experimental setting.

2 Methodology

In Sect. 2 we present the basic methodological elements of this study and the way that these elements are combined into a framework for evaluating forecasting methods in hydrology. Software implementation information is also provided. All R functions are used as specified in this methodology overview. If no specification is made, then the default values are adopted. We note that the use of default values is acknowledged in the literature as a "reasonable and justified choice" in most cases (see, e.g., Arcuri and Fraser 2013). To ensure reproducibility, the R codes are available in the supplementary material (see "Appendix").

2.1 Simulated processes

We simulate time series according to several stochastic models from the frequently used families of ARMA and ARFIMA. This modelling approach is considered appropriate for the achievement of our aim and has been widely applied in hydrology (see, e.g., Montanari et al. 1997, 2000; Ballini et al. 2001; Wang et al. 2009; Valipour et al. 2013). The simulated stochastic processes are presented in Table 2, while for the related definitions the reader is referred to the report entitled "Definitions of the stochastic processes'' of the supplementary material (see also Wei 2006, pp. 6–87, 489–494). These 12 stochastic models correspond to different types of autocorrelation. We use the arima.sim built-in-R function (R Core Team 2018) to simulate the ARMA(p,q) processes and the fracdiff.sim function of the fracdiff R package (Fraley et al. 2012) to simulate the ARFIMA(p,d,q) processes.

Table 2 Simulated stochastic processes of the present study

S/n	Stochastic model	AR and/or MA parameters
1	AR(1)	$\varphi_1 = 0.7$
2	AR(1)	$\varphi_1 = -0.7$
3	AR(2)	$\varphi_1 = 0.7$, $\varphi_2 = 0.2$
4	MA(1)	$\theta_1 = 0.7$
5	MA(1)	$\theta_1 = -0.7$
6	ARMA(1,1)	$\varphi_1 = 0.7$, $\theta_1 = 0.7$
7	ARMA(1,1)	$\varphi_1 = -0.7$, $\theta_1 = -0.7$
8	ARFIMA(0,0.45,0)	
9	ARFIMA(1,0.45,0)	$\varphi_1 = 0.7$
10	ARFIMA(0,0.45,1)	$\theta_1 = -0.7$
11	ARFIMA(1,0.45,1)	$\varphi_1 = 0.7$, $\theta_1 = -0.7$
12	ARFIMA(2,0.45,2)	$\varphi_1 = 0.7$, $\varphi_2 = 0.2$, $\theta_1 = -0.7$, $\theta_2 = -0.2$

Their definitions are given in the supplementary material. The parameters μ and σ of the simulated stochastic processes are set to 0 and 1 respectively

2.2 Real-world time series

We examine 405 mean annual river discharge time series of 100 values, sourced from GRDC (2017). For the exploration of these time series we compute the sample Autocorrelation Function (ACF) and the sample Partial Autocorrelation Function (PACF). The side-by-side boxplots of the ACF and PACF estimates are presented in Fig. 1. The Hurst–Kolmogorov (HK) behaviour is a common property of geophysical properties (see, e.g., Tyralis and Koutsoyiannis 2011). To describe the HK behaviour of river discharge we estimate the Hurst parameter H of the HK process for each time series using the mleHK function of the HKprocess R package (Tyralis 2016). The latter 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 magnitude of the HK behaviour, which can be modelled by an ARFIMA(0,d,0) model. A histogram of the H estimates is presented in Fig. 1. By its examination we observe that the magnitude of the long-range dependence is mostly significant in the real-world time series.

2.3 Forecasting methods

We compare 11 stochastic and 9 ML forecasting methods. These methods are briefly presented in Sects. 2.3.1 and 2.3.2 respectively. The primary forecasting algorithms are well documented in the literature. Therefore, we place emphasis on their software implementation, while the compiled information from books, textbooks and journal articles is limited to their key concepts and their basic theoretical background. Further theoretical details, available in the provided references, are here omitted for reasons of brevity. We note that the understanding from a theoretical point of view of most methods could hardly help in interpreting the algorithmically obtained outcome of the comparison.

2.3.1 Stochastic methods

The stochastic methods are classified into five main categories, as presented in Table 3. Their implementation is mostly made through the forecast R package (Hyndman and Khandakar 2008; Hyndman et al. 2018). Specifically, we use the arfima, Arima, auto.arima, bats, ets, forecast, rwf, ses and thetaf functions of the forecast R package, and the simulate built-in-R function. The arfima and Arima functions use the fracdiff function of the fracdiff R package and the arima built-in-R function respectively. We implement two simple forecasting methods. The Naïve forecasting method simply sets all forecasts equal to the last value. The RW forecasting method, a variation of the Naïve forecasting method also known as Random Walk with drift, is equivalent to drawing a line between the first and the last values, and extrapolating it into the future (Hyndman and Athanasopoulos 2018, chapter 3.1).

ARIMA and ARFIMA methods are also included in the comparisons. For ARIMA_f and ARIMA_s the numbers of the AR and MA parameters (p and q respectively) are set to be the same to those used in the time series simulation process (see Sect. 2.1), while the number of differencing (d) is set to zero. On the contrary, auto_ARIMA_f and auto_ARIMA_s automatically estimate the order of the ARIMA models (and, therefore, the utilized lagged predictor variables) as summarized in the following. First, the d values are estimated via repeated Kwiatkowski–Phillips–Schmidt–Shin tests (Kwiatkowski et al. 1992). Once the d value has been obtained, the p and q values are estimated using a stepwise algorithm aiming at the minimization of

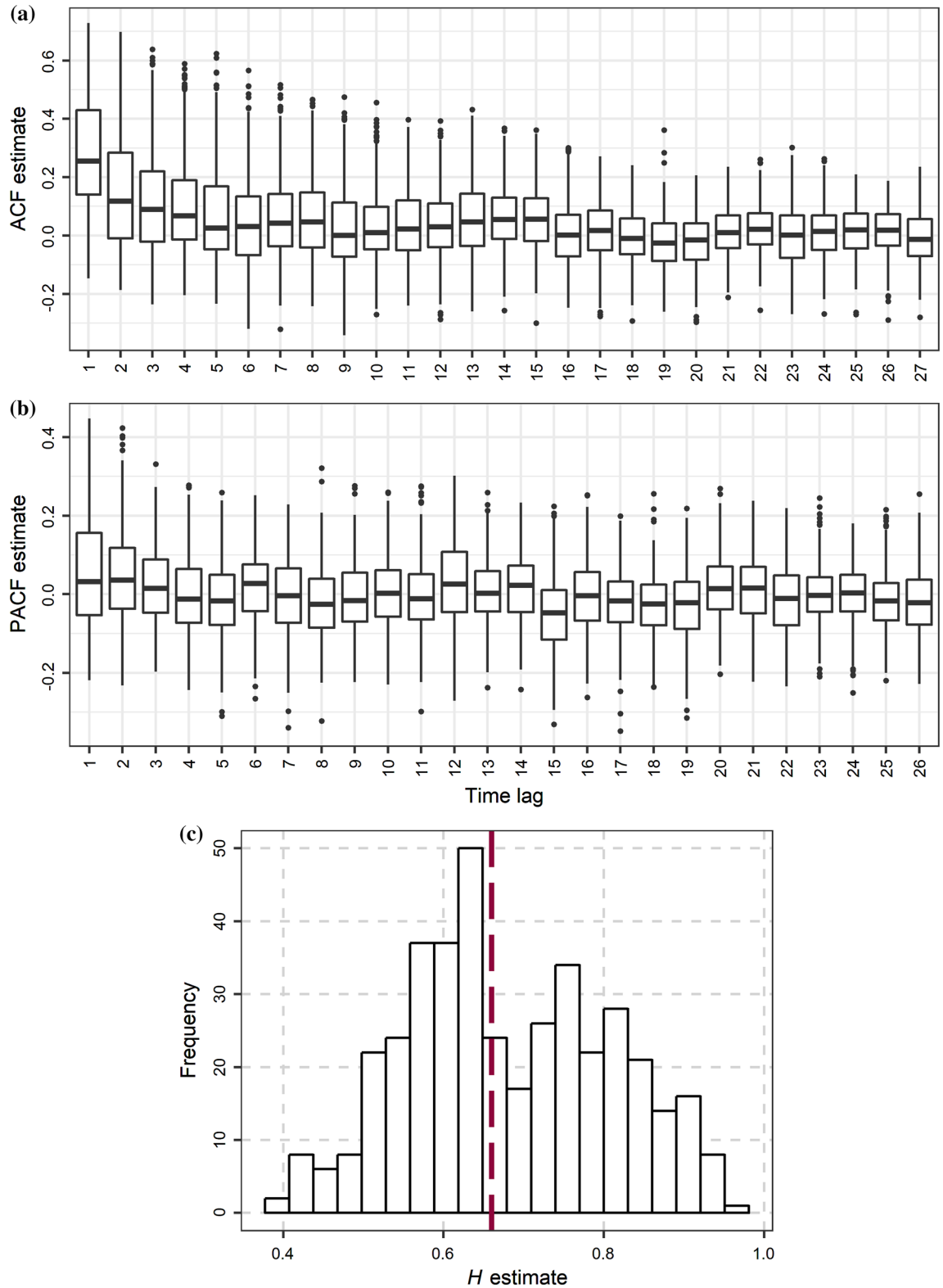

Fig. 1 **a** ACF, **b** PACF, **c** *H* estimates for the real-world time series. Data source: GRDC (2017). The red dashed line in **c** denotes the median of the *H* estimates

Table 3 Stochastic methods of the present study

S/n	Abbreviated name	Category
1	Naïve	Simple
2	RW	
3	ARIMA_f	ARIMA
4	ARIMA_s	
5	auto_ARIMA_f	
6	auto_ARIMA_s	
7	auto_ARFIMA	ARFIMA
8	BATS	Innovations state space
9	ETS_s	
10	SES	Exponential smoothing
11	Theta	

The forecasting methods are available in code form in the supplementary material

the Akaike Information Criterion with a correction for finite sample sizes (AICc). This information (or model discrimination) criterion is introduced in Hurvich and Tsai (1993). Ye et al. (2008) report on a debate in hydrology on the selection between commonly used information criteria, such as the original Akaike Information Criterion (AIC) by Akaike (1974), its corrected version (AICc), and two Bayesian information criteria, i.e., BIC by Schwarz (1978) and KIC by Kashyap (1982). Task-oriented comparisons of information criteria can be found, for instance, in Ye et al. (2004), Billah et al. (2005) and Ye et al. (2008). AICc is preferred in the herein adopted implementation by Hyndman et al. (2018), while other available options are AIC and BIC. We note that AICc reduces asymptotically to AIC as the sample increases (Ye et al. 2008), while for small training samples the minimization of AIC tends to lead to larger number of model parameters (and larger number of lagged predictor variables) compared to the minimization of AICc (Hyndman and Athanasopoulos 2018, chapter 5.5). The exact procedure adopted by auto_ARIMA_f and auto_ARIMA_s for order estimation is available in Hyndman and Khandakar (2008), and Hyndman and Athanasopoulos (2018, chapter 8.6). As explained in the latter-mentioned textbook's chapter, the d value is not estimated simultaneously with the p and q values using AICc, because in this case the estimation would be suboptimal. Once the p, d and q values have been estimated, the four ARIMA methods apply the maximum likelihood method to estimate the AR and MA model parameters (Hyndman and Athanasopoulos 2018, chapter 8.6). Similarly to auto_ARIMA_f and auto_ARIMA_s, auto_ARFIMA estimates d first, and thereupon follows a stepwise procedure to select p and q. Subsequently, it implements the algorithm of Haslett and Raftery (1989) to estimate the ARFIMA parameters. A final value of d is estimated as well in this last step. The latter information is sourced from Hyndman et al. (2018) and Fraley et al. (2012), where related detailed descriptions can be found. The definitions of the ARMA, ARIMA and ARFIMA models are given in the report entitled "Definitions of the stochastic processes'' of the supplementary material (see also Wei 2006, pp. 6–87, 489–494). It is essential to also note that ARIMA_s and auto_ARIMA_s are simulation models, while the innovations are set to zero by the ARIMA_f, auto_ARIMA_f and auto_ARFIMA methods, i.e., the forecasts produced by the latter three methods are the expected future values from the AR(F)IMA model selected during the training process.

Another family of stochastic methods considered herein (that is also broader than the family of ARIMA models; Gardner 2006) includes the Exponential Smoothing models and their underlying methods, i.e., the (Innovations) State Space methods for Exponential Smoothing. Their forecasts are weighted averages of past values, with the weights decaying exponentially as these values get distant in time (Hyndman and Athanasopoulos 2018, chapter 7). Informative reviews by Gardner (1985, 2006) discuss older and latest advances in forecasting with Exponential Smoothing, from the introducing works by Brown and Holt that are available in Brown (1959) and Holt (2004) respectively (the latter paper is a reprinted version of Holt's report of 1957) up to more recent studies (e.g., Assimakopoulos and Nikolopoulos 2000; Hyndman et al. 2002; Hyndman and Billah 2003). The reader is also referred to Hyndman et al. (2008), and Hyndman and Athanasopoulos (2018, chapters 7, 8.3) for further details on the theoretical background of the Exponential Smoothing and State Space models. We implement the Simple Exponential Smoothing (SES) and Theta methods. Similarly to the Naïve and average methods (the forecasts of the latter are simply the average of all training values), SES produces flat forecasts; therefore, it is considered the most simple of its class. An interpretation of the concept behind SES is provided by Hyndman and Athanasopoulos (2018, chapter 7.1). According to this interpretation, SES is a more general version of both the Naïve and average methods. Furthermore, the Theta method by Assimakopoulos and Nikolopoulos (2000) is equivalent to SES with drift (Hyndman and Billah 2003). There are several variations of Theta, each defined by the so-called "Theta lines", i.e., the auxiliary time series (modified versions of the original time series provided as input to the method) used for model fitting and forecasting. A Theta line is characterized by its local curvature, which is determined by the Theta coefficient θ (different for each Theta line). Extrapolations of all Theta lines are averaged to produce the forecast. We implement the version of Theta that performed well in the M3 competition (Makridakis and Hibon 2000), i.e., the one defined by two Theta lines,

Table 4 Machine learning methods of the present study

S/n	Abbreviated name	Category	Basic model information	Hyperparameter optimized (grid values)	Time lag selection procedure
1	NN_1	NN	Single-hidden-layer MLP	Number of hidden nodes (0, 1, …, 15)	1
2	NN_2				2
3	NN_3				3
4	RF_1	RF	Breiman's random forests algorithm with 500 grown trees	Number of variables randomly sampled as candidates at each split (1, …, 5)	1
5	RF_2				2
6	RF_3				3
7	SVM_1	SVM	Radial Basis kernel "Gaussian" function, $C = 1$, $\varepsilon = 0.1$	Sigma inverse kernel width (2^n, $n = -8, -7, \ldots, 6$)	1
8	SVM_2				2
9	SVM_3				3

The time lag selection procedures adopted are defined in Table 5. The forecasting methods are available in code form in the supplementary material

Table 5 Time lag selection procedures adopted for the machine learning methods

S/n	Time lags
1	The corresponding to an estimated value for the ACF using the acf built-in-R function, i.e., the time lags 1, …, 19 for a time series of 90 values and the time lags 1, …, 24 for a time series of 290 values
2	The corresponding to a statistically significant estimated value for the ACF using the acf built-in-R function. If there is no statistically significant estimated value for the ACF, the corresponding to the largest estimated value
3	According to the nnetar R function of the forecast R package, i.e., the time lags 1, …, k, where k is equal to the maximum between 1 and the number of parameters of an AR model fitted to the time series data using the ar built-in-R function. The optimal number of AR parameters is decided using the AIC

The forecasting methods are available in code form in the supplementary material

specifically for $\theta = 0$ and $\theta = 2$ (see Assimakopoulos and Nikolopoulos 2000). Moreover, we implement two State Space methods for Exponential Smoothing. Models from this category produce expected value forecasts and, additionally, provide information about the forecast error variances (Hyndman et al. 2005; see also Hyndman and Athanasopoulos 2018, chapter 7.5). This information can be used either for constructing prediction intervals or for running an Exponential Smoothing model in simulation mode. The latter case applies to the ETS_s method. This method also comprises automatic selection of the Error, Trend and Seasonal components (ETS) using the AICc (Hyndman and Athanasopoulos 2018, chapter 7.6). The expected value forecasts of this model on the M competition and M3 competition data are found to be comparable with the best obtained in these competitions (Hyndman et al. 2002). Another State Space method implemented herein is BATS. This method uses the point forecasts from an Exponential Smoothing State Space model with several key features, i.e., capability of performing Box-Cox transformation and/or including ARMA errors correction, Trend and Seasonal components (BATS), also allowing an optimal model selection using the Akaike Information Criterion (AIC). The original model is introduced and fully documented in De Livera et al. (2011).

2.3.2 Machine learning methods

The ML methods are classified into three main categories, as presented in compact form in Tables 4 and 5. We implement these methods mainly by using the CasesSeries, fit and lforecast functions of the rminer R package (Cortez 2010, 2016) and the nnetar function of the forecast R package (the latter is the NN_3 forecasting method), as well as several built-in-R functions. The rminer R package uses the nnet function of the nnet R package (Venables and Ripley 2002; Ripley 2016), the randomForest function of the randomForest R package (Liaw and Wiener 2002; Liaw 2018) and the ksvm function of the kernlab R package (Karatzoglou et al. 2004, 2018) for the implementation of the NN, RF and SVM methods respectively. The nnetar function also uses the nnet function.

