TRABAJO DE FIN DE MASTER

Development of a daily scale hydrological forecasting system for the Júcar river basin

Presentado por

Avesani Federico

Para la obtención del

Master en Ingeniería de Caminos, Canales y Puertos

Curso: 2018/2019

Fecha: 21/03/2019

Tutor: Pulido Velázquez, Manuel Augusto

Cotutor: Macián Sorribes, Héctor

Thanks

First of all, I need to thank the people behind the Erasmus programme, for giving me the opportunity to move to another country to do my Master's thesis. Without its help, it would have certainly been more difficult to do so.

I thank the ECMWF for providing the forecasts data, which use was fundamental for this work.

I thank AEMET and UC for the precipitation and temperature historical data provided for this work (Spain02 v5 dataset, available at http://www.meteo.unican.es/datasets/spain02).

I thank Prof. Andrea Castelletti, of Politecnico di Milano, for putting me in contact with Prof. Manuel Pulido of UPV, the Advisor for my thesis work.

I thank the Advisor prof. Manuel Pulido, for providing me with a very interesting and challenging topic on which to elaborate my work, and for the kind support and all the suggestions given.

I thank the Post-doc and Co-advisor Hector Macian-Sorribes, which followed me during each stage of the development of my work, with incredibly swift and complete replies to my e-mails, despite being always quite busy with other works, and for encouraging me during the work with his positive attitude.

I thank the professor of my Spanish language course Aina Gallart Fos, which helped my process of learning the Spanish language, fundamental during my stay in Spain, with interesting, light and useful lessons. I enjoyed every single moment of it, and I learned a language also thank to her.

Last, but not less important, I have to thank the people who were not directly involved in my work, but nevertheless provided me with significant personal support which contributed to the final work:

The ESN Valencia volunteers, for giving access to Erasmus students to a wide variety of travels, events and activities, which made me enjoy my stay in Spain even more;

The good friends I met here, with whom I spent an amazing semester;

All my other friends, which were always close to me and kept in touch despite the distance;

My family, which has been supporting my university career during more than 5 years;

My girlfriend Natalia, which I met here. She provided me a lot of help learning the Spanish language and support during the thesis work.

Prospectus

The present work aims to develop a prediction system for the future hydrological conditions of the river Júcar. The HBV model and 7 month-lead daily weather forecasts (precipitation and temperature) provided by the ECMWF (European Center for Medium-Range Weather Forecasts) are the pillars of this research. The sub-basins considered are those competent to the reservoirs of Alarcón, Contreras and Bellús. The model is calibrated with historical data. The evaporation time series, input necessary to the model, was calculated with the Thornthwaite equation. The ECMWF forecasts are manipulated through a statistical method of post-processing, the quantile mapping. The post-processing operation is carried out with the help of the software MATLAB: the amount of data to deal with is considerable and several hundreds of lines of code have been written for that scope. The calibrated model is then run with the bias-corrected forecasts and tested over a limited period of time, to assess its predictive ability. The results observed show a very promising picture: the system reliably simulated, month in advance, the general trend of the runoff time series, especially for dry periods.

Summary

Index of figures

Index of tables

CHAPTER 1
INTRODUCTION

1.1 Hydrological forecasting and decision making

Since ENIAC (Electronic Numerical Integrator and Computer) was used to create the first forecasts via computer in 1950, there have been huge developments on climate and meteorological forecasts and natural resources system models, also thanks to the exponential increase in computation capacity. The main push to this development is given by the fact that many of the Earth's natural resources, on which we as humans depend for our survival and development, are finite or, at best, limited. Yet the global population continues to grow, thus generating an ever-increasing demand for safe living space, fertile land, clean air, and especially fresh water. Fresh water availability and abundance has always been, in human history, a critical topic. Currently, more than two billion people live in highly water-stressed areas because of the uneven distribution of freshwater in time and space. Climate change is expected to accelerate water cycles and thereby increase the available freshwater. However, changes in seasonal patterns and increasing probability of extreme events may offset this effect (Ahuja et al., 2007; Luber & McGeehin, 2008). The two extreme conditions, excess - as in floods - and scarcity - as in droughts - have caused and continue to cause relevant damage to the human civilization and to the environment. Anyway, there is great room for improvement in the reduction of the menace of these natural threats and early warnings of incoming extreme events are the first systems needed to initiate the necessary prevention actions. In this view, hydrological forecasts become of critical importance: they are the initiators of a crucial decision-making process which aims to lower the impacts of floods and droughts.

Decision-making is generally speaking a complex process. There is wide literature about it, the possible techniques to tackle it (cost benefit analysis, multi-objective optimization, ELECTRE

methods, etc.), and the uncertainties which are inherent to the procedure (Coello Coello, Lamont, & Van Veldhuizen, 2007; Dupuit, 1844; Ehrgott, 2005; Figueira, Mousseau, & Roy, 2016). Even in the ideal case, when all the necessary information is provided, it is not simple to take a decision, which could result in a non-efficient action, or in an unequal distribution of resources, and consequently possibly in socio-economical conflicts (Cooper, Barbara, Rice, 1972). The latter are often present, especially if the information on which the decision is taken is not sufficiently clear, or certain. Stakeholders could try to manipulate information, in order to change the decision in their favour (Tversky & Kahneman, 1981). For the aforementioned reasons, it is fundamental to obtain satisfactory and reliable information about the system and to work with trained decision-makers.

1.2 Research purpose

The Júcar River Basin, located in eastern Spain, is subject to a typical Mediterranean climate, which results in a very irregular river flow pattern and a difficult water management operation. In addition, the area is characterized by a very dense population and active industrial and agricultural activities, which, due to the high-water demand, furtherly exacerbate the situation of the water resources management. The proposed research consists in the development of a hydrological daily-scale forecasting system specifically calibrated to the upper Júcar river basin, which contributes to the majority of the water resources available and holds the main reservoirs of the system, able to provide adequate forecasts about its future hydrological state. To do so, the seasonal meteorological forecasts provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) SEAS5 forecasting product (daily scale, 7-month lead time), through the Copernicus climate change service (https://climate.copernicus.eu/), after the application of an appropriate bias-correction algorithm, will be used as input to a daily-scale hydrological model based on the rainfall-runoff model HBV. The bias-correction algorithm is applied to the forecasts in order to adjust the behaviour of the meteorological forecasts, generated using a worldwide coverage model, to the local meteorological conditions, using as reference data historical observations from the Spain02 v5 database. The HBV model will be calibrated and validated using historical meteorological and hydrological data of the Júcar River Basin. Then, the model will be fed with the bias-corrected ECMWF forecast data to get the desired run-off predictions. The resulting model will be able to deliver forecasts of the hydrological conditions, which will provide valuable insights about the future hydrological state of the basin, something crucial to improve the quality of the operational decisions made.

