Evaluation of random forests and Prophet for daily streamflow forecasting

Georgia A. Papacharalampous1,* and Hristos Tyralis2,*

1Department of Water Resources and Environmental Engineering, National Technical University of Athens, Zografou, 157 80, Greece
2Air Force Support Command, Hellenic Air Force, Elefsina, 192 00, Greece

*These authors contributed equally to this work.

Correspondence: Georgia A. Papacharalampous (papacharalampous.georgia@gmail.com)

Received: 27 May 2018 – Revised: 19 August 2018 – Accepted: 20 August 2018 – Published: 27 August 2018

Abstract. We assess the performance of random forests and Prophet in forecasting daily streamflow up to seven days ahead in a river in the US. Both the assessed forecasting methods use past streamflow observations, while random forests additionally use past precipitation information. For benchmarking purposes we also implement a naïve method based on the previous streamflow observation, as well as a multiple linear regression model utilizing the same information as random forests. Our aim is to illustrate important points about the forecasting methods when implemented for the examined problem. Therefore, the assessment is made in detail at a sufficient number of starting points and for several forecast horizons. The results suggest that random forests perform better in general terms, while Prophet outperforms the naïve method for forecast horizons longer than three days. Finally, random forests forecast the abrupt streamflow fluctuations more satisfactorily than the three other methods.

1 Introduction

Streamflow forecasting is important due to its engineering-oriented implementation in flood and water resources management. The large variety of relevant applications includes flood and drought prediction, irrigation and reservoir operation applications (see, for example, Zhang et al., 2018). Therefore, improved hydrological forecasts in various time scales can benefit the society. Data-driven, including machine learning, models are commonly used for streamflow (or river discharge and reservoir inflow) forecasting. The latter can be performed by exclusively using observed streamflow data, as in Papacharalampous et al. (2017a, 2018a) and Zhang et al. (2018), or by also using information obtained from predictor variables (e.g., precipitation variables). Such examples are available in Jain et al. (2018), and Tyralis and Papacharalampous (2018). Recent studies by Papacharalampous et al. (2017a, b, 2018a, b, c), and Tyralis and Papacharalampous (2017) suggest that several classical and/or popular forecasting algorithms are mostly equally useful for hydrological applications when exploiting information from past observations only. Improvements may result from the use of suitable predictor variables.

Let \( x_i \) and \( y_i \) denote daily precipitation and mean daily streamflow at day \( i = 1, \ldots, n \). If the observations are known up to day \( k \), then the \( j \)-step ahead forecast is defined as the forecast of the random variable \( y_{k+j} \) obtained by using information up to day \( k \). Herein, we assess the performance of random forests and Prophet for \( j \)-step ahead forecasting. These two models are introduced by Breiman (2001), and Taylor and Letham (2018a) respectively. The former is a popular machine learning technique successfully applied in forecasting competitions. Tyralis and Papacharalampous (2017) optimize its forecasting use when it is exclusively provided with past information for the process to be forecasted, while here additional information for predictor variables is considered. Random forests are also used in data-driven rainfall-runoff applications (e.g., Shortridge et al., 2016; Petty and Dhingra, 2018), which are similar to forecasting applications with the exception that the predictor variables are considered to be known until time \( k+j \) and streamflow until time \( k+j-1 \). Moreover, streamflow prediction applications of random forests can be found in Lima et al. (2015)
20€ A. Papcharalampous and H. Tyralis: Evaluation of random forests and Prophet for daily streamflow forecasting

Figure 1. Mean daily streamflow of Current river at Doniphan, Missouri (longitude: −90.85, latitude: 36.62) for the years 1981–2013.