The training of the ML methods is traditionally based on (mostly) different strategies than those discussed in Sect. 2.3.1. The input to a ML method is the data matrix used in the regression process (hereafter referred to as

"input data matrix"). The latter is built using a single time series holding the total information provided to the ML method. One column of the input data matrix holds information about the predictand variable and the remaining columns information about lagged (predictor) variables that are assumed to be informative about the predictand. Variable selection (or feature selection) is known as a factor that might affect the performance of ML algorithms in both regression and forecasting applications (see, e.g., Anctil et al. 2009; Papacharalampous et al. 2017c). Thus, many studies specifically focus on the examination of this problem (e.g., Kohavi and John 1997; Tyralis and Papacharalampous 2017). A usual practice in the literature, also adopted herein, is to use a priori determined lagged variables and place emphasis on hyperparameter optimization during the training process (see, e.g., the implementations by Khan and Coulibaly 2006; Lin et al. 2006; Wang et al. 2009). Hyperparameters are parameters that can be optimized (or tuned) to limit overfitting (known to deteriorate the forecasting performance of an algorithm), thereby improving the performance of a ML algorithm (Witten et al. 2017, pp. 171–172). This specific utility of hyperparameters justifies their artificial distinguishment from the parameters of the stochastic models and the basic parameters of the ML models. Several examples of hyperparameters can be found in Luo (2016). A common approach to hyperparameter optimization is the herein implemented automatic grid search (Hutter et al. 2015). In optimization via grid search a complicated optimization problem is solved as the simplified problem of selecting between several candidate model configurations during the training process. The candidate configurations are defined by different predetermined hyperparameter values (Witten et al. 2017, pp. 171–172). The considered hyperparameter values and the adopted procedures for selecting the time lag(s) (one at minimum) are reported in Tables 4 and 5 respectively, while some supporting information to the former table are provided subsequently.

NN are an ensemble approach to regression and, by extension, to forecasting (Hastie et al. 2009, pp. 623), often perceived to mimic the human brain's function. Detailed information about NN is available, for instance, in Lippmann (1987), Murtagh (1991), Lanc (1992, pp. 7–28), Zhang et al. (1998), Hastie et al. (2009, pp. 389–416), Marsland (2011, pp. 71–110), and Hyndman and Athanasopoulos (2018, chapter 11.3), while the below synopsis of this information is largely adapted to our computations. We utilize a single-hidden-layer Multilayer Perceptron (MLP), which consists of interconnected computational units known as nodes or neurons grouped into three layers, namely the input, hidden and output layers. The employed MLP is feed-forward, i.e., the information moves in one direction, specifically from the input nodes to the output nodes through the hidden nodes. This information transit is achieved via (weighted) connections, while all computations are performed in the nodes. The input nodes are inactive, i.e., they do not apply any transfer function (e.g., a sigmoid function) to their inputs before passing them forward, while each of the hidden and output nodes computes the (weighted) sum of its inputs and subsequently applies a transfer function (usually different for the two layers) to this sum. In fact, each group of nodes has its own characteristics that are related to its utility. The number of input nodes is simply the number of lagged variables or the number of time lags. Moreover, the number of output nodes is set to be one, even for multi-step ahead forecasts, since the latter are produced iteratively using one-step ahead predictions as inputs (Cortez 2016; Hyndman and Athanasopoulos 2018, chapter 11.3). The number of hidden nodes, on the other hand, is an (integer-valued) hyperparameter to be optimized during the training process. The candidate architecture configurations are 16 in number. They are defined by fixed numbers of layers, input and output nodes according to the above-outlined information, and 16 different possibilities for the number of hidden nodes according to Table 4. Zero number of hidden nodes and, consequently, no hidden layer is a feasible option within our experiments.

RF can also be considered as ensemble methods (Hastie et al. 2009, pp. 605; Scornet et al. 2015). Herein we use the original RF algorithm by Breiman (2001a), i.e., an evolution of the bagging algorithm by Breiman (1996) applied to regression trees (Liaw and Wiener 2002). The term BAGGING is an acronym for Bootstrap AGGregatING (Breiman 1996). Bagging or bootstrap aggregation is an iterative scheme for building a large number of individual predictors by sampling from the input dataset to finally aggregate the results obtained by them to get the prediction of interest (Biau 2012; Scornet et al. 2015; Biau and Scornet 2016). For continuous variables, the aggregation is made by computing the average of all values obtained by bagged predictors (Sutton 2005; Moisen 2008). This averaging reduces the variance of an estimated prediction function leading to more accurate predictions (Sutton 2005; Hastie et al. 2009, pp. 282–288). Nonetheless, the reduction in variance is limited by large correlation values between pairs of bagged predictors. RF are designed to dominate their precursor by offering a further improvement in terms of variance reduction. This improvement is achieved by reducing the correlation between the tree-structured predictors through random selection of the input variables in the tree-growing process (Hastie et al. 2009, pp. 587–588). We grow 500 randomized regression trees per model run. We grow each tree using a different bootstrap sample drawn from the input data matrix. The sampling is made with replacement. The procedure followed to create each tree is fully described, for example, in Hastie

et al. (2009, pp. 588), and Biau and Scornet (2016). The hyperparameter optimized is the number of variables randomly sampled as candidates at each split (integer-valued hyperparameter) during the tree-growing process, while the candidate configurations to choose from during the training process are five. In Tyralis and Papacharalampous (2017) we describe the Breiman's RF algorithm in greater detail.

A to-the-point summary of SVM is available in Solomatine and Ostfeld (2008), while Hastie et al. (2009, pp. 417–438) review the theoretical background of these models, and Smola and Schölkopf (2004) provide an overview of their underlying idea with an emphasis on regression and forecasting problems. In contrast to NN and RF that can be conceptualized as structured models with fixed and random architecture respectively (see the above paragraphs), SVM are usually perceived as models utilizing a hyperplane for the separation in a two-dimensional space of two different classes in classification (see, e.g., Solomatine and Ostfeld 2008). They are introduced in Cortes and Vapnik (1995) as an extension of the Vapnik's method of optimal hyperplanes. This method is applicable to separable training data, i.e., training data that can be separated without errors, while SVM can be implemented on non-separable training data as summarized subsequently. The input vectors are non-linearly mapped into a high-dimensional feature space, where the hyperplanes are linearly constructed in a way pursuing generalizable (to unobserved situations) solutions. The optimal separating hyperplane is defined as the one that maximizes the margin between the classes in the separable case, and as the one that simultaneously minimizes the number of errors and separates with maximal margin the correctly classified elements in the non-separable case (Smola and Schölkopf 2004). The optimization problem to be solved in regression is a convex optimization problem defined as follows. The objective is to find a function f that simultaneously is as flat as possible and deviates less or equal to ε from all input data values. In cases where this problem is not solvable or we want to allow some errors, the formulation changes so that there is a predefined trade-off between the flatness of f and deviations larger than ε. This trade-off is determined by a constant $C > 0$ (Smola and Schölkopf 2004). Herein, we use the default kernel function and the default C and ε values, and optimize sigma inverse kernel width (continuous hyperparameter) during the training process according to Table 4. Sigma inverse kernel width is a hyperparameter to be specified when using the Radial Basis and the Laplacian kernel functions for the computations in the feature space (Karatzoglou et al. 2004).

2.4 Forecast quality metrics

We utilize the forecast quality metrics briefly presented in Table 6. These metrics do not share one-to-one relationships with each other, emphasizing—more or less—different aspects of the same information. Their classification into six main categories according to the criterion/criteria that is/are (co-)assessed through their use is also presented in Table 6. These criteria are two types of accuracy, the capture of the variance and the correlation. By type 1 accuracy we mean the closeness of the forecasted time series to the target time series, while by type 2 accuracy we mean the closeness of the mean of the forecasts to the mean of the target values. The definitions of the forecast quality metrics are listed in the report entitled "Definitions of the forecast quality metrics" of the supplementary material, while in the below paragraphs we justify their combined use herein.

MAE provides an easily interpretable assessment with respect to the type 1 accuracy criterion, while it is also amongst the most frequently used forecast quality metrics (Hyndman and Koehler 2006). Similarly, the computation of MAPE and RMSE is implied by their traditional use in the forecasting field (Armstrong and Collopy 1992; Hyndman and Koehler 2006). Although RMSE is more sensitive to outliers than MAE (Fildes 1992; Hyndman and Koehler 2006), the former is usually preferred to the latter by forecasting scientists mainly because of its "theoretical relevance in statistical modelling" (Hyndman and Koehler 2006). Furthermore, MAPE is a scale-independent metric, offering an advantage in comparing forecasting methods across different datasets. Nonetheless, this metric is particularly affected by target values close to zero (Fildes 1992; Hyndman and Koehler 2006). The ME and MPE metrics are also utilized herein as they constitute analogues (with similar advantages and disadvantages) to MAE and MAPE respectively for the assessment according to the type 2 accuracy criterion.

Some limitations of the correlation metrics, i.e., the Pr and r2 ones, mainly related to an over-sensitivity to outliers and to the fact that their optimum value does not indicate by itself a perfect forecast, are well understood in hydrology and beyond (see, e.g., Legates and McCabe 1999; Armstrong 2001). However, their use is of traditional significance (Legates and McCabe 1999; Krause et al. 2005) and could not harm the interpretation of the results, when these metrics are used attentively and collectively with others (Krause et al. 2005). Perhaps the most widely used metric in the field of hydrology is the introduced by Nash and Sutcliffe (1970) NSE, while another traditional metric is d (Legates and McCabe 1999; Krause et al. 2005; Schaefli and Gupta 2007). Consequently, these two metrics

Table 6 Forecast quality metrics used in the present study

S/n	Abbreviated name	Full name	Criterion/criteria	Values	Optimum value	Condition (preferred values)
1	MAE	Mean absolute error	Type 1 accuracy	$[0, +\infty)$	0	Smaller MAE
2	MAPE	Mean absolute percentage error		$[0, +\infty)$	0	Smaller MAPE
3	RMSE	Root mean square error		$[0, +\infty)$	0	Smaller RMSE
4	NSE	Nash–Sutcliffe efficiency		$(-\infty, 1]$	1	Larger NSE
5	mNSE	Modified Nash–Sutcliffe efficiency		$(-\infty, 1]$	1	Larger mNSE
6	rNSE	Relative Nash–Sutcliffe efficiency		$(-\infty, 1]$	1	Larger rNSE
7	cp	Persistence index		$(-\infty, 1]$	1	Larger cp
8	ME	Mean error	Type 2 accuracy	$(-\infty, +\infty)$	0	Smaller |ME|
9	MPE	Mean percentage error		$(-\infty, +\infty)$	0	Smaller |MPE|
10	PBIAS	Percent bias		$(-\infty, +\infty)$	0	Smaller |PBIAS|
11	VE	Volumetric efficiency		$(-\infty, +\infty)$	1	Smaller $\|VE - 1\|$
12	rSD	Ratio of standard deviations	Capture of the variance	$[0, +\infty)$	1	Larger min{rSD, 1/rSD}
13	Pr	Pearson' s correlation coefficient	Correlation	$[-1, 1]$	1	Larger Pr
14	r2	Coefficient of determination		[0, 1]	1	Larger r2
15	*d*	Index of agreement	Type 1 accuracy, capture of the variance	[0, 1]	1	Larger *d*
16	md	Modified index of agreement		[0, 1]	1	Larger md
17	rd	Relative index of agreement		$(-\infty, 1]$	1	Larger rd
18	KGE	Kling–Gupta efficiency	Type 2 accuracy, capture of the variance, correlation	$(-\infty, 1]$	1	Larger KGE

Their definitions are given in the supplementary material

are also considered helpful in communicating the results of the present study. The use of their original versions, which are known to be over-sensitive and under-sensitive to high and low outliers respectively (Krause et al. 2005), is herein complemented by the use of their modified and relative versions, i.e., the mNSE, rNSE, md and rd metrics. These four metrics can provide improved forecast evaluation depending on the data (Krause et al. 2005). Moreover, Zambrano-Bigiarini (2014) places cp, VE, PBIAS, rSD and KGE amongst the metrics of potential interest to hydrological scientists and provides references about their use in the hydrological field (see, e.g., Kitanidis and Bras 1980; Yapo et al. 1996). VE and KGE are introduced by Criss and Winston (2008) and Gupta et al. (2009) to overcome some drawbacks of NSE (and mNSE).

The rationale of using this large set of forecast quality metrics is also supported by suggestions made by experts in the field of hydrology and beyond; see Abrahart et al. (2008) and Armstrong (2001) respectively. According to the latter study, when feasible, multiple metrics should be used collectively with an emphasis on the most relevant ones. Herein, we place some emphasis on type 1 accuracy, since a good performance with respect to this criterion is a major pursuance in most of the forecasting applications. Finally, we note that amongst the utilized forecast quality metrics the MAPE, NSE, mNSE, rNSE, cp, MPE, PBIAS, VE, rSD, Pr, r2, *d*, md, rd and KGE ones are dimensionless, while MAE, RMSE and ME are expressed in the same units as the data (and the forecasts).

2.5 Methodology outline

To compare the forecasting methods of Sect. 2.3 we conduct 12 large-scale computational experiments based on simulations. Within each of these experiments we simulate 2000 time series according to a stochastic process (see Sect. 2.1). We conduct each simulation experiment twice; the first time using time series of 100 values and the second time using time series of 300 values. The simulation experiments are hereafter referred to under their code names. The latter are composed by two parts separated by an underscore. The first part is "SE" (acronym for Simulation Experiment), while the second part is the serial number of the simulated process, as reported in Table 2, followed by the letter "a" or "b" to denote the length of the simulated time series, i.e., 100 or 300 values respectively. Additionally, we conduct a real-world experiment using the time series presented in Sect. 2.2. Within the experiments using ARMA simulated processes we test all the forecasting methods except for auto_ARFIMA. The latter method is tested within the experiments using ARFIMA simulated processes or real-world time series instead of the ARIMA_f, ARIMA_s, auto_ARIMA_f and auto_ARIMA_s ones. The total number of forecasts is 858,480, among which 6480 are produced within the real-world experiment.

For the application of the stochastic methods we divide each time series into two segments, i.e., the training segment and the test segment, which contain n_1 and n_2 values respectively, as indicated in Fig. 2a. We fit the stochastic models to the former and produce forecasts for the latter using the recursive multi-step ahead forecasting method. For the total of the conducted experiments n_2 equals 10, while n_1 equals 90 or 290 depending on the length of the used time series. For the application of the ML forecasting methods, we additionally divide the segment of n_1 values into two parts, as presented in Fig. 2b. The tail of the training segment is hereafter referred to as "validation segment" and serves hyperparameter selection, as delineated subsequently. We fit the ML model several times to the first $[2n_1/3]$ values of the training segment, each time using different hyperparameter values according to Table 4. The fitted configurations of the ML model are then utilized to produce forecasts for the validation segment. We compute the RMSE values of these forecasts using the actual values of the validation segment as reference information and decide on an optimum hyperparameter value (i.e., the corresponding to the smallest RMSE). Finally, we fit the ML model with the selected hyperparameter value to the whole training segment and produce forecasts for the test segment. The rationale of adopting this procedure is explained in Witten et al. (2017, pp. 171–172; see also Sect. 2.3.2). In summary, both the validation and test segments are used for testing and comparing models that have been previously fitted to independent (with respect to these segments) information. The former testing facilitates the decision on a ML method variation, so that the ML method is afterwards considered fully trained, while the latter enables the comparison between all the (fully trained) forecasting methods.

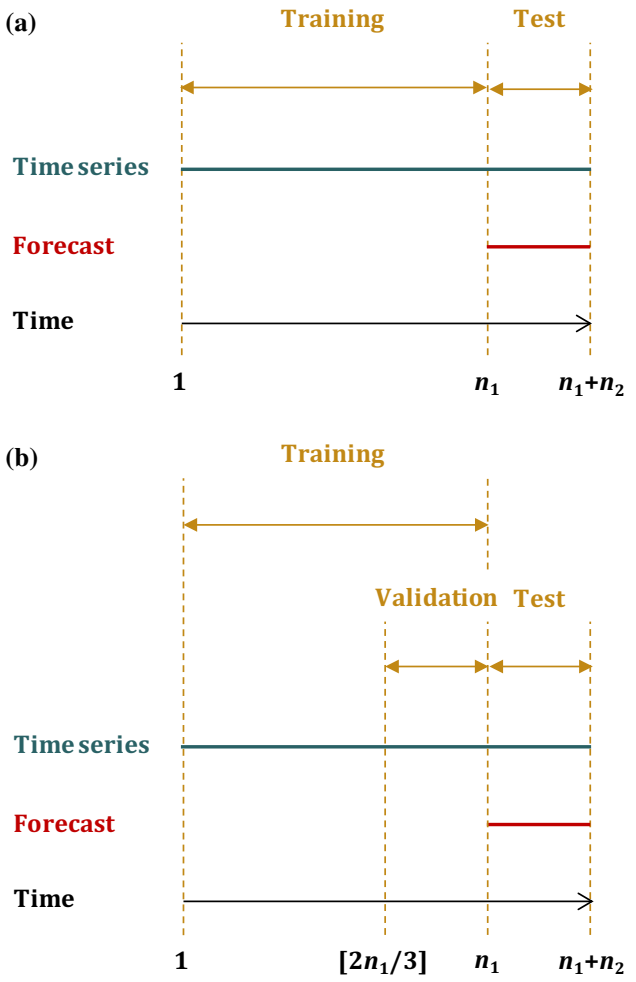

Fig. 2 Time series segment division for the application of the **a** stochastic and **b** machine learning methods. For the latter category the validation segment serves the hyperparameter optimization procedure

We provide a multi-faced assessment and comparison of the forecasting methods by utilizing the forecast quality metrics briefly presented in Sect. 2.4. The values of these metrics are computed for each forecasting test (conducted for a specific forecasting method and a specific time series) on the test segment. We mainly compare the medians and interquartile ranges (iqr) of the metric values, as computed for each forecasting method per experiment. We compare the medians, as described in Table 6, while the smallest the iqr the better the forecasts. We also apply a clustering

analysis on the forecasting methods based on the median values of the forecast quality metrics. This analysis can ease the extraction of information from the experiments. It can also facilitate the identification of possible repeating patterns in the clustering of the forecasting methods. The presence or absence of such repeating patterns could be strongly connected to algorithmic aspects and elements that we aim to reveal with the conducted experiments. In particular for the real-world experiment, we rank the forecasting methods for each individual test and further compute an average-case ranking for each metric. We place our emphasis on the 18 average-case rankings and not directly in the mean or median values of the metrics, because the latter might be more affected by the results of specific time series. This practice was first adopted by Tyralis and Papacharalampous (2017).