1.3 Thesis structure

This work is divided into 7 parts.

- Chapter one is dedicated to a brief introduction to the present research;
- Chapter two provides a deep introspection into the materials and tools used to elaborate the data;
- Chapter three illustrates the detailed procedure followed in the work;
- Chapter four describes the case study: territory, geomorphology, economy, etc;
- Chapter five reports the results obtained and critically discusses them;
- Chapter six resumes the findings of the work, illustrates the conclusions and gives hints on possible future research lines;
- The annex includes more plots of the simulations carried out and the source codes written.

CHAPTER 2
MATERIALS AND TOOLS

2.1 European framework on meteorological and hydrological forecasting

Plenty of services are available all over the world to provide early drought/flood warnings. They are a clear example of two of the many possible uses of hydrological predictions. They are based on models and/or indicators which, using as input meteorological predictions of various forms (short term and long-term, ensemble or deterministic), compute the available water quantity. The EU possesses services for early warning of floods and droughts.

The European Flood Awareness System (EFAS) is the service devoted to increase preparedness for riverine floods across Europe. It was started in response of the disastrous floods in Elbe and Danube rivers in 2002, which confronted the European Commission with non-coherent flood warning information from different sources and of variable quality, complicating planning and organization of aid. The aim of EFAS is to gain time for preparedness measures before major flood events strike both in the Member States as well as on European level. The system runs on both deterministic short-term forecasts and medium-range probabilistic ensemble forecasts, based on past years data. This data is the input to the LISFLOOD hydrological model. This latter is a hybrid between a conceptual and a physical rainfall-runoff model combined with a routing module in the river channel. It consists of 12 different subroutines, which compute the direct runoff from the surface soil and leaching from subsoil to the river catchment. (Bartholmes, Thielen, Ramos, & Gentilini, 2009; Thielen, Bartholmes, Ramos, & De Roo, 2009). EFAS is running fully operational since autumn 2012.

The European drought observatory (EDO) is a service run by the European Commission's Joint Research Centre (JRC) with the aim of early drought warnings and observation. The

monitoring of droughts is based on the analysis of a series of indicators, representing different components of the hydrological cycle (e.g. precipitation, soil moisture, reservoir levels, river flow, groundwater levels) or specific impacts (e.g. vegetation water stress) that are associated with a particular type of drought.

These instruments are based on the climate data provided by the Copernicus programme, the European Union's Earth Observation Programme. Copernicus services are based on information from a dedicated constellation of satellites, known as "Sentinels", as well as tens of third-party satellites known as "contributing space missions", complemented by "in situ" measurement data. The Copernicus Services transform this wealth of satellite and in situ data into value-added information by processing and analysing the data, integrating it with other sources and validating the results. Datasets stretching back for decades and are made comparable and searchable, thus ensuring the monitoring of changes; patterns are examined and used to create better forecasts. These value-adding activities are streamlined through six thematic streams of Copernicus services:

- Atmosphere Monitoring;
- Marine Environment Monitoring;
- Land Monitoring;
- Climate Change;
- Emergency Management;
- Security.

2.2 SEAS5 weather forecasts

The forecasts exploited in this research are generated by the SEAS5 system by ECMWF. SEAS5 forecasts are created using computational models to calculate the evolution of the atmosphere, ocean and land surface starting from an initial state based on observations of the Earth system. Because of its nature of chaotic system, the atmosphere has the property of "forgetting" initial conditions: the importance of the past atmosphere state in determining its current one quickly fades away and after 10 to 15 days it is not relevant, meaning that other mechanisms have to be searched for longer lead-time forecasts. Longer term predictions of the climate are possible due to a number of components in the Earth system - E.g. the Madden-Julian Oscillation (MJO) and the El Niño-Southern Oscillation (ENSO) -, which evolve more slowly than the atmosphere. The SEAS5 system consists of an ocean analysis to estimate the initial state of the ocean, a global coupled ocean-atmosphere general circulation model to calculate the evolution of the ocean and atmosphere, and a post-processing suite to create forecast products from the raw numerical output. The ocean analysis is carried out by the NEMO (Nucleus for European Modelling of the Ocean) ocean model, while the atmospheric component is the ECMWF IFS

(Integrated Forecast System) version 43r1. SEAS5 forecasts with worldwide coverage are open-access available through the Copernicus Climate Change Service (C3S). They offer forecasts since 1981. The hindcast period of the SEAS5 forecasts ranges between 1993 and 2016, and consists on making forecasts of the past in order to set up and adjust the model, as well as to provide the user with data with which the skill of the forecasting system can be assessed. The forecast period ranges between 2017 until now, being continuously updated. The set of forecasts for the hindcast period comprises 25 ensemble members or scenarios, while the set for the forecast period has 51 members. The difference on the size depending on hindcast or forecast is due to computational requirements.

2.3 Bias correction of meteorological forecasts

All models are only abstract representations of reality, and model-generated forecasts are plagued by uncertainties and biases from various sources. Biases can result from the input data (e.g. measurement errors), the estimated model parameters, initial and boundary conditions, model structure or simplifying assumptions. Traditional meteorological forecasts are generated in a deterministic manner, i.e., the forecasts are provided in the form of a single time series. This type of forecasts is inherently incapable of accounting for forecast uncertainty. To assess forecast uncertainty and for different objectives (droughts, floods) the ensemble forecasting approach has gained popularity (Cloke & Pappenberger, 2009; Yang, Zhou, Yu, Krysanova, & Wang, 2015). Ensemble forecasts are generated by running the model (or models) several times with slightly perturbed factors such as model initial condition, model forcing, or model physics. This type of forecasts provides not only the most likely scenario for a given event but also associated quantitative uncertainty information. Studies have shown that ensemble forecasts can improve the forecast accuracy and extend the forecast lead times over deterministic forecast (Krishnamurti et al., 2000; Zappa et al., 2010). Regardless, biases from the forecast model will propagate to each subsequent step of modelling, degrading the overall quality of the resulting forecasts. Hence, it is necessary to apply statistical postprocessing methods to quantify and reduce those uncertainties. A well-constructed postprocessor achieves the following purposes:

- it corrects the biases and dispersion errors in raw forecasts;
- it preserves the predictive skill of the raw forecasts;
- it downscales raw forecasts to the scale of applications (e.g., basin scale);
- it generates ensemble members of interested variables, which preserve the spatiotemporal and intervariable statistical dependency structure.