Figure 2. Sample autocorrelation of the daily streamflow of the Current river and sample cross-correlation with the daily precipitation of the basin. The sample cross-correlation is the estimate of $\text{Corr}[y_i, y_{i+j}]$, where Corr is the cross-correlation function.

and Papcharalampous et al. (2017a, 2018a, b). Prophet is an automatic time series forecasting model, which also allows the incorporation of predictor variables, as well as the computation of prediction intervals. The latter is proposed, for instance, in Tyralis and Koutsoyiannis (2014). Papcharalampous et al. (2018c) investigate the performance of Prophet in monthly temperature and precipitation forecasting without utilizing predictor variables. This is also the way used herein. Since benchmarking forecasting results is essential, we implement a naïve method and a multiple linear regression model alongside with the above outlined sophisticated ones. Our aim is to illustrate important facts about the models for the problem under examination.

2 Data and methods

We forecast the mean daily streamflow of Current River at Doniphan, Missouri (see Fig. 1). The daily precipitation data $x_i$ at the basin and the mean daily streamflow data $y_i$ span in the time period 1981–2013. This dataset was compiled by Addor et al. (2017b, see also the data availability section). The sample autocorrelation $\text{Corr}[y_i, y_{i+j}]$ and the sample cross-correlation $\text{Corr}[y_i, y_{i+j}]$ are presented in Fig. 2. The sample autocorrelation is higher than 0.4 for time lag up to three days, while the sample cross-correlation is higher than 0.4 for time lag up to two days. A correlation equal to 0.4 means that the predictor variable can explain approximately 16% of the variance of the dependent variable in a linear regression model between $y_i$ and $x_i$.

Subsequently we present the forecasting methods of this study, while further implementation details can be found in the code availability section. The forecasts of the naïve benchmark at time $k+j$, $j = 1, \ldots, 7$ are equal to $y_k$, i.e. they are equal to the last observation. The use of this benchmark is documented in Hyndman and Athanasopoulos (2018, Chap. 3.1). Multiple linear regression models are also widely implemented benchmarks (see Solomatine and Ostfeld, 2008). Herein, they are used for benchmarking the results of random forests; therefore, the predictor variables utilized by these two methods are identical. These predictor
variables are reported below together with the justification of their selection. For the same methods, streamflow and precipitation data are pre-processed using the square root, as proposed by Messner (2018), with the aim for them to be normalized.

Prophet is based on the idea of fitting Generalized Additive Models. Its documentation is available in Taylor and Letham (2018a), while details about its software implementation can be found in Taylor and Letham (2018b). We examine three variations of the Prophet model. In the first variation (hereafter named as “Prophet 1”; the remaining variations are named in a similar way) we decompose the streamflow time series up to time $k$ using the STL method (Cleveland et al., 1990) and remove the seasonal component. Then Prophet is fitted to the decomposed time series, it forecasts at times $k + j$, $j = 1, \ldots, 7$ and, finally, the seasonal component is added to the forecast. Prophet 2 is fitted to the streamflow time series up to time $k$, and forecasts at times $k + j$, $j = 1, \ldots, 7$. In this variation the seasonal component is automatically handled by Prophet. Prophet 3 uses the last 30 observed values for fitting.

Literature and technical information on the implementation of random forests is available in Verikas et al. (2011), and Biau and Scornet (2016). Random forests are easy to tune and implement due to the low number of parameters (see also Scornet et al., 2015). Their main parameter is the number of trees. Higher number of trees results in predictions that are more accurate; however, in this case the computation time increases substantially, while there is also an asymptotic limit in the accuracy of the model (see, for example, Biau and Scornet, 2016). We use 100 trees, which is considered as a reasonable and balanced choice regarding their accuracy (with respect to the limit of accuracy) and computational cost (Probst and Boulesteix, 2018). The other parameters are set equal to the default values, as in the implementation by Wright (2018). To forecast 1-step ahead (i.e. to forecast $y_{k+1}$)
we use $x_{k-2}, x_{k-1}, x_k, y_{k-3}, y_{k-2}, y_{k-1}, y_k$ and the month of the $y_{k+1}$ as predictor variables. We also used a lower number of predictor variables and the performance (not presented here for reasons of brevity) was similar. Using more predictor variables would result in considerably higher computation time with little expected gain in performance. We use the month of $y_{k+1}$ for considering the seasonality effect. To forecast $y_{k+2}$ we use again $x_{k-2}, x_{k-1}, x_k, y_{k-3}, y_{k-2}, y_{k-1}, y_k$ and the month of $y_{k+2}$ as predictor variables. We apply the same procedure to forecast $y_{k+j}$, $j = 3, \ldots, 7$. Regarding the training period, if we want to forecast $y_{k+1}$, random forests are fitted using the respective $x_{i-3}, x_{i-2}, x_{i-1}, y_{i-4}, y_{i-3}, y_{i-2}, y_{i-1}$ as predictor variables and $y_i$ as dependent variable for $i = 1, \ldots, k$. Each time that a new forecast is required (i.e., when $i$ increases by 1), the model is trained again, so that it includes the latest information. A similar procedure is followed for longer forecast horizons.