Finally, we measure the total computational time consumed by each forecasting method within the various experiments using the system.time built-in-R function. We present these measurements to allow a simplified and easily interpretable comparison of the implemented methods in terms of computational requirements. The computations are performed in our regular home PC, while the computational times could differ significantly for other PCs.

2.6 Benchmarking information

Although our computational experiments are designed to produce new knowledge in the field of hydrological time series forecasting, there are several outcomes rather well known at the forefront of our methodological framework. In more detail, ARIMA_f is expected to produce optimal forecasts with respect to the type 1 accuracy criterion, mainly in terms of RMSE, on the time series resulting from the simulation of ARMA processes because of its theoretical background, specifically for two reasons. Firstly, it uses by design the p, d, q numbers that are used in the simulation procedure; therefore, in its case the forecasting procedure is in essence the inverse of the simulation procedure. Furthermore, it produces minimum mean square error forecasts by setting the innovations to zero (see Wei 2006, pp. 88–93 for the related theoretical proof). Moreover, auto_ARIMA_f should be slightly worse, since it exploits information about the type of the simulated processes, although to a lesser extent, since the values of p, d, q are not known a priori (but they are estimated during the training process). Similarly to the ARIMA_f and the auto_ARIMA_f methods, auto_ARFIMA is expected to exhibit the best performance in terms of RMSE when applied to the time series resulting from the simulation of ARFIMA processes. Finally, ARIMA_s and auto_-ARIMA_s are expected to be best performing in capturing the variance exhibited by the simulated time series, while together with ETS_s are expected to not be amongst the most accurate. The six forecasting methods mentioned in the above lines play the role of benchmarks within our methodological approach, since they serve as a reference for the assessment of the remaining methods within the simulation experiments. Other benchmarks used herein are the simple methods. These two methods are amongst the most commonly used benchmarks in the forecasting field (Hyndman and Athanasopoulos 2018, chapter 3.1). The above-outlined information is used in interpreting and discussing our results.

3 Results

3.1 Simulation experiments

Section 3.1 aims at providing a synopsis of the results of the simulation experiments. To support our key findings, here we present a small representative sample of the entire information. For the about 13,000 figures, conducted in the context of an exploratory visualization, as well as for the numerical summaries of the results in table form, the reader is referred to the fully reproducible reports, which are available together with their codes in the supplementary material. In the latter we also enclose the report entitled "Selected figures for the qualitative comparison of the forecasting methods", which includes Figures S.1–S.24. These figures can support the main conclusions of this paper in a satisfactory manner.

In Figs. 3, 4, 5, 6, 7, 8 and 9 we present the side-by-side boxplots of the values of the forecast quality metrics computed within the SE_1a simulation experiment. These figures can provide a rough outline of the forecasting methods and the utility of the forecast quality metrics within this study. By their examination, we observe that the ARIMA_f and auto_ARIMA_f benchmarks are the best performing with respect to type 1 accuracy, as assumed in Sect. 2.6, while BATS exhibits a very close to these methods performance, perhaps because it uses information from an ARMA model. We also note that the total of the ML methods except for NN_1 are competitive with BATS and with each other, while they are also better than the stochastic SES and Theta. The latter forecasting methods share a quite similar performance, a fact also applying to Naïve and RW. These simple benchmarks are better than NN_1 and the simulation models (ARIMA_s, auto_-ARIMA_s, ETS_s), amongst which ETS_s produces forecasts with the most varying metric values and the worst median. Regarding the type 2 accuracy, all the methods seem to have rather equally good average-case performance, since the differences in the latter are small and not perceivable from these figures. However, the metric values

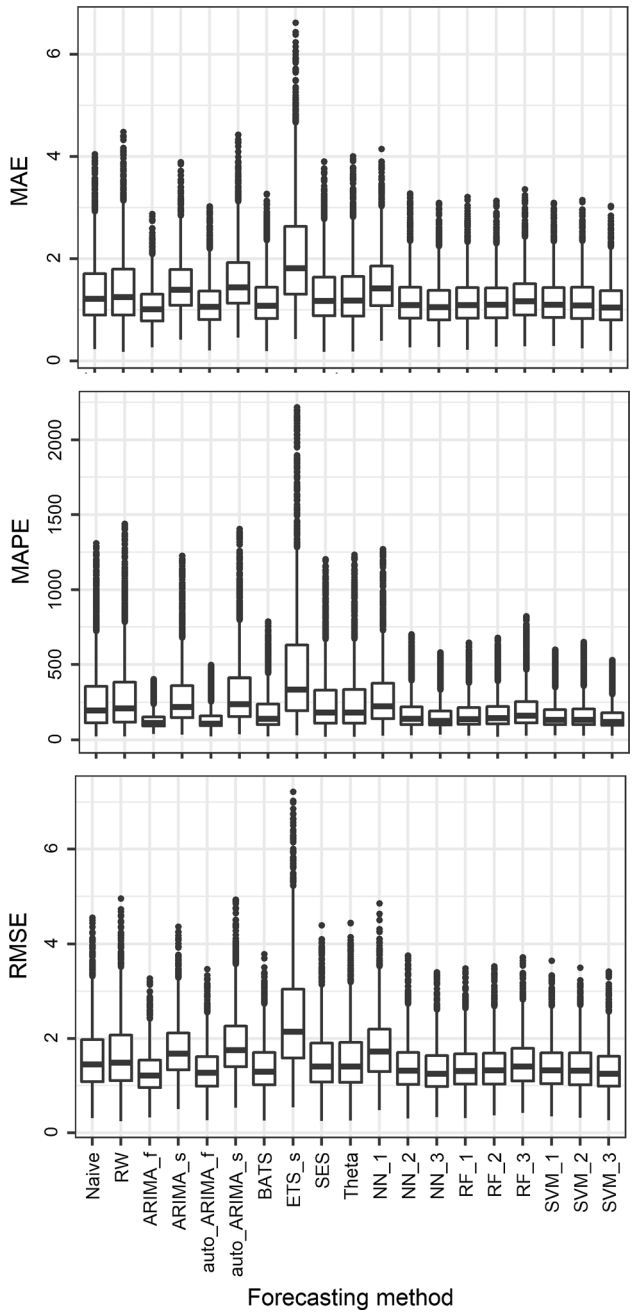

Fig. 3 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the type 1 accuracy criterion within the SE_1a simulation experiment (part 1). The far outliers have been removed

computed for ETS_s are the most scattered with respect to each other, while the opposite applies to the metric values computed for ARIMA_f, auto_ARIMA_f, BATS and all the ML methods apart from NN_1. The metric values computed for the remaining forecasting methods are scattered with respect to each other to an extent in between.

In terms of rSD, the image is mostly reversed compared to the one produced by the type 1 accuracy metrics. Naïve, RW, SES and Theta are clearly the worst, while the ML methods are more segregated. The average-case performance of NN_1, ARIMA_s, auto_ARIMA_s and ETS_s is good. Nevertheless, the rSD values for these four forecasting methods can vary significantly from the one forecasting attempt to the other, more than the rSD values computed for the remaining forecasting methods, a fact also applying to the rest of the forecast quality metrics. Regarding the average-case performance with respect to correlation, ARIMA_f, auto_ARIMA_f and BATS are the best, followed by NN_3. With respect to both type 1 accuracy and capture of the variance, ARIMA_f, auto_-ARIMA_f, BATS and all the ML methods except for NN_1 are clearly better than the simple benchmarks and competitive with each other. SES and Theta, on the other hand, exhibit a very close performance to the one of Naïve and RW. Finally, in terms of KGE, the best performing methods are the same three stochastic and eight ML ones. NN_1, ARIMA_s and auto_ARIMA_s are better than Theta, which is competitive with RW. Overall, we observe that for the SE_1a simulation experiment the forecast quality metrics (even the corresponding to the same criterion) provide different aspects of the same information to an extent larger or smaller (as it is expected; see Sect. 2.4), while these 18 different aspects may also be conflicting to each other.

Subsequently, we state the main observations obtained from the total of the simulation experiments. To base these observations, in Fig. 10 we present the heatmaps of the average-case performance of the forecasting methods within the SE_1a, SE_1b, SE_2a and SE_2b simulation experiments, while in Figs. 11, 12 and 13 we present the heatmaps formed using the medians of the total of the RMSE, rSD and d values respectively. In these figures the scaling is performed in the row direction and the darker the colour the better the forecasts. The conducted clustering analysis on the forecasting methods based on their performance is also presented. Some observations obtained from SE_1a apply to the rest of the simulation experiments as well. These are the following (see, e.g., Figs. 10, 11, 12, 13): (a) forecasting methods from both the stochastic and ML categories are amongst the best and worst performing ones, (b) the metrics can provide significantly different, even conflicting, image regarding the performance of the forecasting methods, (c) the ARIMA_f, auto_ARIMA_f and auto_ARFIMA benchmarks are the best performing in terms of type 1 accuracy, while ETS_s, ARIMA_s and auto_ARIMA_s exhibit a good average-case performance in terms of rSD, (d) the image produced by rSD is mostly reversed with respect to the one produced by the type 1 accuracy metrics, i.e., methods that are well performing according to the latter criterion are bad performing with respect to the capture of the variance of the time series,

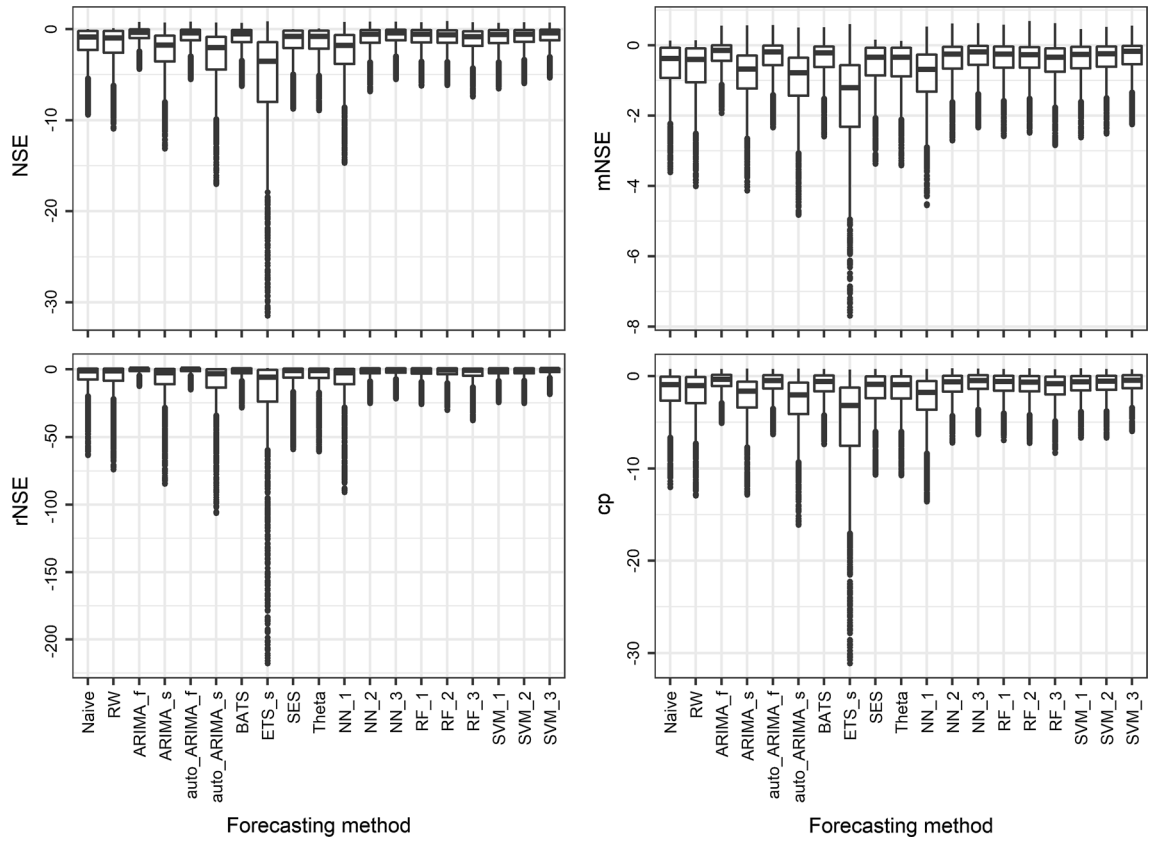

Fig. 4 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the type 1 accuracy criterion within the SE_1a simulation experiment (part 2). The far outliers have been removed

(e) BATS is very close to ARIMA_f, auto_ARIMA_f and auto_ARFIMA, and (f) Naïve and RW, as well as SES and Theta, exhibit similar performance to each other. Nevertheless, the Pr, r2 and KGE metrics are not defined for the forecasts produced by Naïve and SES. Finally, by the examination of the side-by-side boxplots produced for each and every of the simulation experiments we note that (g) ARIMA_s, auto_ARIMA_s, ETS_s and NN_1 seem to share a form of instability, i.e., their metric values vary more than the metric values of other forecasting methods. The latter concerns the results obtained from all the forecast quality metrics except for Pr and r2.

By the examination of Figs. 10, 11, 12 and 13 (or Figures S.1–S.24) we observe that the image provided by the metrics and the resulted clustering of the forecasting methods can also vary from the one simulation experiment to the other. Especially Figs. 11, 12 and 13 (or Figures S.7–S.24), allow us to easily perceive that the differences in the results of the various simulation experiments, also depicted in the clustering of the forecasting methods, are more related with the information provided by specific metrics and mostly concern specific forecasting methods. In fact, the heatmaps formed for the MAE, MAPE, RMSE, NSE, mNSE, rNSE, cp, rSD and KGE metrics are smoother than those formed for the remaining forecast quality metrics. In particular, the pictures obtained from ME, MPE, VE, r2, *d* and md are the most dispersed. On the other hand, the Naïve, RW, ARIMA_s, auto_ARIMA_s, ETS_s, SES, Theta and NN_1 forecasting methods are more likely to have a varying performance (which results in varying clustering of forecasting methods). For example, we observe that Naïve and RW exhibit rather the best average-case performance in terms of *d* (see Fig. 13) and md (see Figure S.22), while they have either bad, moderate or good average-case performance in terms of MAE, MAPE, PBIAS and VE depending on the simulation experiment (see Figures S.7, S.8, S.16 and S.17 respectively). The same applies to SES and Theta in terms of *d*, etc. We also note that forecasting methods resulting from the implementation of the same algorithm can exhibit a far distant or always close performance depending on the algorithm, as it is also perceivable by the examination of the resulted clustering of the forecasting methods. For instance, NN_1 and NN_2 (or NN_3) may differ with each other to a great extent, a fact also applying to ARIMA_s and ARIMA_f, but not to the RF and SVM forecasting methods. Interestingly, we observe that the training length largely affects the performance of NN_1 in a systematic way, while the

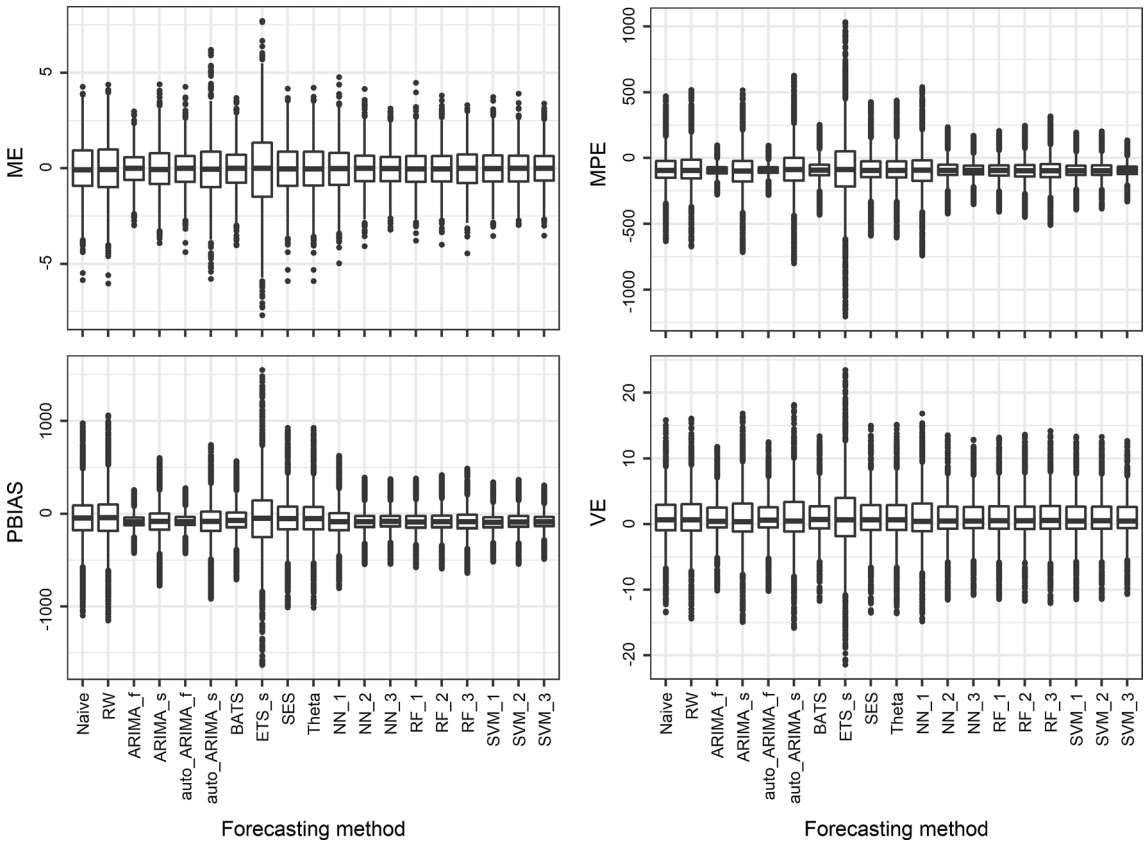

Fig. 5 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the of type 2 accuracy criterion within the SE_1a simulation experiment. The far outliers have been removed

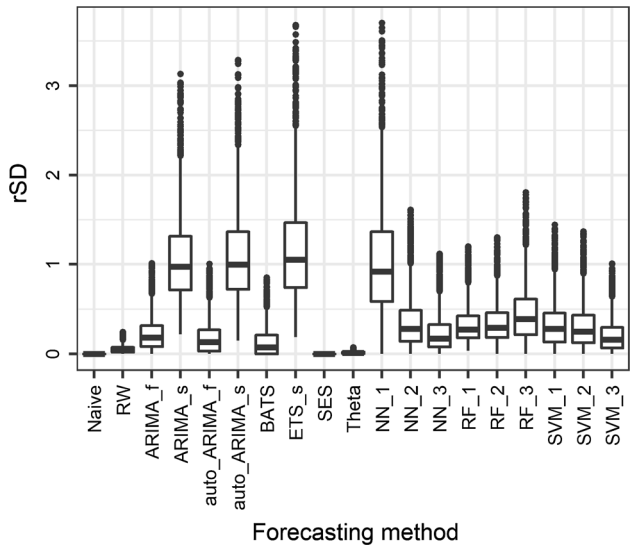

Fig. 6 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the capture of the variance criterion within the SE_1a simulation experiment. The far outliers have been removed

performance of the remaining forecasting methods is less or even slightly affected. The latter effect depends on the forecasting method, as well as on the simulated process. In detail, the NN_1 forecasting method exhibits a bad performance with respect to type 1 accuracy (and a good one in terms of rSD; see Fig. 12) within the simulation experiments using time series of 100 values, i.e., for 90-value training segments. On the contrary, its performance is good with respect to type 1 accuracy (and bad in terms of rSD) within the simulation experiments using time series of 300 values, i.e., for 290-value training segments. The latter observations concerning NN_1 might apply to a small extent to some of the remaining ML methods.