Different postprocessing methods have been developed for meteorological and other types of forecasts, due to their different statistical properties. For meteorological forecasts such as

surface air temperature and atmospheric pressure, the forecast errors can be represented by Gaussian distributions and can be corrected relatively easily by conventional regression methods. Postprocessing precipitation forecasts is more complicated because the distribution of forecast or observation of precipitation is a mixed discrete/continuous distribution, the forecast error is heteroscedastic and the extreme events are hard to represent because of limited samples (Scheuerer & Hamill, 2015).

Over the recent years, plenty postprocessing methods have been proposed (Li et al., 2017), among them some examples are:

- Quantile mapping: it aims to adjust the forecasts cumulative distribution function to match that of the historical observations (Piani, Haerter, & Coppola, 2010);

- Rank histogram calibration: it is a method designed to calibrate forecasts based on the rank histogram of historical ensemble forecasts. Firstly, the constant biases in raw forecasts are removed. Then, the rank histogram is constructed based on the debiased forecasts and observations in the training datasets (Hamill & Colucci, 2002);

- Ensemble pre-processor: this scheme is an example of a conditional distribution-based method. It applies the conditional distribution formula to single-value (for example, the ensemble mean) raw forecasts to postprocess them. The conditional distribution of observations, given forecasts, combines both the prior climatological information and the model forecasts information. It is called 'pre-processor' because it deals with meteorological forecasts, which are inputs for hydrological models. It is a more complicated system, well suited to variables which present a degree of autocorrelation (J. Schaake et al., 2007);

- Ensemble model output statistics: it is an example of regression-based method. These are convenient tools for modelling the statistical correlation between the predictand (i.e., the observation) and the predictors (i.e., the model forecasts). It is an evolution of classical regression0-based models, as in it takes into account the ensemble spread information to transmit it to the bias-corrected forecasts (Gneiting & Katzfuss, 2014; Gneiting, Raftery, Westveld, & Goldman, 2005).

Quantile mapping is the bias-correction algorithm which has been selected to post-process the weather forecasts datasets employed in this work. Since the algorithm requires observations, it is calibrated and validated on historical weather forecasts (hindcasts) and the corresponding observations. This operation implies the hypothesis that the system will continue to behave in a similar way.

Quantile mapping or cumulative distribution function (CDF) matching is a relatively simple postprocessing method that adjusts the CDF of the forecasts according to that of the observations. It receives its name due to the fact that performing it, each forecast value is

'mapped' to the corresponding quantile in the observation CDF. The exact procedure will be explained in details chapter 4.

Quantile mapping has already been applied, among others, to the postprocessing of precipitation and temperature forecasts (Piani et al., 2010; Thrasher, Maurer, McKellar, & Duffy, 2012; Verkade, Brown, Reggiani, & Weerts, 2013).

2.4 Hydrological forecasts

Beyond providing protection from water shortages and possible flood damage, as mentioned before, proper forecasts of river flows and hydrological discharge can have a substantial economic impact, as this can help to improve the operation of water resource systems with significant benefits from water use and in the contract negotiation and hydropower sales (Harou et al., 2009; Kumar, 2012). Moreover, it is known that the flow of a river directly and indirectly influences its ecosystem: a higher flow results in a higher dilution effect, if any contaminant is present (almost in every catchment, in the modern era); different river flows change the state of river banks, modifying the habitat and thus the composition of the host species; different flow regimes have different sediment transport patterns, etcetera (Cheng et al., 2014; Falkenmark, 2004; Merritt, Letcher, & Jakeman, 2003). In order to take proper water management decisions, it is necessary to possess reliable hydrological forecasts.

2.5 Hydrological models

A mathematical hydrological model is an abstract model that uses mathematical language to describe the behaviour of a hydrological system. A model works receiving a state and an input (data collected or produced by another model) and generating an output. Models are generally composed by relationship and variables. Relationships describe the behaviour of the variables among each other, while variables are abstractions of system parameters of interest, that can be quantified. There are several types of models. The subdivision can be operated according to the structure of the model:

- empirical models**:** they are purely data based, and don't have any physical meaning behind the structure of the model itself. They look for a correlation between state, input and output, without trying to simulate any physical behaviour;
- conceptual models**:** they try to describe the main dynamics occurring in the system, without the zeal of physical models;
- physical models**:** they attempt at describing all dynamics occurring in the system. They are based on physical laws and are usually quite complex.

The relationships constituting the model can have different mathematical forms:

- Linear or non-linear**:** related to the linearity or non-linearity of the equation describing the system;

- Static or Dynamic**:** a dynamic model accounts for time-dependent changes in the state of the system, while a static (or steady-state) model calculates the system in equilibrium;

- Discrete or continuous**:** a discrete model treats objects as discrete, such as the particles in a molecular model or the states in a statistical model, while a continuous model represents the objects in a continuous manner;

- Deterministic or probabilistic (stochastic)**:** a deterministic model is one in which every set of variable states is uniquely determined by parameters in the model and by sets of previous states of these variables; therefore, a deterministic model always performs the same way for a given set of initial conditions. Conversely, in a stochastic model randomness is present, and variable states are not described by unique values, but rather by probability distributions.

Moreover, a hydrological model can operate in two modes: simulation and forecasting. A model used in simulation can be exploited to get information about the current state of a system, or the possible evolution of a system, given different circumstances (e.g., if decision "A" is taken, what will the outcome be). A model used in forecasting mode provides information on the (likely) future evolution of the system, providing different possible future scenarios and their respective probability of occurrence.

Hydrological models are generally in the form of rainfall-runoff models: runoff is proportional to rainfall, with other factors modifying more or less the intensity of the proportionality between the two quantities. Conceptual rainfall-runoff models are widely used tools in hydrology. Contrary to more complex, physically-based models, the required input data are readily available for most applications. In spite of their attractiveness, conceptual models suffer from some fundamental problems. Some model parameters have a physical basis, but since they are effective parameters on the catchment scale, they are hardly measurable in the field. This makes a model calibration inevitable. However, it is often not possible to find one unique "best" parameter set, i.e. different parameters sets give similar good results during a calibration period (Vaze et al., 2010). Parameter uncertainty makes simulations for periods outside the calibration period less reliable. In addition, "model uncertainty" may exist, i.e. an uncertainty about which model to choose.

However, the outstandingly higher simplicity, and the impossibility, at present time, of implementing (and executing) physical-based models of such complex systems, make of conceptual models the widest applied choice in the hydrology field. (Jakeman & Hornberger, 1993; Moore, 2007; Todini, 1996).