Forecasting is performed for all days in the years 2004–2013. The reason for using 1/3 of the dataset for testing is justified on the ground of the large variability of streamflow explained from climatic and other factors (e.g., Kingston et al., 2006; Li et al., 2018; Tyralis et al., 2018). Testing in an independent set is also a standard practice in the assessment of data-driven models (e.g., Solomatine and Ostfeld, 2008; Elshorbagy et al., 2010a, b; Wu et al., 2014). In particular, for observations up to day $k$ we forecast the streamflow at days $k + j$, $j = 1, \ldots, 7$. We produce forecasts for values of $k$ in {2003-12-21, \ldots, 2013-12-30}. The forecasts are summarized conditional upon the forecasting method and the forecast horizon.

### 3 Results

Section 3 is devoted to the presentation of the results, which emphasizes on the 1-, 4- and 7-day ahead forecasts. In Figs. S1 and S2 (see the Supplement) we present these forecasts in comparison to the observations, while Fig. 3 focuses on the 1-day ahead forecasts. The differences between the methods are better presented in Fig. S2 in the Supplement. This figure zooms in the period 2012–2013. In general, the forecasts of the naïve, multiple linear regression and random forests methods are close to their target values. When the length of the forecast horizon increases, the distance between the observations and the forecasts increases as well. The forecasts of Prophet 1 and 2 are smooth lines, i.e., they do not capture the abrupt streamflow changes. In addition, they lay far from the actual streamflow values. The forecasts of Prophet 3 seem to be in better agreement with the observed streamflow: still, they are worse than those produced by the naïve, multiple linear regression and random forests methods.

In Fig. 4 we present the root mean square errors (RMSE) and root median square errors (RMdSE) for all forecast horizons. Random forests have the lowest RMSE followed by the multiple linear regression, the naïve and Prophet 3 methods for short forecast horizons (with length less than three days). For forecast horizons longer than four days random forests still perform the best, while Prophet 1 and 2 are better than the naïve and Prophet 3 methods. The performance of the naïve, multiple linear regression and random forests methods decreases with increasing length of the forecast horizon and gets stabilized for long forecast horizons due to the reduction of the available information used by the predictor variables. Prophet 1 and 2, on the other hand, seem to have a constant performance for all forecast horizons. In terms of RMdSE the naïve method is better than Prophet 3, which in turn is better than Prophet 1 and 2 for all forecast horizons. The performance of Prophet 1 and 2 is constant regardless of the forecast horizon. Random forests are the best method for the 1-day ahead forecast horizon, and the second best for the
Figure 5. Notched boxplots of the absolute forecast errors of the 1, 4 and 7-step ahead forecasts (a to c) of the daily streamflow of Current river in the period 2004–2013. The x axis of the three graphs has been truncated at 100 m$^3$ s$^{-1}$.

2-day ahead and higher forecast horizons. RMdSE is lower than RMSE for all methods.

To further investigate the above rankings and the difference in the magnitude between RMSE and RMdSE, in Fig. 5 we present the notched boxplots of the absolute errors for the 1-, 4- and 7-day ahead forecast horizons. The medians of the absolute error are similar to the RMdSE values presented in Fig. 4. The boxplots are positively skewed, resulting in higher RMSE than RMdSE values. In addition, the dispersion of absolute errors is higher for longer forecast horizons.