Next, we summarize some important information about the best performing forecasting methods in terms of type 1 accuracy, which has been identified as the criterion of focus herein. In terms of MAE (see Figure S.7) BATS is very close to the ARIMA_f, auto_ARIMA_f and auto_ARFIMA benchmarks, while SES, Theta and all the ML methods except for NN_1 have always a good or moderate performance. With respect to the MAPE metric (see Figure S.8) SVM_3 and BATS are mostly close to ARIMA_f, auto_ARIMA_f and auto_ARFIMA, and NN_2, NN_3, RF_1, RF_2, RF_3, SVM_1, SVM_2, SVM_3, SES and Theta are well performing for the greatest part of the simulation experiments. The same observations apply with respect to RMSE (see Fig. 11). Nevertheless, for this

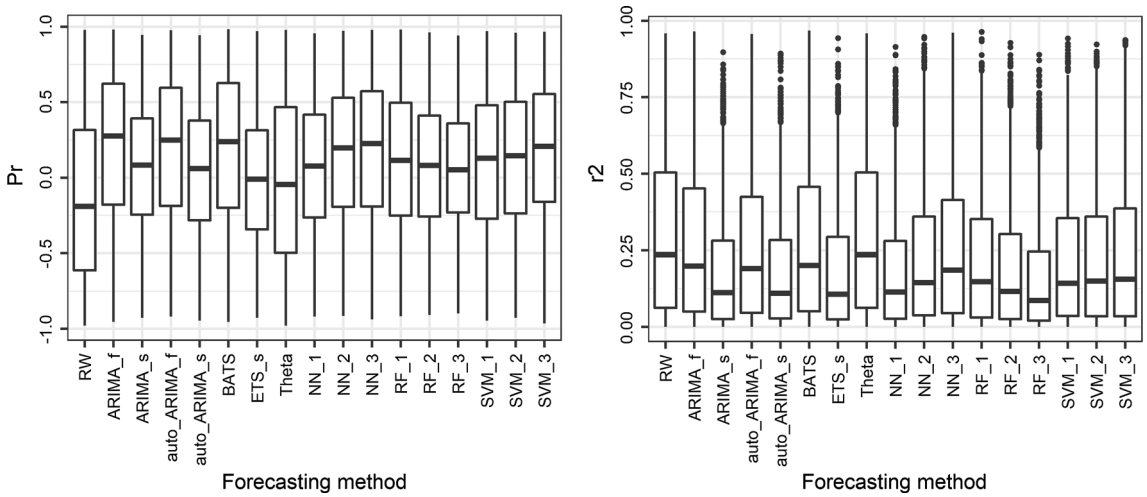

Fig. 7 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the correlation criterion within the SE_1a simulation experiment. The Pr and r2 metrics are not defined for the forecasts produced by the Naïve and SES forecasting methods and, thus, the corresponding boxplots are not presented

metric NN_2 and NN_3 are rather very close to the good benchmarks as well. Regarding the NSE, mNSE, rNSE and cp values (see Figures S.10, S.11, S.12 and S.13 respectively), most of the stochastic and ML methods are competitive to each other and to the good benchmarks. The only ones that are not competitive are the simulation models, the simple benchmarks and NN_1 (the latter for 90-value training segments).

Finally, in Tables 7 and 8 we present the total computational time consumed by the forecasting methods within the simulation experiments. In summary, the following related observations are important. Naïve, SES, Theta, ARIMA_s, ARIMA_f, ETS_s and RW consume considerably less time than the remaining methods. Moreover, NN_3 is faster than auto_ARIMA_f, auto_ARIMA_s and auto_ARFIMA for the 90-value training segments, and faster than BATS for both lengths of training segments. The computational time consumed by RF_2 and RF_3 is mostly comparable with the computational time consumed by auto_ARIMA_f, auto_ARIMA_s and auto_ARFIMA for the 90-value training segments, while it is much higher for 290-value training segments. This computational time is also lower (higher) than the computational time reported for BATS for the former (latter) category of experiments. The three SVM methods are mostly faster than BATS, which in turn consumes less time than RF_1 for 290-value training segments. NN_1 and NN_2 are found to be the most computationally intensive. Overall, the ML methods collectively consume disproportionately more computational time than the stochastic ones.

3.2 Real-world experiment

In full correspondence to the results of the simulation experiments, the results of the real-word experiment are presented in both quantitative and qualitative forms. In Figs. 14, 15, 16 and 17 we present the side-by-side boxplots of the MAPE, NSE, cp, MPE, *d* and KGE values. Additionally, in Table 9 we present the median values of the dimensionless metrics, while in Fig. 18 the average-case rankings of the forecasting methods. Here as well, we observe small differences between most of the methods, especially with respect to specific forecast quality metrics (e.g., MAPE, cp, MPE, *d*). For example, the median values of MAPE computed for auto_ARFIMA, BATS, SES, Theta, NN_3, RF_1, SVM_1, SVM_2 and SVM_3 are very close to each other. The same applies to the median values of NSE computed for the same methods, although the differences in the respective side-by-side boxplots seem to be larger in the latter case than in the former. Because of the small differences in the performance of the forecasting methods, the median metric values of Table 9 (e.g., the median MAPE values) may result to a different ranking of the forecasting methods than the average-case ranking presented in Fig. 18.

Furthermore, while the average-case rankings with respect to accuracy mostly favour stochastic methods (SES, Theta, auto_ARFIMA and BATS), SVM_1 is also ranked amongst the best performing methods. In more detail, SES is ranked first according to MAE, RMSE, NSE, mNSE, cp, ME, MPE, PBIAS and VE, but it is worse than SVM_1, and SVM_1 and SVM_2 according to MAPE and rNSE respectively. According to the latter metrics, the best performing method is BATS. This method has a rather

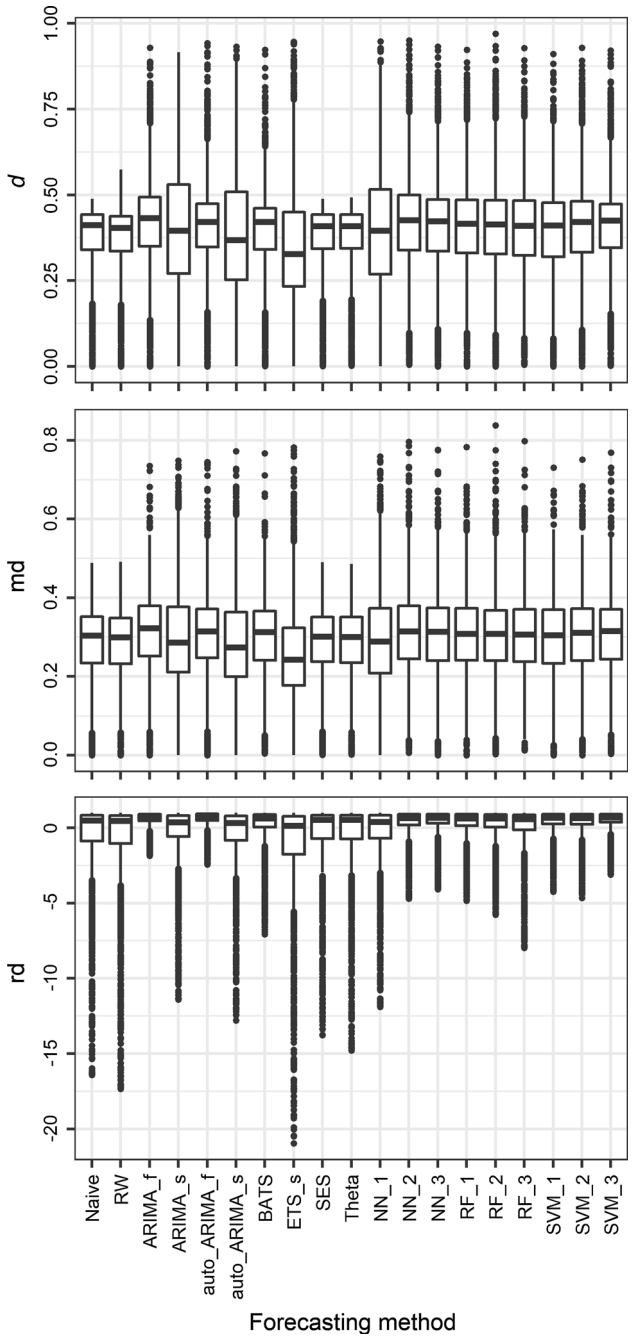

Fig. 8 Side-by-side boxplots for the comparative co-assessment of the forecasting methods regarding their performance according to the type 1 accuracy and capture of the variance criteria within the SE_1a simulation experiment. The far outliers have been removed from the side-by-side boxplots of the rd values

moderate overall performance in terms of accuracy. The less accurate methods, on the other hand, are Naïve, RW, ETS_s and NN_1, as it is expected from the simulation experiments. With respect to the remaining criteria, SES is clearly the worst performing method, while Theta, Naïve, BATS, SVM_1, NN_3 and auto_ARFIMA are also ranked behind the remaining ML methods, amongst which NN_1

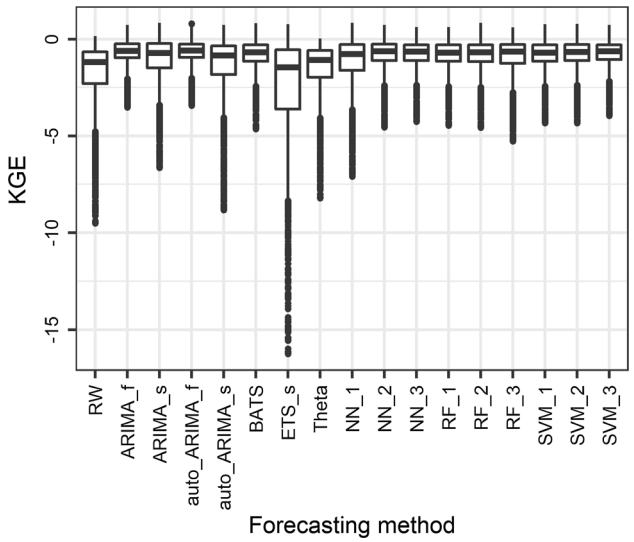

Fig. 9 Side-by-side boxplots for the comparative co-assessment of the forecasting methods regarding their performance according to the type 2 accuracy, capture of the variance and correlation criteria within the SE_1a simulation experiment. The far outliers have been removed. The KGE metric is not defined for the forecasts produced by the Naïve and SES forecasting methods and, thus, the corresponding boxplots are not presented

is mostly ranked first. Finally, in terms of computational requirements within this real-world experiment the methods could be ranked from best (1st) to worst (16th) as follows: Naïve, SES, Theta, RW, ETS_s, NN_3, auto_ARFIMA, RF_3, RF_2, SVM_3, SVM_2, SVM_1, RF_1, NN_2, BATS and NN_1 (see also Table 10).

4 Discussion

4.1 Contribution in hydrology and beyond

The present study contributes by developing a detailed framework for assessing forecasting techniques in hydrology. Furthermore, its findings can provide new insights into the nature of short hydrological time series forecasting at large time scales, while they concern all natural processes that could be modelled by linear stationary processes. A first view of the results suggests that the differences in the forecasting performance of the methods are mostly small (insignificant for hydrometeorological applications; see also the experiments of Papacharalampous et al. 2018b), while the stochastic and ML methods can share a quite similar forecasting performance when implemented to hydrological time series of small length and small temporal resolution (e.g., annual or monthly). In fact, methods from both these categories are found to perform better or worse mainly depending on the forecast quality metric, but on the experiment as well. Regarding the type 1 accuracy, in the

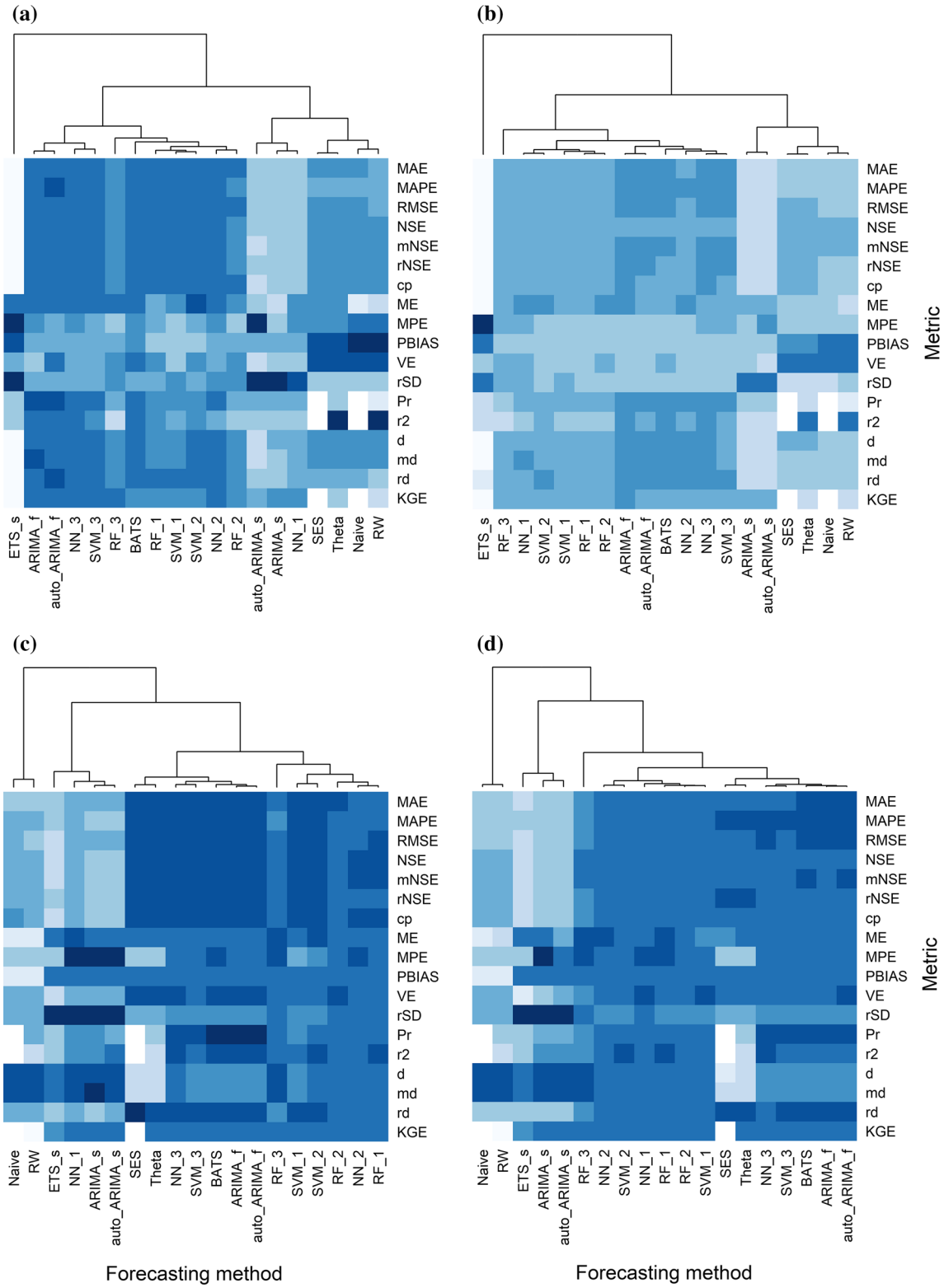

Fig. 10 Heatmaps for the comparative assessment of the forecasting methods within the **a** SE_1a, **b** SE_1b, **c** SE_2a, **d** SE_2b simulation experiments according to the median values of the forecast quality metrics and the conditions listed on Table 6. The Pr, r2 and KGE metrics are not defined for the forecasts produced by the Naïve and SES forecasting methods. Their missing values are not taken into consideration during the comparative assessment and are imprinted with white colour