 The main parts of a conceptual hydrological model are usually precipitation, snow, soil moisture, subsurface, and ground water models, which collaborate to produce an estimate of the total flow. Usually, a watershed is divided into sub-basins, in order to increase the accuracy in the simulation and to account for spatial heterogeneities: each sub-basin is calibrated separately in the model, obtaining this way different model parameter sets for each sub-section. The runoff from each sub-basin is then connected to the main river and, eventually, to a lake. In the most recent years, research has focused more on stochastic manipulation of data (e.g., ensemble predictions, Monte Carlo analyses, etc) and forecasts quality, calibration and validation, uncertainty quantification, adaptation of the model to particulars situation, etc, rather than on the creation of new model types (Addor, Clark, & Nijssen, 2016; Arsenault, Poulin, Côté, & Brissette, 2014; R. J. Donohue, Roderick, & McVicar, 2010; Randall J. Donohue, Roderick, & McVicar, 2012; Holkje Barendrecht et al., 2018; Jin, Xu, Zhang, & Singh, 2010) .

In this research, a conceptual model will be used to solve a water balance problem. A general water balance equation is:

$$P = R + E + \Delta S \hspace{6cm} \text{Eq. (2.1)}$$

where **P** is precipitation, **R** evapotranspiration, **E** streamflow and Δ**S** is the change in storage (in soil or the bedrock/ground water). Eq. (2.1) uses the principles of conservation of mass in a closed system, whereby any water entering a system (via precipitation), must be transferred into either evaporation, surface runoff (eventually reaching the channel and leaving in the form of river discharge), or stored in the ground.

Use of water balance models, for different purposes (drought forecasting, runoff evaluation, crop yield estimation, climate impact assessment, etc), have been widely explored in the recent past. (Gleick, 1987; Granier, Bréda, Biron, & Villette, 1999; Reynolds et al., 2000; J. C. Schaake, Koren, Duan, Mitchell, & Chen, 1996). Through mass conservation law, the relationship between water inflow, outflow, and storage in a specified catchment is mathematically derived by the water balance model, whose parameters represent the behaviours of the catchment.

2.5.1 Model calibration and validation

Conceptual and empirical hydrological models need a calibration process, which consists in the search for the best performing parameter set, with respect to previously established objective function(s). Up to this day, plenty of methods have been developed with the end of calibration (J. S. Bergstra, Bardenet, Bengio, & Kégl, 2011; Claesen & De Moor, 2015). Each one presents advantages and disadvantages.

The traditional way of performing optimization has been grid search, or a parameter sweep, which is simply an exhaustive searching through a manually specified subset of the parameter

space. This method may require an excessively long optimization time, especially in the case of a high number of degrees of freedom (i.e., a high dimension parameter set) and/or complicated calculation of the objective function.

An evolution of the traditional grid search is the random search, which randomly select the possible parameter combination. It can outperform the previous method, especially if the algorithm is not very sensitive to some of the parameters (J. Bergstra & Bengio, 2012), however, it only partially reduces the problem of dimensionality, and the computation time may still be too long.

Gradient based methods, which evaluate the gradient or the Hessian matrix of the objective or its approximation through finite differences, are way faster than the previously mentioned ones. However, some of them need the objective function to be differentiable, and they may end up in local optimums. Evolutionary algorithms can be exploited for the optimization any problem, with any objective function. Although they perform the optimization very quickly when compared to other methods, they present the inconvenience of a less precise result.

The methods can be applied in series, in order to compensate the disadvantages and get to a more precise result more efficiently. An example is the search for the "rough" optimal parametrization with an evolutionary algorithm, followed by a grid search, restricted to a smaller parameter space, to refine the result.

2.5.2 The HBV model

The Hydrologiska Byråns Vattenbalansavedlning (HBV) model used in the proposed research is a widely used water balance model. It is a deterministic conceptual model for runoff simulation, developed by Sten Bergström of the Swedish Meteorological and Hydrological Institute (SMHI) in 1976, and furtherly modified by the same author several times. The capability of the HBV model in conducting hydrological analysis related to the water balance is well known and has been used in more than 30 countries all around the world (Bergström, 1976; Driessen et al., 2010; Engeland & Hisdal, 2009; Kobold & Brilly, 2006; Lindström, Johansson, Persson, Gardelin, & Bergström, 1997; Seibert, 1997; Şorman, Şensoy, Tekeli, Şorman, & Akyürek, 2009).

HBV model structure

The HBV model has a relatively simple structure, divided in 4 subroutines:
- snow accumulation and melt by a degree-day method;
- groundwater recharge and actual evaporation as functions of actual water storage in a soil box;
- groundwater storage and discharge by three linear reservoir equations;

- channel routing by a triangular weighting function.

It is semi-distributed, i.e. it allows to divide the catchment into sub-basins, elevation and vegetation zones. It requires a moderate amount of input data: areal precipitation, temperature and potential evaporation. The first two sets of data are readily available, while the potential evaporation needs to be computed, starting from the soil moisture and the land use. There are several ways to compute this variable, in this research the Thornthwaite equation will be chosen (Thornthwaite, 1948). For a more detailed description of the model, the interested reader can see the original paper by S. Bergstrom (Lindström et al., 1997).

Creation of the .txt files

The HBV light application offers a user-friendly tool to develop HBV models, but requires a specific file format for the input files. Moreover, each catchment must have its own folder, containing the sub-folders *data* and *results*. The *results* folder contains simulation and model parameter search results. The *data* sub-folder contains 2 files:

- The PTQ-file contains time series of daily precipitation [mm/day], temperature [^{o}C] and discharge [mm/day]. The name of the input file is always ptq.txt and the format is as follows:

  a) a header of two lines, the first one contains a name for the catchment, the second line is not used by the program;

  b) Date (YYYYMMDD), precipitation, temperature, discharge in one row per day (separated by tabs).

  Example (Figure 2.1)

```
Alarcon
Date     P[mm/d] T[°C]    Q[mm/d]
20090101         0.90     3.21     0.222
20090102         0.49     4.88     0.227
20090103         10.59    2.74     0.189
20090104         4.92     1.74     0.323
20090105         0.19     -0.08    0.265
20090106         1.00     -1.04    0.168
20090107         0.89     -4.57    0.256
20090108         5.76     -6.88    0.127
20090109         3.43     -3.32    0.178
20090110         0.33     -2.96    0.229
20090111         0.72     -1.86    0.331
20090112         0.00     -2.01    0.179
20090113         1.26     -3.44    0.179
20090114         0.51     -1.60    0.230
20090115         0.00     -1.37    0.231
20090116         0.88     -1.03    0.180
20090117         0.64     0.70     0.180
20090118         1.07     0.70     0.180
20090119         10.43    5.01     0.232
20090120         2.03     0.00     0.232
```

Figure 2.1: example of the ptq.txt file. Source: self-made.