To understand how close the forecasts are to their corresponding observations we present the scatterplots of Fig. 6. For all the methods excluding Prophet 1 and 2 the plots of the linear models fitted between the forecasts and the observations are close to the black line, which corresponds to forecasts equal to the observations, indicating a good performance in 1-day ahead forecasting. The distance between the black line and the other linear regression lines increases,

Figure 6. 1-, 4- and 7-step ahead forecasts (a to c) and their corresponding mean daily streamflow values. The black line corresponds to forecasts equal to observations, while the remaining lines are the plots of the linear regression models fitted between forecasts and observations.
when the forecast horizon increases. The increase is less pronounced for the Prophet 1 and 2 methods.

4 Discussion and conclusions

In summary, the following remarks are important, especially in light of Abrahart et al. (2008) who comment on the need for documenting the performance assessment of data-driven models on the grounds of specific questions. Random forests are a better predictor compared to the multiple linear regression models, while they outperform the naive method in terms of root mean square error. The use of the selected precipitation predictor variables considerably improves the forecasts, probably due to the nature of the examined problem; however, their influence diminishes for forecast horizons longer than four days. This is also expected from the magnitude of autocorrelations and cross-correlations with precipitation, which indicate that precipitation should influence the magnitude of streamflow for some days. The forecasting error of the Prophet 1 and 2 methods (which are fitted to the whole sample) is independent of the forecast horizon. Nevertheless, these two methods perform consistently worse than the other methods in terms of root median square error, while they have a comparable (to the other methods) performance in terms of root mean square error. Furthermore, Prophet exhibits a worse performance than the naive method when it exclusively uses observations from the last 30 days (Prophet 3). Random forests are a good method for obtaining optimal forecasts, while their performance could be further improved by using more predictor variables, e.g. temperature variables. The naive method is also good; therefore, it should be used as a benchmark, in spite of the fact that it is rarely met in the hydrological forecasting literature. The Prophet model should be used for forecasting at long horizons.

We note that this study is among the first implementing random forests and Prophet for streamflow forecasting. We have thoroughly investigated the performance of all methods, looking at their predictive performance at several forecast horizons. The visualization of all aspects helped in better understanding important facts about the models’ performance and, thus, could be used as a guide for the assessment of methods in streamflow forecasting.

Code availability. This paper is easily reproducible using the R Programming Language (R Core Team, 2018). We used the following R packages: bestNormalize (Peterson, 2018), devtools (Wickham et al., 2018b), gdata (Warnes et al., 2017), ggplot2 (Wickham, 2016; Wickham et al., 2018a), gridExtra (Auguie, 2017), knitr (Xie, 2014; 2015; 2018), lubridate (Grolemund and Wickham, 2011; Spinu et al., 2018), prophet (Taylor and Letham, 2018b), ranger (Wright, 2018; Wright and Ziegler, 2017), readr (Wickham et al., 2017), rmarkdown (Allaire et al., 2018), scales (Wickham, 2018), stringi (Gagolewski, 2018), zoo (Zeileis and Grothendieck, 2005; Zeileis et al., 2018).

Data availability. The data used in the present study can be obtained from the CAMELS dataset (Addor et al., 2017a, b; Newman et al., 2014, 2015). The daily precipitation data included in the CAMELS dataset were sourced by Thornton et al. (2014).

Supplement. The supplement related to this article is available online at: https://doi.org/10.5194/advgeo-45-201-2018-supplement.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank the Editor Luke Griffiths and one anonymous reviewer, whose comments have led to the improvement of this paper.

Edited by: Luke Griffiths
Reviewed by: Luke Griffiths and one anonymous referee

References


Elshorbagy, A., Corzo, G., Srinivasulu, S., and Solomatine, D. P.: Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology –
208. A. Papacharalampous and H. Tyralis: Evaluation of random forests and Prophet for daily streamflow forecasting


Wickham, H.: scales: Scale Functions for Visualization, R package version 1.0.0, available at: https://CRAN.R-project.org/package=scales (last access: 16 August 2018), 2018.