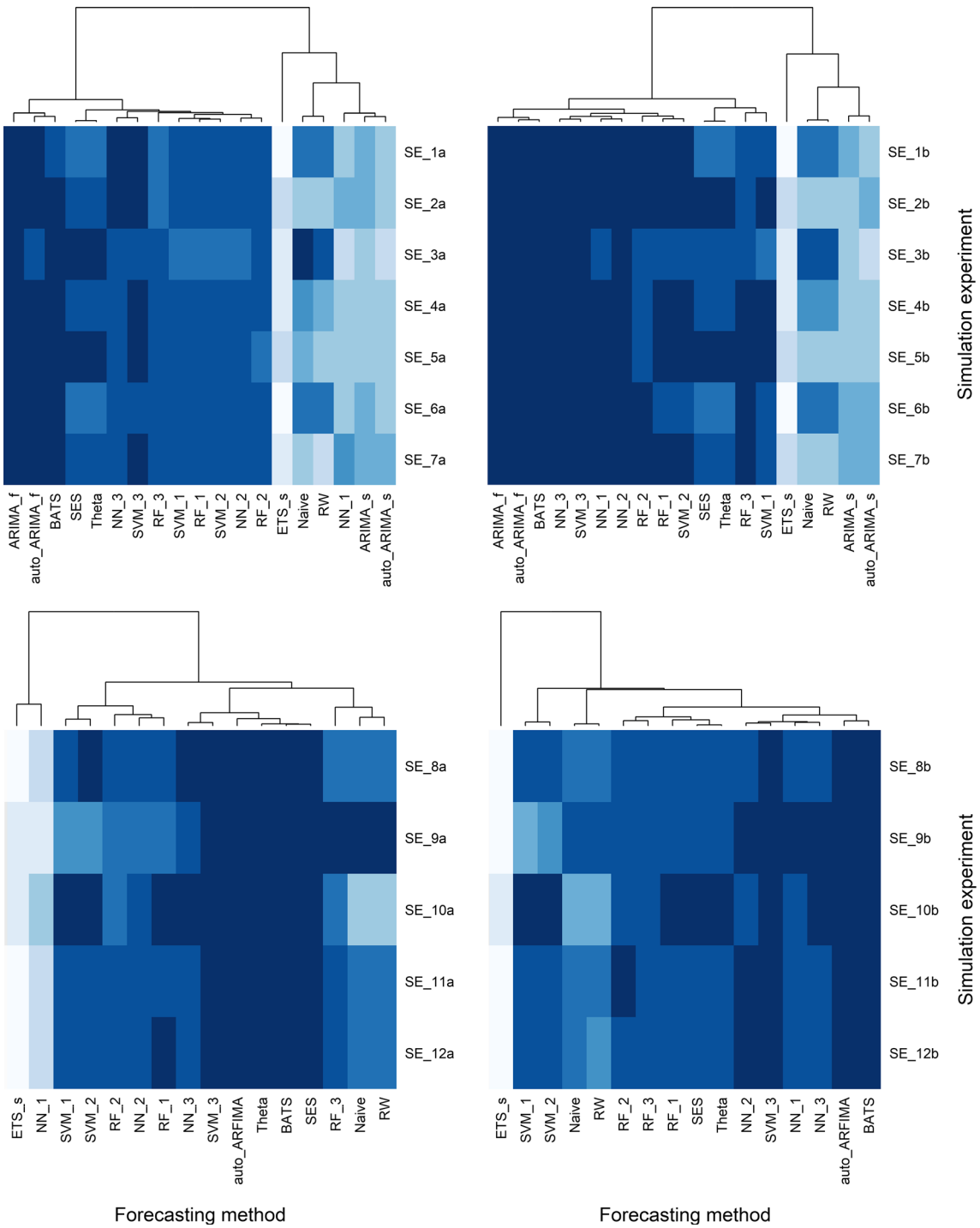

Fig. 11 Heatmaps for the comparative assessment of the forecasting methods according to the median values of the RMSE metric and the condition stated on Table 6

simulation experiments BATS is always close to the ARIMA_f, auto_ARIMA_f and auto_ARFIMA benchmarks, probably because it uses information from an ARMA model, while most of the ML methods (e.g., NN_3 and SVM_3) are amongst the best performing and often better than SES and Theta. Nevertheless, in the real-world experiment SES is mostly ranked first, followed by auto_ARFIMA, BATS, SVM_1 and Theta, while NN_3, RF_1, SVM_2, and SVM_3 are also close to the latter methods. A possible interpretation of this outcome is that for a different sample of river discharge time series, the average-case rankings would differ as well, and that there

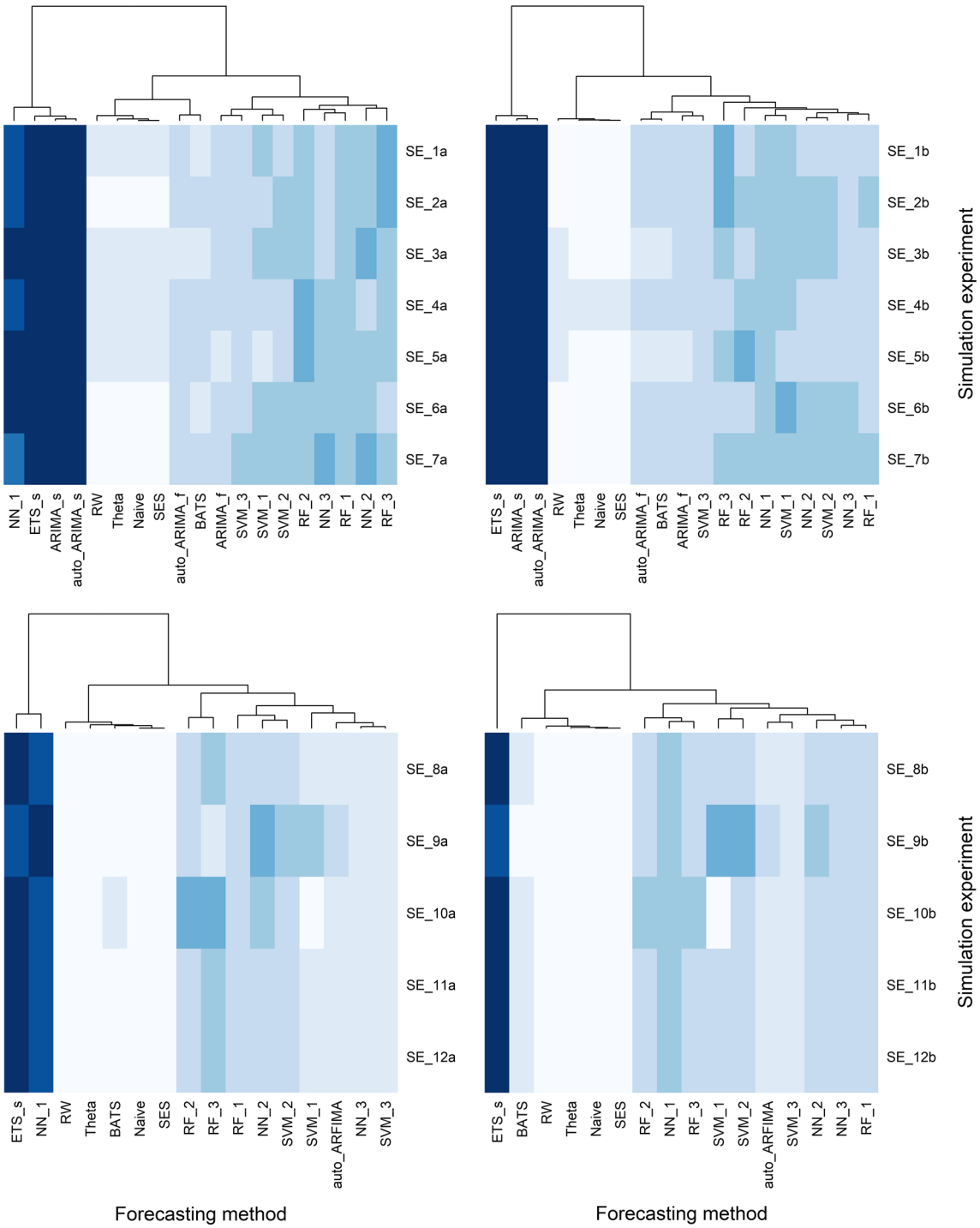

Fig. 12 Heatmaps for the comparative assessment of the forecasting methods according to the median values of the rSD metric and the condition stated on Table 6

might be no particular reason to choose some methods over others for this specific process. Given the claims that in linear situations (e.g., the simulation experiments of this study) the ML methods are more likely to be inferior to the stochastic ones, while in non-linear situations, as it is usually asserted to apply to river discharge processes, the ML methods are more likely to outperform, the algorithmically obtained results of the present study are even more interesting. Noteworthy is also the fact that our results differ from the results of Makridakis et al. (2018), which

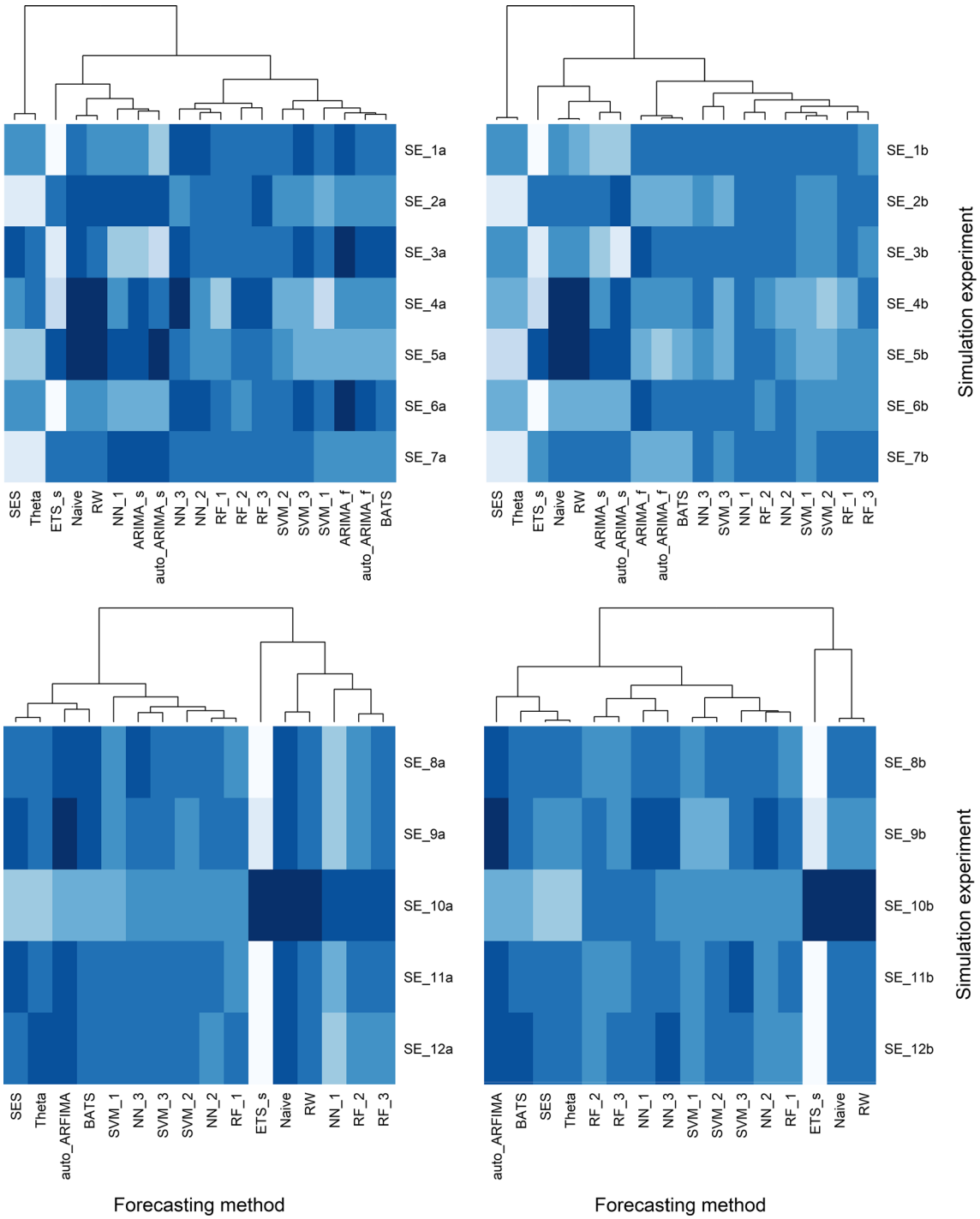

Fig. 13 Heatmaps for the comparative assessment of the forecasting methods according to the median values of the *d* metric and the condition stated on Table 6

favour the stochastic methods, probably due to the different experimental setting adopted therein (determined by the required degree of forecast accuracy, the lengths of the examined time series, the selected algorithms for performing multi-step ahead forecasting, the forecast quality metrics used for evaluating the methods and the optimization procedures of the ML methods, among others).

In this view, we would like to emphasize that the ML algorithms are accurate enough. Yet they have the worthmentioning particularity that their forecasting performance might be largely affected by the number of the utilized

Table 7 Total computational time (s) consumed by the forecasting methods within the simulation experiments (part 1)

	Naïve	RW	ARIMA_f	ARIMA_s	auto_ARIMA_f	auto_ARIMA_s	BATS	ETS_s	SES	Theta
SE_1a	0	18	11	7	127	124	331	15	3	4
SE_2a	0	19	13	10	173	171	1003	24	5	6
SE_3a	0	22	23	17	196	192	410	23	5	6
SE_4a	0	21	15	12	168	163	926	22	4	5
SE_5a	0	24	17	12	186	180	885	24	5	6
SE_6a	0	23	19	15	255	251	562	23	5	6
SE_7a	0	21	21	17	223	217	1381	21	5	6
SE_1b	0	18	15	12	148	146	1083	51	7	8
SE_2b	0	22	16	13	109	105	1222	56	8	9
SE_3b	0	25	37	30	161	155	579	51	8	8
SE_4b	0	24	21	16	129	124	1218	49	7	8
SE_5b	0	26	20	17	114	109	1159	53	8	9
SE_6b	0	26	30	24	184	180	1517	51	7	9
SE_7b	0	25	28	22	126	123	1782	49	7	8

	NN_1	NN_2	NN_3	RF_1	RF_2	RF_3	SVM_1	SVM_2	SVM_3
SE_1a	1301	827	90	343	178	141	312	215	187
SE_2a	1679	1099	129	447	242	184	449	328	278
SE_3a	1797	1312	140	440	316	184	448	448	287
SE_4a	1597	946	189	466	186	223	445	266	309
SE_5a	1693	965	198	467	178	222	452	268	302
SE_6a	1748	1073	195	405	225	194	393	259	265
SE_7a	1614	1127	213	433	249	209	397	323	297
SE_1b	6364	3645	391	3061	1421	808	890	643	539
SE_2b	6353	3726	421	3038	1466	802	892	650	531
SE_3b	6401	5349	543	2995	2414	808	894	801	529
SE_4b	6282	2986	823	3148	766	1020	786	482	542
SE_5b	6032	2829	817	3069	811	1098	895	547	620
SE_6b	6352	4012	940	2952	1561	1124	882	674	625
SE_7b	6555	4212	954	3062	1591	1089	834	630	583

The numbers have been rounded up to the nearest integer. The computations have been performed in a regular home PC

lagged variables. This number is directly related to the length of the segment used for model fitting (Tyralis and Papacharalampous 2017). Specifically, a significant decrease of this length may deteriorate the forecasting performance of a ML algorithm, as largely perceivable through the examination of the results obtained for the NN methods of the present study. In detail, for the simulation experiments using 90-value training segments, NN_1 exhibits a bad performance in terms of type 1 accuracy, a fact not applying to NN_2 and NN_3 that use less and very few lagged variables respectively. On the contrary, for the simulation experiments using 290-value training segments, NN_1 is amongst the most accurate methods. The same number of lagged variables is used by RF_1 and SVM_1. Nevertheless, the performance of the herein implemented RF and SVM algorithms seems to be less affected by the number of lagged variables than the NN algorithm. These large-scale results on time lag (or lagged variable) selection could be encountered as contributed information to the subject.

Another particularity of the ML methods is related to their computational requirements, which seem to considerably increase with increasing length of the training segment. In fact, for our regular home PC the computational time consumed by the NN and SVM methods is found to be approximately four to eight times higher for 290-value training segments than for 90-value training segments. The respective difference in computational time is smaller for the SVM methods. The number of lagged variables seems to also affect the computational requirements. Specifically,

Table 8 Total computational time (s) consumed by the forecasting methods within the simulation experiments (part 2)

	Naïve	RW	auto_ARFIMA	BATS	ETS_s	SES	Theta
SE_8a	0	23	207	457	21	4	5
SE_9a	0	23	277	458	25	5	5
SE_10a	0	25	217	689	27	5	6
SE_11a	0	18	178	402	19	4	5
SE_12a	0	20	184	406	18	4	5
SE_8b	0	24	199	743	44	6	7
SE_9b	0	26	242	902	56	8	9
SE_10b	0	23	196	860	61	9	10
SE_11b	0	20	168	641	38	5	6
SE_12b	0	23	175	653	38	5	6

	NN_1	NN_2	NN_3	RF_1	RF_2	RF_3	SVM_1	SVM_2	SVM_3
SE_8a	1614	1050	127	436	234	183	417	295	262
SE_9a	1908	1445	172	457	312	201	461	369	284
SE_10a	1681	964	127	479	176	199	432	255	265
SE_11a	1488	966	119	404	216	170	381	271	240
SE_12a	1496	970	117	406	218	169	383	272	227
SE_8b	6426	5111	654	2882	1999	872	752	667	524
SE_9b	6558	5395	525	2480	2083	665	716	625	417
SE_10b	6189	2600	564	2796	696	897	722	462	464
SE_11b	5602	4142	533	2480	1839	773	683	593	453
SE_12b	5614	4107	543	2483	1820	780	683	590	449

The numbers have been rounded up to the nearest integer. The computations have been performed in a regular home PC

the computational time increases when moving from the third to the first time lag selection procedures of Table 5, i.e., from less to more lagged variables, indicating increasing computational requirements (although the length of the lagged time series decreases), with this increase to be higher for the NN methods. Overall, the computational time collectively consumed by the herein implemented ML methods is considerably higher than the respective time measured for the stochastic methods. Nonetheless, it is also shown that there are computational intensive stochastic methods (mainly BATS), as well as ML methods with lower or comparable computational requirements with stochastic methods (e.g., NN_3, RF_3).