- The evaporation-file contains values for the potential evapotranspiration [mm/day]. Its name is evap.txt and it is composed by one line (header) followed by the values (one value per row, as many values as time steps in the PTQ-file, i.e. one value for each day).

  Example (Figure 2.2):

```
Alarcon evap [mm/d]
6.833476692
11.88137592
5.576675679
3.068467734
0
0
0
0
0
0
0
0
0
0
0
0.953735435
0.958681382
12.7698081
0.000401489
0
```

Figure 2.2: example of the evap.txt file. Source: self-made.

HBV model calibration and validation

The HBV-light software conveniently includes an automatic GAP (Genetic Algorithm and Powell's method) tool for parameter search, which couples an evolutionary algorithm with a gradient-based algorithm, the Powell's method. The Powell's method is useful for calculating the local minimum of any objective functions, even complex and not differentiable ones, because it is not necessary to take derivatives. The strategy adopted by the GAP tool consists in finding first the approximate solution through the evolutionary algorithm, and then refining it with the Powell's method. Within the tool, it is possible to set the parameter range, the probabilities of mutation, crossover and reselection of old characteristics, together with number of individuals and number of model runs. Depending on the size of the dataset, model runs and population size, it is possible to reach convergence of the results within a few minutes of automatic calibration.

<u>Genetic algorithms</u>

The general process followed by genetic algorithms is parallel to that followed by the Darwinian biological evolution, and consists of the following steps:

1. Creation of an initial population of random solutions (i.e., randomly generate tuples of parameter sets, typically 100+);

2. Evaluation of the parameter sets tuples (individuals of the population) through the acquisition of their fitness function;

3. Ranking of the parameter sets tuples by their relative fitness;

4. Replacing of the worst-performing individuals with new ones generated through crossover and mutation. This process consists of randomly selecting the "characteristics", i.e., parameters values, of the remaining population, modify them slightly and unite them in the new individuals;

5. Repeating steps 2-4 until satisfactory algorithm performance is reached or algorithm performance is no longer improving.

Powell's method

Powell's method is employed in parameter search to refine the result, that is, it can be applied after a "rougher" optimization method, to get a more precise parameter set, improving model performance. It finds the local minimum of the objective function, and carries the convenient property of not needing the function to be differentiable. The method entails the following procedure (Powell, 2005):

1. Definition of an initial position x_0;

2. Definition of a set of initial search vectors, typically *n* search vectors which are simply the normals aligned to each axis $\{s_1, \cdots, s_n\}$;

3. Minimization of the objective function by a bi-directional search along each search vector, in turn. The bi-directional line search along each search vector can be done by Golden-section search or Brent's method;

4. If a new minimum x_1 is found, it can be expressed as a linear combination of the search vectors, i.e. $x_1 = x_0 + \sum_{i=1}^{N} a_i s_i$;

5. The new displacement vector $\sum_{i}^{N} a_i s_i$ becomes a new search vector, and is added to the end of the search vector list;

6. The search vector which contributed most to the new direction, i.e. the one which was most successful ($i_d = \arg\max_{i=1}^{N} a_i \|s_i\|$), is deleted from the search vector list;

7. The new set of search vectors is $\{s_1, \cdots, s_{i_d-1}, s_{i_d+1}, \cdots, s_n\}$.

Steps 3-7 are iterated an arbitrary number of times until no significant improvement is made.

2.6 Collected data

In this work, a substantial amount of data has been necessary to calibrate and validate the model/system. The time lag between each observation/forecast entry is one day (daily-scale).

High-resolution precipitation and temperature data have been lumped into the sub-basin scale for each of the studied sub basins. The inflow to each reservoir is not directly measured, instead to obtain it a balance equation in the outlets of each sub-basin (which in all cases correspond to reservoirs) is used, Eq. (2.2):

$$0 = Q_{in} - Q_{out} - \frac{\Delta V}{\Delta t}$$

Eq. (2.2)

Where Q_{in} and Q_{out} are, respectively, the inflow and the outflow in m³/s, ΔV is the reservoir volume variation between two consecutive days [m³] and Δt [s] is the duration of the time gap between two lake volume measurements (86400 seconds). This mass balance equation does not take into account losses: they will be added in this work.

The *Anuario de Aforos* (Gauging Yearbook) and the *Sistema Automático de Información Hidrológica* (SAIH, Automatic Hydrological Information System) service offered by the Júcar Basin Agency provided the daily dammed volume and the average daily discharge from the reservoirs for the analysis period.

Each reservoir area of competence (or sub-basin), that is, the total surface of the territory which run-off ends in the reservoir, has been calculated from a digital terrain model, supplied by the CHJ. This data is necessary in order to compute the area-specific precipitation. The CHJ also provided the mean monthly evaporation rates (for the Alarcón and Contreras reservoirs) and the infiltration rates (only for the Contreras reservoir, since Alarcón and Bellús don't show significant infiltration), necessary to compute the mass balance. No evaporation rate was considered for the Bellús reservoir because, given its relatively small surface, it can be deemed irrelevant with respect to the other mass flows.

Historical precipitation and temperature data have been provided by the Spain02 v5.0 service, offered by the *Grupo de Meteorología de Santander* (Santander Meteorology Group). The service delivers high-resolution daily precipitation and (maximum and minimum) temperature from 1951 to 2015 in a 0.1º (~10km) regular grid for peninsular Spain and the Balearic Islands. A dense network of around 2500 quality-controlled stations was selected from the *Agencia Estatal de Meteorología* (AEMET, Spanish Meteorological Agency) in order to build the gridded products. Daily precipitation records (and the resulting gridded values) for any given day n correspond to the precipitation registered between 0700UTC of day n and 0700UTC of day n+1.

The meteorological forecasts (precipitation and temperature) have been provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) SEAS5 system. The datasets are delivered through the Copernicus Climate Data Store Service. They are in the form of an ensemble, being each ensemble member generated from the same model, changing slightly the initial conditions and boundary conditions from one member with respect to the others. A seasonal forecast is produced on the 1st day of each month and run for 7 months. The set of hindcasts (or re-forecasts) start on the 1st of every month for the years 1981-2016 and

have 25 ensemble members. The forecasts start on the 1st of January 2017 and have 51 ensemble members.

The computation of the actual evaporation requires, among others, daylight duration for each day of the year. This dataset has been retrieved from the website https://ptaff.ca/. The daylight duration refers to the year 2017. It is known that day duration changes during the years, as complex mechanisms are involved (mainly the tidal deceleration effect), but as this phenomenon only affects the day duration in the order of milliseconds each century, it has consequently been promptly ignored, and the 2017 data have been deemed good enough.

CHAPTER 3
PRESENTATION OF THE CASE STUDY

3.1 Location

The Júcar river basin territory *(Demarcación Hidrográfica del Júcar, DHJ),* managed by the
Júcar River Basin Agency (CHJ), is located geographically in the east central end of the Iberian
Peninsula (see Figure 3.1). Its physical boundaries are defined by the *Real Decreto* (Royal
Decree) 125/2007, later modified by the *Real Decreto* 775/2015. It includes:

- The territory of the inter-community river basins;
- The territory of the intra-community river basins comprised between the left bank of
  the Gola del Segura at its mouth and the mouth of the Cenia River, including its basin;
- The endorheic basin of Pozohondo
- The natural endorheism formed by the system that constitute the Quejola, Jardín and
  Lezuza rivers and the Los Llanos area, together with the transitional waters.

Its physiography is described as an interior mountainous area, with higher altitude points and
a coastal area consisting of plains among which are those of Oropesa-Torreblanca, Castellón-
Sagunto, Valencia-La Ribera, Favara-Gandía-Denia. Its maximum level, Peñarroya, is located
in the Iberian System, with an altitude of 2024 meters above sea level.

The Demarcación Hidrográfica del JúcarDHJ (DHJ, Júcar Hydrographic Demarcation)
borders the Ebro and Segura demarcations to the north and south respectively and the Tagus,
Guadiana and Guadalquivir to the west, bordering the east with the Mediterranean. The total
area of the territory of the demarcation is 42,735 km2.

The CHJ DHJ is spread over five autonomous communities: The Valencian Community, with
a surface participation of almost 50% of the total area, followed by the Community of Castilla-
La Mancha, with 37.6%, Aragón with 12.6%, Catalonia with 0.20% and Murcia with 0.15%.

There are 789 municipalities in the CHJ, 751 of which have their urban core located within the basin. Of these, 86% host less than 10000 inhabitants. Table 3.1 shows the distribution by autonomous community and province.

Figure 3.1: Territorial scope of the DHJ. Source: https://www.chj.es.

Table 3.1: surface distribution of the CHJ by autonomous community and province.

Province	Surface [km²]	Autonomous Community	Surface [km²]
Tarragona	88.00	Cataluña	88.00
Teruel	5373.84	Aragón	5373.84
Cuenca	8680.54	Castilla-La Mancha	16089.34
Albacete	7408.80		
Castellón	5785.11	Comunidad Valenciana	21120.13
Valencia	10813.30		
Alicante	4521.72		
Murcia	64.01	Región de Murcia	64.01
Total	42735.32	-	42735.32

3.2 State of the art

The Júcar River Basin, due to its precarious conditions and on the push of the European Water Framework Directive (WFD), has been subject to a number of studies, which investigated, among others, the aspects of hydrological conditions, water allocation, water quality. In this vast amount of research, the hydrological studies, which are those of interest for this work, have focused on:

- Simulation of water quality, under different release policies and with historical data, through the application of a water quality model and a water quantity model: possible use as decision support systems (Ferrer, Pérez-Martín, Jiménez, Estrela, & Andreu, 2012; Paredes-Arquiola, Andreu-Álvarez, Martín-Monerris, & Solera, 2010);
- Influence of different hydrological input data (generated using different data-based models) on the simulation of a water resources management system (C., J., & R., 2007);
- Assessing and forecasting the impacts of global change on Mediterranean rivers, through a joint contribution of different Spanish universities and institutions (the SCARCE project) (Navarro-Ortega et al., 2012);
- Integrated hydro-economic simulation and optimization model at the river basin scale: that is, combining economic management concepts and performance indicators with an engineering-level of understanding of a local hydrologic system (Heinz, Pulido-Velazquez, Lund, & Andreu, 2007);
- Drought monitoring, early warning, planning, and ad-hoc decision support system (Andreu, Ferrer-Polo, Pérez, & Solera, 2009; Haro-Monteagudo, Solera, & Andreu, 2017).

Moreover, there have been pan-European projects on hydrological forecasts which included, as study case, the Júcar river basin, among them SWICCA and EDgE. Project SWICCA (http://swicca.eu/) develops a system of drought indicators designed to estimate and act efficiently in the face of climate change. This system of indicators, however, is focused on climate change scenarios (time range of decades), while the prediction over several months has a limited number of variables and scenarios that do not correspond to the drought indicators currently used by the CHJ. Moreover, having been obtained with hydrological models calibrated on a pan-European scale, the adjustment of this system to the hydrology of the Júcar can be improved (Donnelly et al., 2016). In conclusion, the SWICCA project is not able to accurate and reliable hydrological forecasts for the Júcar river basin.

The European project EDgE (http://edge.climate.copernicus.eu/), currently under development, with the participation of the CHJ and in which the Júcar is a case study, seeks to obtain relevant indicators for the decision-making in the water sector. As in the case of SWICCA, the project focuses on climate change scenarios (in greater detail than SWICCA), while the seasonal prediction part suffers from the same weaknesses as those mentioned for SWICCA and with more limited selection of variables.

In conclusion, there is a lack in the research about a system able to predict the future hydrological state of the river basin: hence the innovation of this research, which proposes a hydrological forecasting system for the Júcar River Basin. This will be achieved using the HBV model, which will be fed with the seasonal meteorological forecasts provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) System5 product, properly bias-corrected through the quantile mapping method. The result will be a reliable and innovative hydrological daily-scale forecasting system for the Júcar River Basin.