While there are forecasting methods regularly better or worse than others with respect to specific criteria, this does not apply to all the forecasting methods neither to all the criteria. For example, we observe that Theta can exhibit good, moderate or bad average-case performance in terms of specific forecast quality metrics depending on the simulation experiment. Furthermore, sophisticated forecasting methods (such us the above mentioned ones) do not necessarily (but mostly) provide better forecasts than the simple Naïve and RW, as also shown in previous studies (e.g., Makridakis and Hibon 2000; Cheng et al. 2017). These two methods perform almost identically in the experiments of the present study, but not for longer forecast horizons (see Papacharalampous et al. 2017b, 2018b). Another pair of similarly performing forecasting methods is SES and Theta. This latter outcome is consistent with Hyndman and Billah (2003).

In general, we cannot decide on a universally best or worst forecasting method (stochastic or ML), neither we can rank the forecasting methods based on the results of the simulation experiments. Even the relative metrics, i.e., the corresponding to the same criterion (see Table 6), provide measurements which lead us to different aspects of the same information to an extent larger or smaller depending on the pair of forecast quality metrics considered. Some of these 18 different aspects are also conflicting to each other. Any ranking of the forecasting methods would require the a priori selection of an experiment and a criterion of interest, as well as the application of a simplification procedure (e.g., use of the median values of the selected metric) and, thus, would not be general. However, the clustering of the forecasting methods is possible, though only to some extent. This clustering could be based on the similar or contrasting performance of the forecasting methods with respect to the various metrics. For example, the simulation

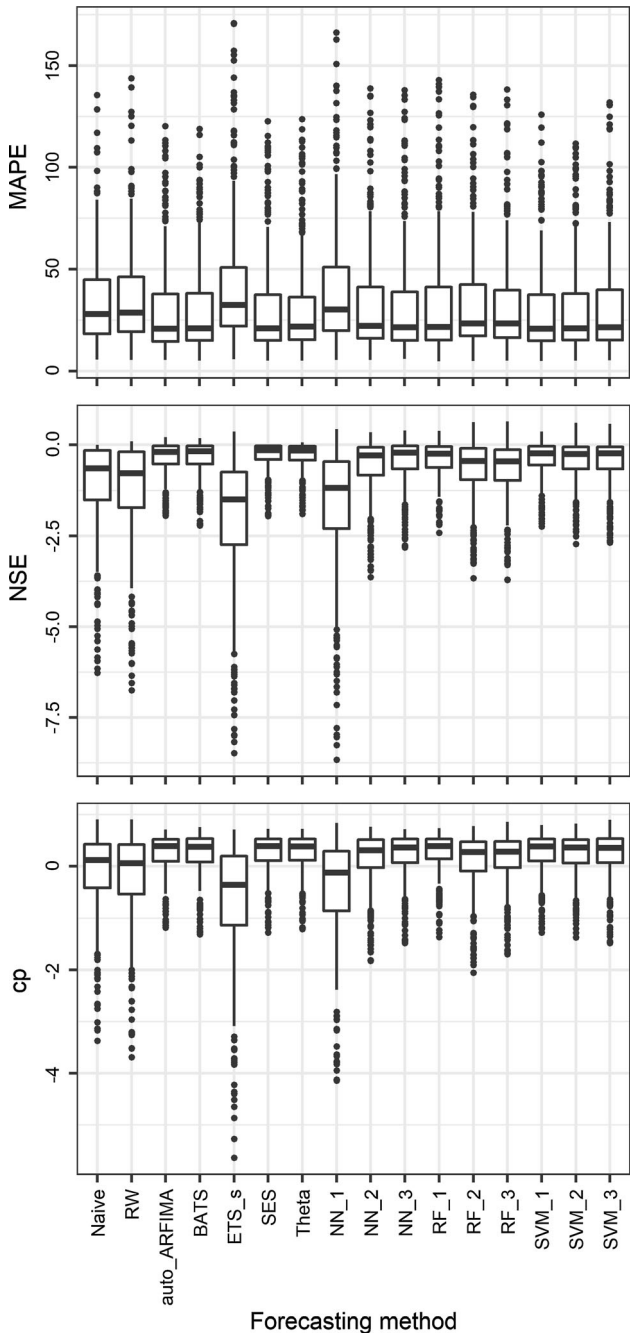

Fig. 14 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the type 1 accuracy criterion within the real-word experiment. The far outliers have been removed

models (ARIMA_s, auto_ARIMA_s and ETS_s) exhibit the best average-case performance with respect to the capture of the variance, while they are clearly the worst performing in terms of type 1 accuracy. This happens, since these two criteria are contradictory. For instance, the optimum forecast for an ARFIMA model is obtained when the innovations are set to zero.

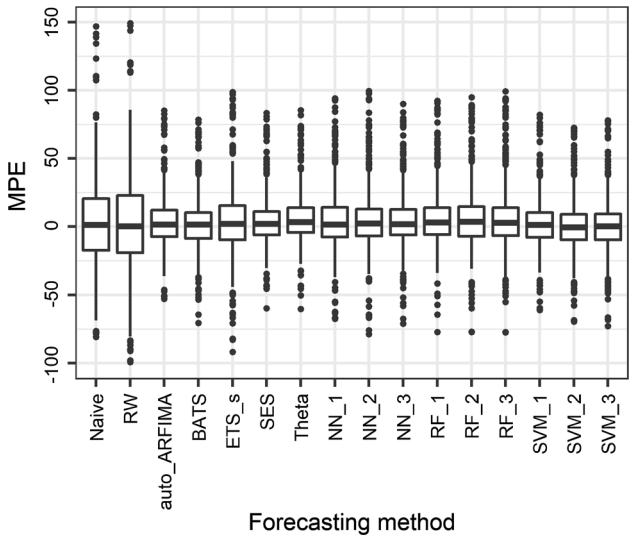

Fig. 15 Side-by-side boxplots for the comparative assessment of the forecasting methods regarding their performance according to the type 2 accuracy criterion within the real-word experiment. The far outliers have been removed

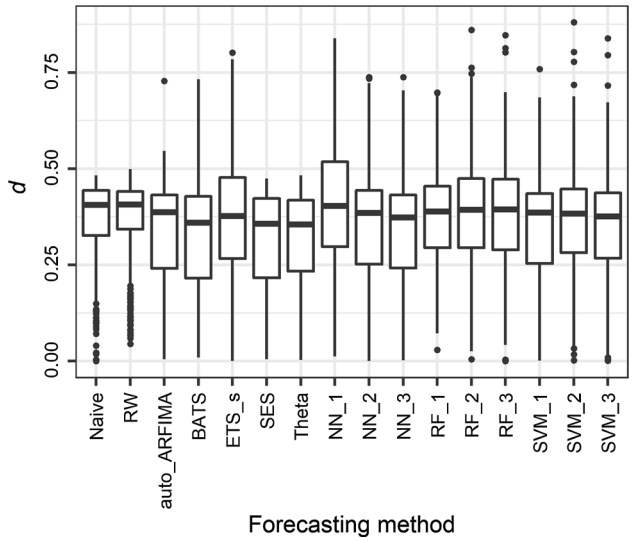

Fig. 16 Side-by-side boxplots for the comparative co-assessment of the forecasting methods regarding their performance according to the type 1 accuracy and capture of the variance criteria within the real-word experiment

Our contribution in the field of hydrology also includes the implementation of several forecasting models barely used in hydrometeorological concepts, but commonly used in the forecasting field (RW, BATS, ETS, SES and Theta) or for regression purposes (RF). This innovation holds, especially if we could exclude from the hydrological literature the large-scale companions of this study, i.e., Papacharalampous et al. (2017b, 2018a, b), and Tyralis and Papacharalampous (2017), while its practical value is indisputable. One could claim that there may be an

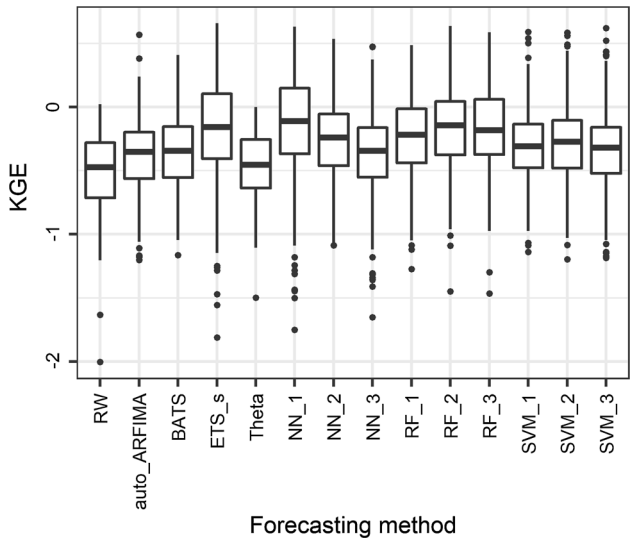

Fig. 17 Side-by-side boxplots for the comparative co-assessment of the forecasting methods regarding their performance according to the type 2 accuracy, capture of the variance and correlation criteria within the real-word experiment. The far outliers have been removed. The KGE metric is not defined for the forecasts produced by the Naïve and SES forecasting methods and, thus, the corresponding boxplots are not presented

undiscovered forecasting method (stochastic or ML), which will be better than the existing ones. As regards the "myth of the best method" the reader is referred to Hong and Fan (2016), who mention that the original techniques are countable and have been exhausted, while the hybrid techniques, i.e., combinations of original techniques, cannot further improve the forecasting performance.

Another important contribution of the present study is related to the so-called "no free lunch theorem" by Wolpert (1996). According to this theorem, in the space of all possible problem instances, there is not a model that will always perform better than the other models in the absence of significant information for the problem at hand. The present empirical study shows that even in the finite space of simple (simulated) and real-world time series examined herein there is not an optimal forecasting solution. Finding the best algorithm mostly depends on our knowledge of the system. For example, using ARFIMA models for forecasting the ARFIMA simulated time series is obviously the best choice, due to the prior known information about the system. The other methods are equivalent in performance since they cannot incorporate this knowledge. In the specific class of hydrological process forecasting finding information about the examined system could be possible, for example, with the application of principles of physics, such as the maximum entropy principle, incorporation of information from deterministic models (see, e.g., Tyralis and Koutsoyiannis 2017), understanding the processes from a chaotic perspective (see, e.g., Sivakumar 2004) and other approaches. Obviously, the knowledge of the system is not simply equivalent to the knowledge of its statistical properties (e.g., the mean, variance, ACF), but should be deeper. Therefore, the frequently met in the hydrological literature blind use of forecasting methods is not suggested.

Additionally, it seems that major advancements in the time series forecasting performance of all methods can be achieved by incorporating appropriate exogenous variables in the model, while the potential for improving their performance in univariate time series forecasting seems limited. The latter in our opinion is also due to the nature of the problem, which is simple. Therefore, methods that are more complicated will not necessarily yield better results. A relevant example is, for instance, the difference in the games of tic-tac-toe and Go. The former game is simple and can be solved by simple algorithms; therefore, the choice of a complex method is not necessary. On the other hand, the best performance on the more complex game of Go was achieved by the use of complicated machine learning algorithms (see Silver et al. 2016).

Regarding the extent to which the conclusions could be generalizable for the forecasting of short hydrological time series at large time scales, we note that the stationarity assumption and the reasoning of its appropriateness for the modelling of geophysical properties, documented in Koutsoyiannis and Montanari (2015), is consistent with the no free lunch theorem. In particular, if we cannot explain the behaviour of a geophysical process based on a deterministic mechanism, then the most appropriate models are stationary. Even in cases of deterministic systems, stochastic approaches are appropriate (Koutsoyiannis 2010). This is a frequently met case in the modelling of geophysical processes (i.e., there is not an adequate explanation for the behaviour of the geophysical process), proving that our conclusions could be generalizable.

4.2 On the methodological approach

The above section highlights the efficiency of our methodological approach in producing large-scale and representative for the field of hydrology results. Moreover, the real-world experiment particularly accounts for the case of river discharge forecasting. Someone who examines both the results of the simulation experiments and the real-world experiment has a more complete picture of the underlying phenomena than whom considering only the results of the simulation experiments. On the other hand, the use of simulated processes combined with benchmarking has proved pivotal in achieving our aim under the linearity and stationarity assumptions. Additionally, the use of an adequate number of forecasting methods and forecast quality metrics in the present study is also of crucial importance. Using fewer forecasting methods and fewer

Table 9 Median values of the dimensionless metrics computed within the real-word experiment

	Naïve	RW	auto_ARFIMA	BATS	ETS_s	SES	Theta
MAPE	29.21	29.83	22.04	22.04	33.81	22.02	22.86
NSE	− 0.72	− 0.84	− 0.20	− 0.19	− 1.57	− 0.17	− 0.18
mNSE	− 0.27	− 0.31	− 0.07	− 0.07	− 0.61	− 0.06	− 0.07
rNSE	− 0.81	− 0.90	− 0.35	− 0.39	− 2.24	− 0.35	− 0.45
cp	0.09	0.03	0.39	0.38	− 0.37	0.39	0.38
MPE	2.83	1.47	2.99	2.20	3.29	3.32	5.07
PBIAS	− 6.34	− 6.34	− 3.14	− 4.25	− 2.72	− 2.90	− 1.56
VE	0.71	0.71	0.78	0.78	0.67	0.78	0.78
rSD	0.00	0.03	0.05	0.00	1.02	0.00	0.01
Pr	–	− 0.05	0.06	0.04	0.00	–	− 0.04
r2	–	0.07	0.06	0.05	0.06	–	0.07
d	0.41	0.41	0.39	0.36	0.38	0.36	0.36
md	0.31	0.31	0.28	0.28	0.28	0.27	0.26
rd	0.29	0.30	0.25	0.26	0.30	0.22	0.18
KGE	–	− 0.47	− 0.35	− 0.34	− 0.17	–	− 0.46

	NN_1	NN_2	NN_3	RF_1	RF_2	RF_3	SVM_1	SVM_2	SVM_3
MAPE	32.30	24.05	22.95	23.06	25.19	24.81	22.03	22.24	22.29
NSE	− 1.26	− 0.33	− 0.22	− 0.25	− 0.47	− 0.46	− 0.24	− 0.26	− 0.23
mNSE	− 0.51	− 0.14	− 0.09	− 0.11	− 0.20	− 0.19	− 0.09	− 0.10	− 0.10
rNSE	− 1.83	− 0.59	− 0.45	− 0.46	− 0.86	− 0.78	− 0.36	− 0.40	− 0.42
cp	− 0.16	0.30	0.36	0.37	0.27	0.25	0.38	0.35	0.34
MPE	2.94	4.61	3.36	4.31	4.62	3.96	3.00	1.17	1.49
PBIAS	− 3.05	− 2.09	− 2.41	− 1.19	− 1.80	− 2.59	− 4.50	− 5.84	− 4.60
VE	0.69	0.76	0.78	0.78	0.75	0.76	0.78	0.78	0.78
rSD	0.94	0.21	0.05	0.24	0.42	0.40	0.00	0.12	0.07
Pr	0.08	0.08	0.02	0.08	0.08	0.04	0. 08	0.07	0.05
r2	0.05	0.06	0.06	0.06	0.07	0.06	0.06	0.06	0.06
d	0.40	0.39	0.37	0.39	0.39	0.39	0.39	0.38	0.38
md	0.30	0.29	0.28	0.28	0.29	0.30	0.29	0.30	0.29
rd	0.33	0.28	0.22	0.30	0.30	0.31	0.29	0.34	0.30
KGE	− 0.12	− 0.24	− 0.35	− 0.22	− 0.15	− 0.19	− 0.31	− 0.27	− 0.32

forecast quality metrics would have led to a very different overall picture, particularly if those fewer metrics corresponded to fewer criteria. Besides, the comparison is rather the only available research method for any evaluation and, consequently, the larger its scale the more generalized the derived results. For this specific reason, the novel (mainly for hydrology) methodological approach of the present study is considered appropriate for the assessment of forecasting methods in hydrology. Furthermore, the qualitative form of the results facilitates their handy examination and, thus, eases the delivery of the large-scale findings. In fact, our methodology enables the assessment of the failure risk or, alternatively worded, the available opportunities for success that accompany the use of a specific forecasting method to a significant extent, while it also leads to the recognition of several advantages/disadvantages characterizing the latter. This knowledge is fundamental to the forecasters and the users of the forecasts, since a specific forecasting method can be both useful and useless, depending on the forecasting task.