3.3 Climate

The climate in the DHJ is a typical Mediterranean climate with warm summers and mild winters. It is situated within the thermo-Mediterranean and meso-Mediterranean bioclimatic zones. The thermal maximums are recorded in the months of July and August, coinciding with the dry season. The average annual temperatures range between 14 and 16.5 ºC. The average annual rainfall is about 500 mm, however there is great spatial variability with values at 300 mm in the southernmost regions, while in other areas it reaches values above 750 mm, as shown in Figure 3.2. Also, during the months of October and November can occur episodes of precipitation of great intensity and short duration, commonly known as "*gota fria*". Most of the surface of the Confederation is covered by very permeable materials that favour the infiltration of the superficial waters of precipitation towards underground strata.
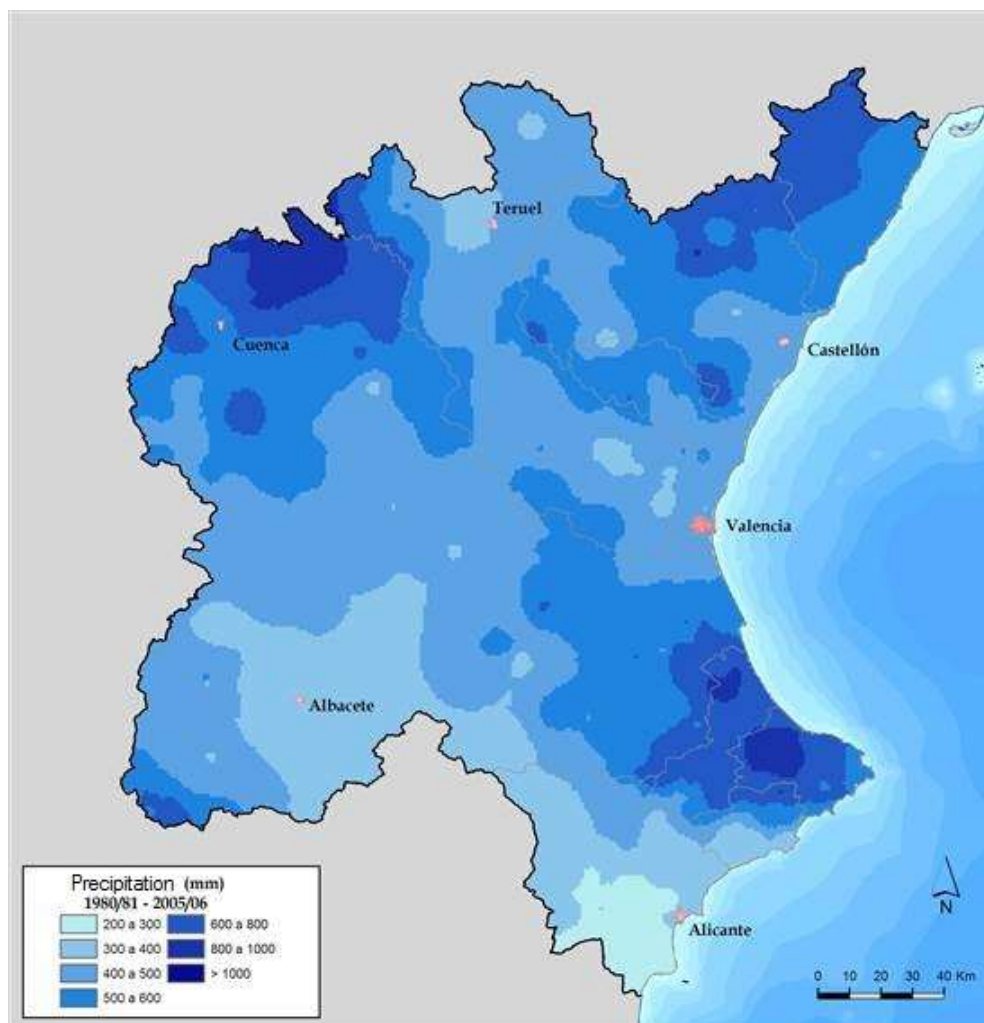
Figure 3.2: map of average annual precipitation in the CHJ from years 1980/1981 to
2005/2006. Source: https://www.chj.es.

3.4 Water resources

The Júcar river basin can be classified as a semi-arid or arid watershed, with large space and
time variability in precipitation and river flows. In addition, water usage is very intense, with
a ratio of used resources to renewable resources close to 1. The basin has suffered several
droughts of different degree recently: they were in the years 1982-1986,1191-1996, 1997-2001,
2004-2008.

Among the several basins included in CHJ territory, the larger ones are the Júcar River Basin
(22378 km2), and the neighbouring Turia River Basin (6913 km2). In the Valencia coastal
plain, where both rivers have their final parts and between both mouths, there is a shallow lake
called Albufera (2300 ha), with an associated wetland (23000 ha). Both the lake and the

wetland depend on return flows from irrigation areas belonging to both basins and on groundwater flows from the coastal aquifer beneath the plain.

The main reservoirs for water supply are Alarcón (1118 hm^3), Contreras (852 hm^3), Tous (379 hm^3). Groundwater plays an important role: the basin has large calcareous aquifers in the north western upper parts, where the rivers are born, providing base flows. Also worth noting in the middle part, is the Mancha Oriental aquifer, that used to provide important base flow to the river, but nowadays it is being overexploited. This is causing the inversion of flows, so the river loses water to the aquifer in spring and summer.

The main rivers of the CHJ are: Cenia, Mijares, Palancia, Turia, Júcar, Serpis and Vinalopó. The Júcar and Turia rivers, with a length of 512 and 280 km respectively, are the most important. The channels that constitute the main fluvial network have a markedly Mediterranean flow regime, characterized by drier periods in summer and growth of circulating flows during the autumn.

The sub-basins investigated in this research are those competent to the reservoirs of Alarcón, Contreras and Bellús, located in the Júcar, the Cabriel (Júcar's tributary) and the Albaida (Júcar's tributary) respectively. They are highlighted in the map in Figure 3.3.
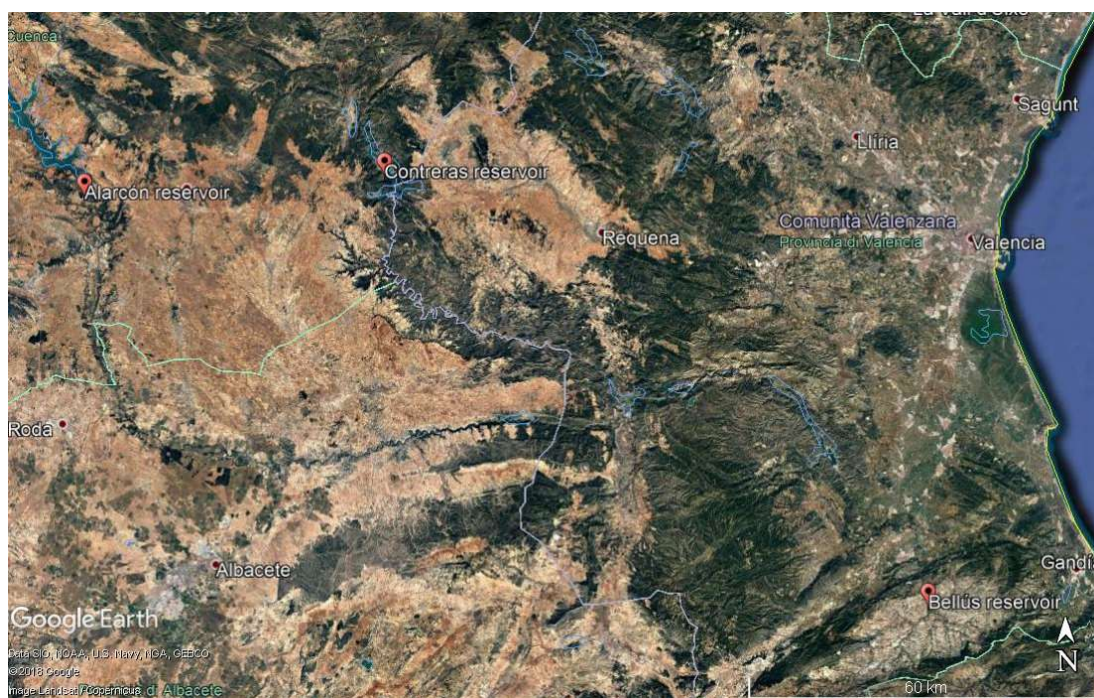
Figure 3.3: map of a portion of the CHJ, with the three reservoirs investigated. Source: Google Earth.