5 Conclusions

We conduct an extensive comparison between several stochastic and machine learning methods for the multi-step ahead forecasting of hydrological processes by performing large-scale computational experiments based on simulations under the linearity and stationarity assumptions. The implemented stochastic methods include simple models,

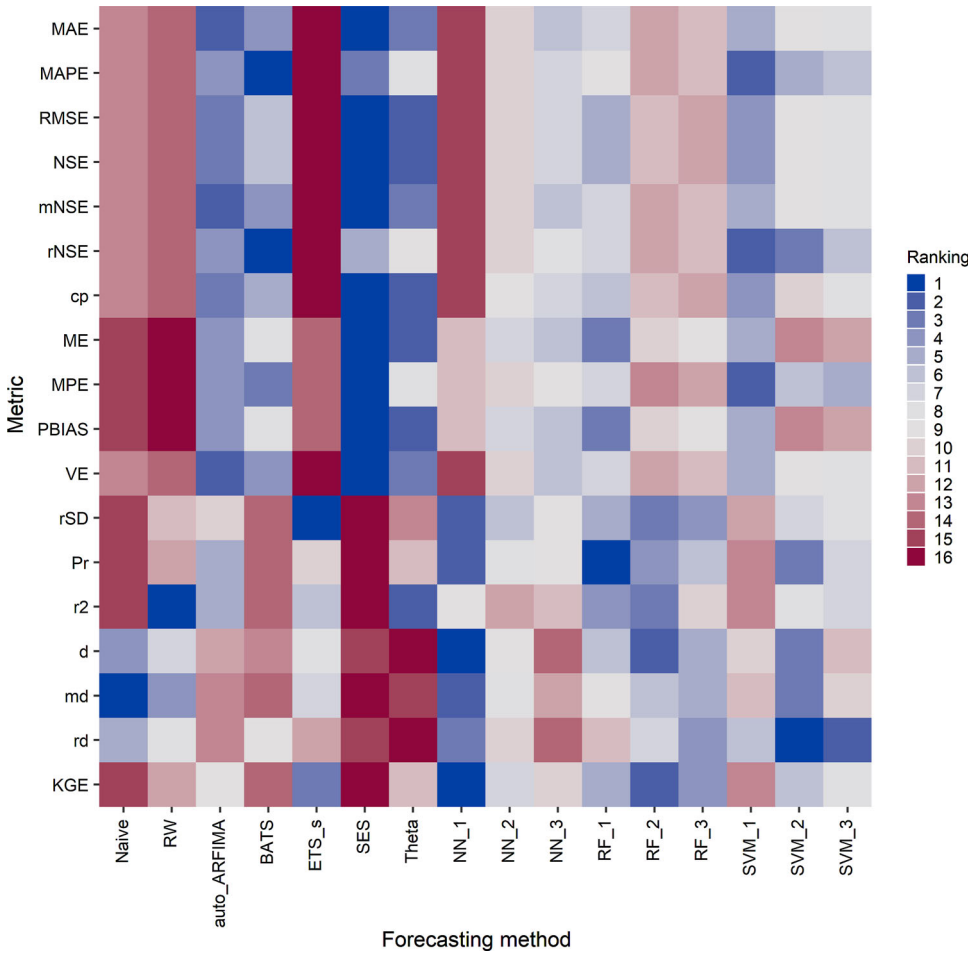

Fig. 18 Heatmap for the comparative assessment of the forecasting methods within the real-world experiment according to their average-case rankings. The latter are based on the values of the forecast quality metrics and the conditions listed on Table 6. The Naïve and SES forecasting methods are ranked 15th and 16th according to rSD, Pr, r2 and KGE. Their rSD values are 0, while the Pr, r2 and KGE metrics are not defined for their forecasts

Table 10 Total computational time (s) consumed by the forecasting methods within the real-world experiment

Naïve	RW	auto_ARFIMA	BATS	ETS_s	SES	Theta	NN_1	NN_2	NN_3	RF_1	RF_2	RF_3	SVM_1	SVM_2	SVM_3
0	3	26	200	6	1	1	283	174	20	70	32	31	62	42	40

The numbers have been rounded up to the nearest integer. The computations have been performed in a regular home PC

models from the frequently used families of Autoregressive Moving Average and Autoregressive Fractionally Integrated Moving Average, as well as Innovations State Space and Exponential Smoothing models, while the machine learning ones are Neural Networks, Random Forests and Support Vector Machines. The aim is to provide large-scale results, while the respective comparisons in the literature are usually based on case studies. We also run a real-world experiment on the largest river discharge dataset ever used for forecasting purposes within a framework that is purely statistical. Despite this specific focus, the results concern all natural processes in large time scales (e.g., annual or monthly) that could be modelled by stationary processes. The findings suggest that stochastic and machine learning methods do not differ dramatically. In fact, methods from both these categories are found to be equally useful in univariate short time series forecasting at large time scales. This is particularly important, because it reveals that the forecast quality is subjected to limitations. The latter are imposed by the nature of the examined problem and manifest themselves in the computed forecast quality metric values. We have empirically proved that these values do not favour any specific forecasting method, stochastic or machine learning, in a long run. In fact, the results are consistent with the no free lunch theorem, albeit the theorem refers to an infinite space of problem instances, while here we examine a finite space of problems. The empirical investigation shows that in the given finite space, formed by simulated and annual river discharge time series, the no free lunch theorem is still satisfied.

Acknowledgements We thank the Associate Editor and two reviewers for their useful suggestions. Part of the Discussion section, in particular the comments on the no free lunch theorem and the use of exogenous variables, has been inspired by the "Energy Forecasting" blog (http://blog.drhongtao.com/).

Author contributions HT conceived the idea of comparing stochastic and machine learning methods in hydrological univariate time series forecasting using large datasets. GP designed the experiments, performed the computations and wrote the manuscript under the supervision of HT and DK during her MSc thesis. All authors have discussed the results and edited the manuscript.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix: Statistical software and supplementary material

The analyses and visualizations have been performed in R Programming Language (R Core Team 2018). We have used the following contributed R packages: cgwtools (Witthoft 2015), devtools (Wickham and Chang 2018), EnvStats (Millard 2013, 2018), forecast (Hyndman and Khandakar 2008; Hyndman et al. 2018), fracdiff (Fraley et al. 2012), gdata (Warnes et al. 2017), ggplot2 (Wickham 2016a; Wickham et al. 2018), HKprocess (Tyralis 2016), kernlab (Karatzoglou et al. 2004, 2018), knitr (Xie 2014, 2015, 2018), nnet (Venables and Ripley 2002; Ripley 2016), plyr (Wickham 2011, 2016b), randomForest (Liaw and Wiener 2002; Liaw 2018), readr (Wickham et al. 2017), rmarkdown (Allaire et al. 2018), rminer (Cortez 2010, 2016) and tidyr (Wickham and Henry 2018).

The supplementary material is available in Papacharalampous and Tyralis (2018). We provide the fully reproducible reports together with their codes. We also provide the reports entitled "Definitions of the stochastic processes'', "Definitions of the forecast quality metrics'' and "Selected figures for the qualitative comparison of the forecasting methods'', which we suggest to be read alongside with Sects. 2.1, 2.4 and 3.1 respectively.

References

Abrahart RJ, See LM, Dawson CW (2008) Neural network hydroinformatics: maintaining scientific rigour. In: Abrahart RJ, See LM, Solomatine DP (eds) Practical hydroinformatics. Springer, Berlin, pp 33–47. https://doi.org/10.1007/978-3-540-79881-1_3

Abrahart RJ, Anctil F, Coulibaly P, Dawson CW, Mount NJ, See LM, Shamseldin AY, Solomatine DP, Toth E, Wilby RL (2012) Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. Prog Phys Geog 36(4):480–513. https://doi.org/10.1177/0309133312444943

Abudu S, Cui C, King JP, Abudukadeer K (2010) Comparison of performance of statistical models in forecasting monthly streamflow of Kizil River, China. Water Sci Eng 3(3):269–281. https://doi.org/10.3882/j.issn.1674-2370.2010.03.003

Ahmed NK, Atiya AF, GayarAn NE, El-Shishiny H (2010) An empirical comparison of machine learning models for time series forecasting. Econom Rev 29(5–6):594–621. https://doi.org/10.1080/07474938.2010.481556

Akaike H (1974) A new look at statistical model identification. IEEE Trans Autom Control 19(6):716–723. https://doi.org/10.1109/TAC.1974.1100705

Allaire JJ, Xie Y, McPherson J, Luraschi J, Ushey K, Atkins A, Wickham H, Cheng J, Chang W (2018) rmarkdown: dynamic documents for R. R package version 1.10. https://CRAN.R-project.org/package=rmarkdown

Alpaydin E (2010) Introduction to machine learning, 2nd edn. MIT Press, Cambridge

Anctil F, Filion M, Tournebize J (2009) A neural network experiment on the simulation of daily nitrate-nitrogen and suspended sediment fluxes from a small agricultural catchment. Ecol Model 220(6):879–887. https://doi.org/10.1016/j.ecolmodel.2008.12.021

Arcuri A, Fraser G (2013) Parameter tuning or default values? An empirical investigation in search-based software engineering. Empir Softw Eng 18(3):594–623. https://doi.org/10.1007/s10664-013-9249-9

Armstrong JS (2001) Evaluating forecasting methods. In: Armstrong JS (ed) Principles of forecasting. International series in operations research & management science, vol 30. Springer, Boston, pp 443–472. https://doi.org/10.1007/978-0-306-47630-3_20

Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting methods: empirical comparisons. Int J Forecast 8(1):69–80. https://doi.org/10.1016/0169-2070(92)90008-W

Assimakopoulos V, Nikolopoulos K (2000) The theta model: a decomposition approach to forecasting. Int J Forecast 16(4):521–530. https://doi.org/10.1016/S0169-2070(00)00066-2

Atiya AF, El-Shoura SM, Shaheen SI, El-Sherif MS (1999) A comparison between neural-network forecasting techniques-case study: river flow forecasting. IEEE Trans Neural Netw 10(2):402–409. https://doi.org/10.1109/72.750569

Ballini R, Soares S, Andrade MG (2001) Multi-step-ahead monthly streamflow forecasting by a neurofuzzy network model. In: IFSA world congress and 20th NAFIPS international conference, pp 992–997. https://doi.org/10.1109/NAFIPS.2001.944740

Biau G (2012) Analysis of a random forests model. J Mach Learn Res 13(Apr):1063–1095

Biau G, Scornet E (2016) A random forest guided tour. TEST 25(2):197–227. https://doi.org/10.1007/s11749-016-0481-7

Billah B, Hyndman RJ, Koehler AB (2005) Empirical information criteria for time series forecasting model selection. J Stat Comput Simul 75(10):831–840. https://doi.org/10.1080/00949650410001687208

Bontempi G (2013) Machine learning strategies for time series prediction. European Business Intelligence Summer School, Hammamet, Lecture. 2013. https://pdfs.semanticscholar.org/f8ad/a97c142b0a2b1bfe20d8317ef58527ee329a.pdf. Accessed 12 Sept 2018

Box GEP, Jenkins GM (1968) Some recent advances in forecasting and control. J R Stat Soc C Appl 17(2):91–109. https://doi.org/10.2307/2985674

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140. https://doi.org/10.1007/BF00058655

Breiman L (2001a) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324

Breiman L (2001b) Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat Sci 16(3):199–231

Brown RG (1959) Statistical forecasting for inventory control. McGraw-Hill, New York

Carlson RF, MacCormick AJA, Watts DG (1970) Application of linear random models to four annual streamflow series. Water Resour Res 6(4):1070–1078. https://doi.org/10.1029/WR006i004p01070

Cheng CT, Xie JX, Chau KW, Layeghifard M (2008) A new indirect multi-step-ahead prediction model for a long-term hydrologic prediction. J Hydrol 361(1–2):118–130. https://doi.org/10.1016/j.jhydrol.2008.07.040

Cheng KS, Lien YT, Wu YC, Su YF (2017) On the criteria of model performance evaluation for real-time flood forecasting. Stoch Environ Res Risk Assess 31(5):1123–1146. https://doi.org/10.1007/s00477-016-1322-7

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. https://doi.org/10.1007/BF00994018

Cortez P (2010) Data mining with neural networks and support vector machines using the R/rminer tool. In: Perner P (ed) Advances in data mining. Applications and theoretical aspects. Springer, Berlin, pp 572–583. https://doi.org/10.1007/978-3-642-14400-4_44

Cortez P (2016) rminer: data mining classification and regression methods. R package version 1.4.2. https://CRAN.R-project.org/package=rminer

Criss RE, Winston WE (2008) Do Nash values have value? Discussion and alternate proposals. Hydrol Process 22:2723–2725. https://doi.org/10.1002/hyp.7072

De Gooijer JG, Hyndman RJ (2006) 25 Years of time series forecasting. Int J Forecast 22(3):443–473. https://doi.org/10.1016/j.ijforecast.2006.01.001

De Livera AM, Hyndman RJ, Snyder RS (2011) Forecasting time series with complex seasonal patterns using exponential smoothing. J Am Stat Assoc 106(496):1513–1527. https://doi.org/10.1198/jasa.2011.tm09771

De Vos NJ (2013) Echo state networks as an alternative to traditional artificial neural networks in rainfall–runoff modelling. Hydrol Earth Syst Sci 17:253–267. https://doi.org/10.5194/hess-17-253-2013

Fildes R (1992) The evaluation of extrapolative forecasting methods. Int J Forecast 8(1):81–98. https://doi.org/10.1016/0169-2070(92)90009-X

Fraley C, Leisch F, Maechler M, Reisen V, Lemonte A (2012) fracdiff: fractionally differenced ARIMA aka ARFIMA(p,d,q) models. R package version 1.4-2. https://CRAN.R-project.org/package=fracdiff

Gardner ES Jr (1985) Exponential smoothing: the state of the art. J Forecast 4(1):1–28. https://doi.org/10.1002/for.3980040103

Gardner ES Jr (2006) Exponential smoothing: the state of the art—part II. Int J Forecast 22(4):637–666. https://doi.org/10.1016/j.ijforecast.2006.03.005

GRDC (2017) Long-term statistics and annual characteristics of GRDC timeseries data. Online provided by the Global Runoff Data Centre of WMO. Koblenz: Federal Institute of Hydrology (BfG). Date of retrieval 06 Jan 2018. http://www.bafg.de/GRDC/EN/03_dtprdcts/32_LTMM/longtermstat_node.html

Guo J, Zhou J, Qin H, Zou Q, Li Q (2011) Monthly streamflow forecasting based on improved support vector machine model. Expert Syst Appl 38(10):13073–13081. https://doi.org/10.1016/j.eswa.2011.04.114

Gupta HV, Kling H, Yilmaz KK, Martinez GF (2009) Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling. J Hydrol 377(1–2):80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003

Harvey AC (1984) A unified view of statistical forecasting procedures. J Forecast 3(3):245–275. https://doi.org/10.1002/for.3980030302

Haslett J, Raftery AE (1989) Space-time modelling with long-memory dependence: assessing Ireland's wind power resource. J R Stat Soc C Appl 38(1):1–50. https://doi.org/10.2307/2347679

Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York

He Z, Wen X, Liu H, Du J (2014) A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flown in the semiarid mountain region. J Hydrol 509:379–386. https://doi.org/10.1016/j.jhydrol.2013.11.054

Holt CC (2004) Forecasting seasonals and trends by exponentially weighted moving averages. Int J Forecast 20(1):5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

Hong WC (2008) Rainfall forecasting by technological machine learning models. Appl Math Comput 200(1):41–57. https://doi.org/10.1016/j.amc.2007.10.046

Hong T, Fan S (2016) Probabilistic electric load forecasting: a tutorial review. Int J Forecast 32(3):914–938. https://doi.org/10.1016/j.ijforecast.2015.11.011

Hothorn T, Leisch F, Zeileis A, Hornik K (2005) The design and analysis of benchmark experiments. J Comput Graph Stat 14(3):675–699. https://doi.org/10.1198/106186005X59630

Hu J, Liu J, Liu Y, Gao C (2001) EMD-KNN model for annual average rainfall forecasting. J Hydrol Eng 18(11):1450–1457. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000481

Humphrey GB, Maier HR, Wu W, Mount NJ, Dandy GC, Abrahart RJ, Dawson CW (2017) Improved validation framework and R-package for artificial neural network models. Environ Modell Softw 92:82–106. https://doi.org/10.1016/j.envsoft.2017.01.023

Hurvich CM, Tsai CL (1993) A corrected Akaike information criterion for vector autoregressive model selection. J Time Ser Anal 14(3):271–279. https://doi.org/10.1111/j.1467-9892.1993.tb00144.x

Hutter F, Lücke J, Schmidt-Thieme L (2015) Beyond manual tuning of hyperparameters. KI 29(4):329–337. https://doi.org/10.1007/s13218-015-0381-0

Hyndman RJ, Athanasopoulos G (2018) Forecasting: principles and practice. OTexts, Melbourne, Australia. https://otexts.org/fpp2/. Accessed 12 Sept 2018

Hyndman RJ, Billah B (2003) Unmasking the Theta method. Int J Forecast 19(2):287–290. https://doi.org/10.1016/S0169-2070(01)00143-1

Hyndman RJ, Khandakar Y (2008) Automatic time series forecasting: the forecast package for R. J Stat Softw 27(3):1–22. https://doi.org/10.18637/jss.v027.i03

Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Hyndman RJ, Koehler AB, Snyder RD, Grose S (2002) A state space framework for automatic forecasting using exponential smoothing methods. Int J Forecast 18(3):439–454. https://doi.org/10.1016/S0169-2070(01)00110-8

Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2005) Prediction intervals for exponential smoothing using two new classes of state space models. J Forecast 24(1):17–37. https://doi.org/10.1002/for.938

Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with exponential smoothing: the state space approach. Springer, Berlin, pp 3–7. https://doi.org/10.1007/978-3-540-71918-2

Hyndman RJ, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2018) forecast: forecasting functions for time series and linear models. R package version 8.4. https://cran.r-project.org/web/packages/forecast/index.html

Jain SK, Das A, Srivastava DK (1999) Application of ANN for reservoir inflow prediction and operation. J Water Res Plan Man 125(5):263–271. https://doi.org/10.1061/(ASCE)0733-9496(1999)125:5(263)

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004) kernlab—an S4 package for kernel methods in R. J Stat Softw 11(9):1–20

Karatzoglou A, Smola A, Hornik K (2018) kernlab: Kernel-Based Machine Learning Lab. R package version 0.9-27. https://cran.r-project.org/web/packages/kernlab/index.html

Kashyap RL (1982) Optimal choice of AR and MA parts in autoregressive moving average models. IEEE Trans Pattern Anal 4(2):99–104. https://doi.org/10.1109/TPAMI.1982.4767213

Khan MS, Coulibaly P (2006) Application of support vector machine in lake water level prediction. J Hydrol Eng 11(3):199–205. https://doi.org/10.1061/(ASCE)1084-0699(2006)11:3(199)

Kim TW, Valdés JB (2003) Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks. J Hydrol Eng 8(6):319–328. https://doi.org/10.1061/(ASCE)1084-0699(2003)8:6(319)

Kişi Ö (2004) River flow modeling using artificial neural networks. J Hydrol Eng 9(1):60–63. https://doi.org/10.1061/(ASCE)1084-0699(2004)9:1(60)

Kişi Ö (2007) Streamflow forecasting using different artificial neural network algorithms. J Hydrol Eng 12(5):532–539. https://doi.org/10.1061/(ASCE)1084-0699(2007)12:5(532)

Kişi Ö, Cimen M (2011) A wavelet-support vector machine conjunction model for monthly streamflow forecasting. J Hydrol 399(1–2):132–140. https://doi.org/10.1016/j.jhydrol.2010.12.041

Kişi Ö, Cimen M (2012) Precipitation forecasting by using wavelet-support vector machine conjunction model. Eng Appl Artif Intell 25(4):783–792. https://doi.org/10.1016/j.engappai.2011.11.003

Kişi Ö, Shiri J, Nikoofar B (2012) Forecasting daily lake levels using artificial intelligence approaches. Comput Geosci 41:169–180. https://doi.org/10.1016/j.cageo.2011.08.027

Kitanidis PK, Bras RL (1980) Real time forecasting with a conceptual hydrologic model: 2. Applications and results. Water Resour Res 16(6):1034–1044. https://doi.org/10.1029/WR016i006p01034

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Koutsoyiannis D (2010) HESS Opinions "A random walk on water". Hydrol Earth Syst Sci 14:585–601. https://doi.org/10.5194/hess-14-585-2010

Koutsoyiannis D (2011) Hurst–Kolmogorov dynamics and uncertainty. J Am Water Resour Assoc 47(3):481–495. https://doi.org/10.1111/j.1752-1688.2011.00543.x

Koutsoyiannis D, Montanari A (2015) Negligent killing of scientific concepts: the stationarity case. Hydrol Sci J 60(7–8):1174–1183. https://doi.org/10.1080/02626667.2014.959959

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

Krause P, Boyle DP, Bäse F (2005) Comparison of different efficiency criteria for hydrological model assessment. Adv Geosci 5:89–97. https://doi.org/10.5194/adgeo-5-89-2005

Krzysztofowicz R (2001) The case for probabilistic forecasting in hydrology. J Hydrol 249(1–4):2–9. https://doi.org/10.1016/S0022-1694(01)00420-6

Kwiatkowski D, Phillips PCB, Schmidt P, Shin Y (1992) Testing the null hypothesis of stationarity against the alternative of a unit root: how sure are we that economic time series have a unit root? J Econom 54(1–3):159–178. https://doi.org/10.1016/0304-4076(92)90104-Y

Lambrakis N, Andreou AS, Polydoropoulos P, Georgopoulos E, Bountis T (2000) Nonlinear analysis and forecasting of a brackish karstic spring. Water Resour Res 36(4):875–884. https://doi.org/10.1029/1999WR900353

Lanc TL (1992) The importance of input variables to a neural network fault-diagnostic system for nuclear power plants. MSc thesis. https://lib.dr.iastate.edu/rtd/208. Accessed 12 Sept 2018

Legates DR, McCabe GJ Jr (1999) Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resour Res 35(1):233–241. https://doi.org/10.1029/1998WR900018

Liaw A (2018) randomForest: Breiman and Cutler's random forests for classification and regression. R package version 4.6-14. https://CRAN.R-project.org/package=randomForest

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

Lin JY, Cheng CT, Chau KW (2006) Using support vector machines for long-term discharge prediction. Hydrol Sci J 51(4):599–612. https://doi.org/10.1623/hysj.51.4.599

Liong SY, Sivapragasam C (2002) Flood stage forecasting with support vector machines. J Am Water Resour Assoc 38(1):173–186. https://doi.org/10.1111/j.1752-1688.2002.tb01544.x

Lippmann R (1987) An introduction to computing with neural nets. IEEE ASSP Mag 4(2):4–22. https://doi.org/10.1109/MASSP.1987.1165576

Lu K, Wang L (2011) A novel nonlinear combination model based on support vector machine for rainfall prediction. In: Fourth international joint conference on computational sciences and optimization, p 1343. https://doi.org/10.1109/CSO.2011.50

Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Netw Model Anal Health Inform Bioinform 5:18. https://doi.org/10.1007/s13721-016-0125-6

Maier HR, Dandy GC (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environ Modell Softw 15(1):101–124. https://doi.org/10.1016/S1364-8152(99)00007-9

Makridakis S, Hibon M (2000) The M3-competition: results, conclusions and implications. Int J Forecast 16(4):451–476. https://doi.org/10.1016/S0169-2070(00)00057-1

Makridakis S, Hibon M, Lusk E, Belhadjali M (1987) Confidence intervals: an empirical investigation of the series in the M-competition. Int J Forecast 3(3–4):489–508. https://doi.org/10.1016/0169-2070(87)90045-8

Makridakis S, Spiliotis E, Assimakopoulos V (2018) Statistical and machine learning forecasting methods: concerns and ways forward. PLoS ONE 13(3):e0194889. https://doi.org/10.1371/journal.pone.0194889

Marsland S (2011) Machine learning: an algorithmic perspective, 2nd edn. Chapman and Hall, New York

Millard SP (2013) EnvStats: an R package for environmental statistics. Springer, New York

Millard SP (2018) EnvStats: package for environmental statistics, including US EPA guidance. R package version 2.3.1. https://cran.r-project.org/web/packages/EnvStats/index.html

Mishra AK, Desai VR, Singh VP (2007) Drought forecasting using a hybrid stochastic and neural network model. J Hydrol Eng 12(6):626–638. https://doi.org/10.1061/(ASCE)1084-0699(2007)12:6(626)

Moisen GG (2008) Classification and regression trees. In: Jørgensen SE, Fath BD (eds) Encyclopedia of ecology, vol 1. Elsevier. Oxford, UK, pp 582–588

Montanari A, Rosso R, Taqqu MS (1997) Fractionally differenced ARIMA models applied to hydrologic time series: identification, estimation, and simulation. Water Resour Res 33(5):1035–1044. https://doi.org/10.1029/97WR00043

Montanari A, Rosso R, Taqqu MS (2000) A seasonal fractional ARIMA model applied to the Nile River monthly flows at Aswan. Water Resour Res 36(5):1249–1259. https://doi.org/10.1029/2000WR900012

Murphy AM (1993) What is a good forecast? An essay on the nature of goodness in weather forecasting. Weather Forecast 8:281–293. https://doi.org/10.1175/1520-0434(1993)008%3c0281:WIAGFA%3e2.0.CO;2

Murtagh F (1991) Multilayer perceptrons for classification and regression. Neurocomputing 2(5–6):183–197. https://doi.org/10.1016/0925-2312(91)90023-5

Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I—a discussion of principles. J Hydrol 10(3):282–290. https://doi.org/10.1016/0022-1694(70)90255-6

Pai PF, Hong WC (2007) A recurrent support vector regression model in rainfall forecasting. Hydrol Process 21:819–827. https://doi.org/10.1002/hyp.6323

Papacharalampous GA (2016) Theoretical and empirical comparison of stochastic and machine learning methods for hydrological processes forecasting. MSc thesis. http://www.itia.ntua.gr/en/docinfo/1670/. Accessed 12 Sept 2018

Papacharalampous GA, Tyralis H (2018) Supplementary material for the paper "Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes". figshare. https://doi.org/10.6084/m9.figshare.7092824

Papacharalampous GA, Tyralis H, Koutsoyiannis D (2017a) Comparison between stochastic and machine learning methods for hydrological multi-step ahead forecasting: all forecasts are wrong!, European Geosciences Union General Assembly 2017, Vienna, Geophysical Research Abstracts, vol 19, EGU2017-3068-2. https://doi.org/10.13140/RG.2.2.17205.47848

Papacharalampous GA, Tyralis H, Koutsoyiannis D (2017b) Error evolution in multi-step ahead streamflow forecasting for the operation of hydropower reservoirs. https://doi.org/10.20944/preprints201710.0129.v1 (Preprints 2017100129)

Papacharalampous GA, Tyralis H, Koutsoyiannis D (2017c) Forecasting of geophysical processes using stochastic and machine learning algorithms. Eur Water 59:161–168

Papacharalampous GA, Tyralis H, Koutsoyiannis D (2018a) One-step ahead forecasting of geophysical processes within a purely statistical framework. Geosci Lett 5:12. https://doi.org/10.1186/s40562-018-0111-1

Papacharalampous GA, Tyralis H, Koutsoyiannis D (2018b) Predictability of monthly temperature and precipitation using automatic time series forecasting methods. Acta Geophys 66(4):807–831. https://doi.org/10.1007/s11600-018-0120-7

Pappenberger F, Ramos MH, Cloke HL, Wetterhall F, Alfieri L, Bogner K, Mueller A, Salamon P (2015) How do I know if my forecasts are better? Using benchmarks in hydrological ensemble prediction. J Hydrol 522:697–713. https://doi.org/10.1016/j.jhydrol.2015.01.024

Patel SS, Ramachandran P (2015) A comparison of machine learning techniques for modeling river flow time series: the case of upper Cauvery river basin. Water Resour Manag 29(2):589–602. https://doi.org/10.1007/s11269-014-0705-0

R Core Team (2018) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Raghavendra NS, Deka PC (2014) Support vector machine applications in the field of hydrology: a review. Appl Soft Comput 19:372–386. https://doi.org/10.1016/j.asoc.2014.02.002

Ramos MH, Mathevet T, Thielen J, Pappenberger F (2010) Communicating uncertainty in hydro-meteorological forecasts: mission impossible? Meteorol Appl 17(2):223–235. https://doi.org/10.1002/met.202

Ramos MH, Van Andel SJ, Pappenberger F (2013) Do probabilistic forecasts lead to better decisions? Hydrol Earth Syst Sci 17:2219–2232. https://doi.org/10.5194/hess-17-2219-2013

Ripley B (2016) nnet: feed-forward neural networks and multinomial log-linear models. R package version 7.3-12. https://cran.r-project.org/web/packages/nnet/index.html

Sapankevych NI, Sankar R (2009) Time series prediction using support vector machines: a survey. IEEE Comput Intell Mag 4(2):24–38. https://doi.org/10.1109/MCI.2009.932254

Schaefli B, Gupta HV (2007) Do Nash values have value? Hydrol Process 21(15):2075–2080. https://doi.org/10.1002/hyp.6825

Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464. https://doi.org/10.1214/15-AOS1321

Scornet E, Biau G, Vert JP (2015) Consistency of random forests. Ann Stat 43(4):1716–1741

Shabri A, Suhartono (2012) Streamflow forecasting using least-squares support vector machines. Hydrol Sci J 57(7):1275–1293. https://doi.org/10.1080/02626667.2012.714468

Shi Z, Han M (2007) Support vector echo-state machine for chaotic time-series prediction. IEEE Trans Neural Netw 18(2):359–372. https://doi.org/10.1109/TNN.2006.885113

Shmueli G (2010) To explain or to predict? Stat Sci 25(3):289–310. https://doi.org/10.1214/10-STS330

Silver D, Huang A, Maddison C, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529:484–489. https://doi.org/10.1038/nature16961

Sivakumar B (2004) Chaos theory in geophysics: past, present and future. Chaos Solitons Fractals 19(2):441–462. https://doi.org/10.1016/S0960-0779(03)00055-9

Sivapragasam C, Liong SY, Pasha MFK (2001) Rainfall and runoff forecasting with SSA-SVM approach. J Hydroinform 3(3):141–152

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Solomatine DP, Ostfeld A (2008) Data-driven modelling: some past experiences and new approaches. J Hydroinform 10(1):3–22. https://doi.org/10.2166/hydro.2008.015

Sutton CD (2005) Classification and regression trees, bagging, and boosting. Handb Stat 24:303–329. https://doi.org/10.1016/S0169-7161(04)24011-1

Thissen U, Van Brakel R, De Weijer AP, Melssena WJ, Buydens LMC (2003) Using support vector machines for time series prediction. Chemom Intell Lab 69(1–2):35–49. https://doi.org/10.1016/S0169-7439(03)00111-4

Tyralis H (2016) HKprocess: Hurst–Kolmogorov process. R package version 0.0-2. https://CRAN.R-project.org/package=HKprocess

Tyralis H, Koutsoyiannis D (2011) Simultaneous estimation of the parameters of the Hurst–Kolmogorov stochastic process. Stoch Environ Res Risk Assess 25(1):21–33. https://doi.org/10.1007/s00477-010-0408-x

Tyralis H, Koutsoyiannis D (2014) A Bayesian statistical model for deriving the predictive distribution of hydroclimatic variables. Clim Dyn 42(11–12):2867–2883. https://doi.org/10.1007/s00382-013-1804-y

Tyralis H, Koutsoyiannis D (2017) On the prediction of persistent processes using the output of deterministic models. Hydrol Sci J 62(13):2083–2102. https://doi.org/10.1080/02626667.2017.1361535

Tyralis H, Papacharalampous GA (2017) Variable selection in time series forecasting using random forests. Algorithms 10(4):114. https://doi.org/10.3390/a10040114

Valipour M, Banihabib ME, Behbahani SMR (2013) Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. J Hydrol 476(7):433–441. https://doi.org/10.1016/j.jhydrol.2012.11.017

Vapnik VN (1995) The nature of statistical learning theory, 1st edn. Springer, New York. https://doi.org/10.1007/978-1-4757-3264-1

Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999. https://doi.org/10.1109/72.788640

Venables WN, Ripley BD (2002) Modern applied statistics with S, 4th edn. Springer, New York. https://doi.org/10.1007/978-0-387-21706-2

Wang WC, Chau KW, Cheng CT, Qiu L (2009) A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. J Hydrol 374(3–4):294–306. https://doi.org/10.1016/j.jhydrol.2009.06.019

Warnes GR, Bolker B, Gorjanc G, Grothendieck G, Korosec A, Lumley T, MacQueen D, Magnusson A, Rogers J et al (2017) gdata: various R programming tools for data manipulation. R package version 2.18.0. https://CRAN.R-project.org/package=gdata

Wei WWS (2006) Time series analysis, univariate and multivariate methods, 2nd edn. Addison Wesley, Boston

Weijs SV, Schoups G, Van de Giesen N (2010) Why hydrological predictions should be evaluated using information theory. Hydrol Earth Syst Sci 14:2545–2558. https://doi.org/10.5194/hess-14-2545-2010

Wickham H (2011) The split-apply-combine strategy for data analysis. J Stat Softw 40(1):1–29

Wickham H (2016a) ggplot2. Springer, New York. https://doi.org/10.1007/978-3-319-24277-4

Wickham H (2016b) plyr: tools for splitting, applying and combining data. R package version 1.8.4. https://cran.r-project.org/web/packages/plyr/index.html

Wickham H, Chang W (2018) devtools: tools to make developing R packages easier. R package version 1.13.6. https://CRAN.R-project.org/package=devtools

Wickham H, Henry L (2018) tidyr: easily tidy data with 'spread()' and 'gather()' Functions. R package version 0.8.1. https://CRAN.R-project.org/package=tidyr

Wickham H, Hester J, Francois R, Jylänki J, Jørgensen M (2017) readr: read rectangular text data. R package version 1.1.1. https://CRAN.R-project.org/package=readr

Wickham H, Chang W, Henry L, Pedersen TL, Takahashi K, Wilke C, Woo K (2018) ggplot2: create elegant data visualisations using the grammar of graphics. R package version 3.0. https://cran.r-project.org/web/packages/ggplot2/index.html

Witten IH, Frank E, Hall MA, Pal CJ (2017) Data mining: practical machine learning tools and techniques, fourth edition. Elsevier Inc. ISBN:978-0-12-804291-5

Witthoft C (2015) cgwtools: miscellaneous tools. R package version 3.0. https://cran.r-project.org/src/contrib/Archive/cgwtools/

Wolpert DH (1996) The lack of a priori distinctions between learning algorithms. Neural Comput 8(7):1341–1390. https://doi.org/10.1162/neco.1996.8.7.1341

Xie Y (2014) knitr: A comprehensive tool for reproducible research in R. In: Stodden V, Leisch F, Peng RD (eds) Implementing reproducible computational research. Chapman and Hall, New York

Xie Y (2015) Dynamic documents with R and knitr, 2nd edn. Chapman and Hall, New York

Xie Y (2018) knitr: a general-purpose package for dynamic report generation in R. R package version 1.20. https://cran.r-project.org/web/packages/knitr/index.html

Yapo PO, Gupta HV, Sorooshian S (1996) Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. J Hydrol 181(1–4):23–48. https://doi.org/10.1016/0022-1694(95)02918-4

Yaseen ZM, Allawi MF, Yousif AA, Jaafar O, Hamzah FM, El-Shafie A (2016) Non-tuned machine learning approach for hydrological time series forecasting. Neural Comput Appl 30(5):1479–1491. https://doi.org/10.1007/s00521-016-2763-0

Ye M, Neuman SP, Meyer PD (2004) Maximum likelihood Bayesian averaging of spatial variability models in unsaturated fractured tuff. Water Resour Res 40(5):W05113. https://doi.org/10.1029/2003WR002557

Ye M, Meyer PD, Neuman SP (2008) On model selection criteria in multimodel analysis. Water Resour Res 44(3):W03428. https://doi.org/10.1029/2008WR006803

Yevjevich VM (1987) Stochastic models in hydrology. Stoch Hydrol Hydraul 1(1):17–36. https://doi.org/10.1007/BF01543907

Yu X, Liong SY (2007) Forecasting of hydrologic time series with ridge regression in feature space. J Hydrol 332(3–4):290–302. https://doi.org/10.1016/j.jhydrol.2006.07.003

Zambrano-Bigiarini M (2014) hydroGOF: goodness-of-fit functions for comparison of simulated and observed hydrological time series. R package version 0.3-8. https://CRAN.R-project.org/package=hydroGOF

Zhang GP (2001) An investigation of neural networks for linear time-series forecasting. Comput Oper Res 28(12):1183–1202. https://doi.org/10.1016/S0305-0548(00)00033-2

Zhang GP, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14(1):35–62. https://doi.org/10.1016/S0169-2070(97)00044-7