3.5 Water demands

The sectors that contributes the most to the to the total Gross Domestic Product (GDP) in the Júcar river basin are the services and tourism activities. Industry is the second most important sector, after agriculture and energy. However, agriculture, both rainfed and irrigated, is the economic activity that occupies almost half of the surface of the DHJ. The DHJ has an irrigated area representative of the current situation of approximately 390,000 ha, mainly concentrated in the Plana de Castellón, Valencia and the lower basin of the Turia, the Mancha Oriental, the Ribera and the lower basin of the Júcar and the irrigation of the Vinalopó and Monegre valleys.

Accordingly, out of a total water demand of the river basin (year 2017) of 3132 hm³/year, the agrarian one represents almost 80% of the total demand, with 2503 hm³/year.

Regarding industrial activity, the most significant sectors are those corresponding to non-metallic mineral products (which includes the ceramic sector), textiles, clothing, leather and footwear, food, beverages and tobacco. The greatest industrial activity is located in the exploitation systems of Turia, Júcar and Vinalopó-Alacantí. Industrial demand not dependent on urban supply networks (including the manufacturing industry and energy consumptive demand), is 123 hm³/year (almost 4%).

The demand for urban supply is 494 hm³/year, which represents almost 16% of the total. Finally, there is the recreational demand with 12 hm³ / year (<0.5%).

In 2012, the installed capacity in the Demarcation was 13313 MW, 12.11% of the total installed in Spain. Of this power, 8% corresponds to nuclear energy, 36% to other thermoelectric power plants and 56% to renewable energies (38% to wind energy, 7% to solar energy and 11% to hydroelectric power).

CHAPTER 4
METHOD
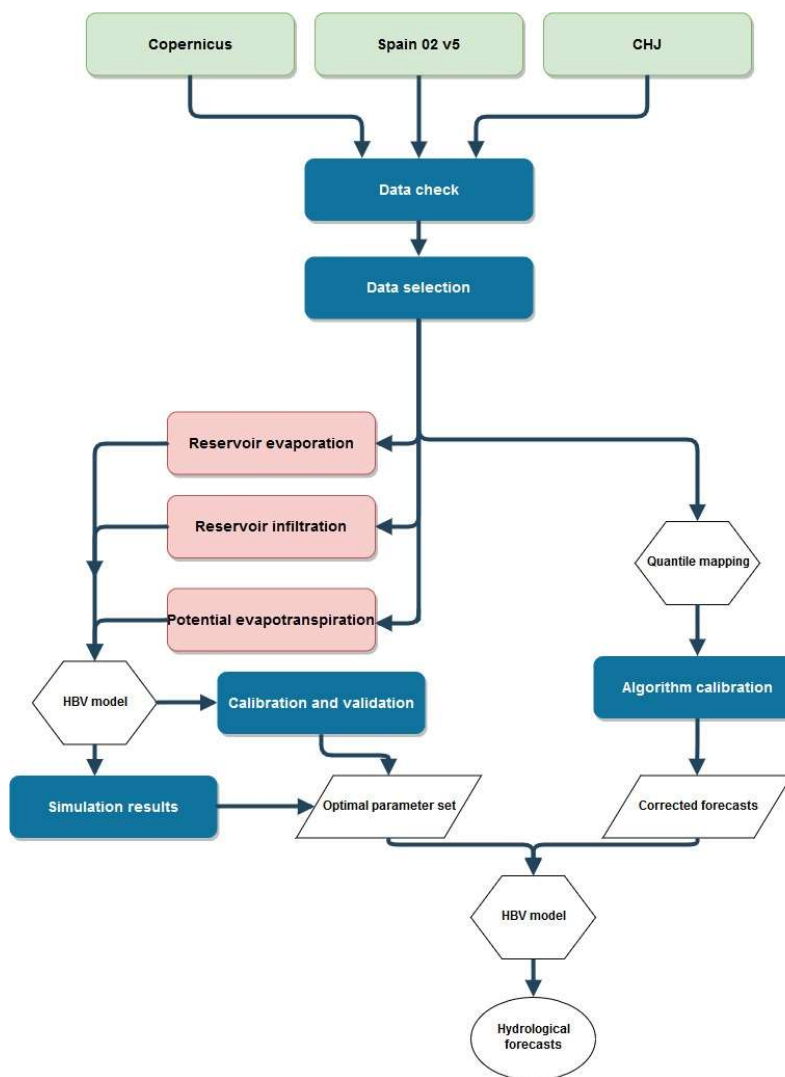
Figure 4.1: schematic representation of the operations carried out. Source: self-made.

4.1 Observation data processing

In this paragraph will follow a detailed description of the techniques, equation and methods used to process the observation data, before feeding it to the HBV-light software. The MS Office EXCEL software has been employed throughout this phase to carry out these operations. The main steps of the process are:

1. Check of the data: search for repeated data and for syntax mistakes, for example negative precipitation values;
2. Data selection: a part of the data has been selected, from 01/01/1995 till 11/12/2013. Older datasets have not been used to avoid the effect of climatic unsteady trends (due to climate change phenomena) and to avoid excessive computational times. Year 1995 has been chosen as the beginning to capture the end of the 1991-1995 Jucar River Basin drought and the subsequent 1997-2001 drought period. Historical meteorological data are available till the end of 2015, but they are of no use, given the lack of complete reservoir data for 2014 and 2015. In the case of the Bellús reservoir, the starting date is 13/01/1998, because the reservoir became operational in that day. This selection of data is only applied in the HBV calibration-validation process: the whole dataset until 2015 will be used in bias-correction;
3. Calculation of the actual inflow in the reservoir: this step is necessary due to the way the inflow is computed, as stated before in Paragraph 2.6. A correction is necessary, considering the non-neglectable contribution of evaporation and infiltration to the mass balance. The actual inflow is then divided by the competent area, to transform it into specific inflow, as required by the model. Paragraph 4.1.1 is devoted to a detailed explanation of this process;
4. Computation of the potential evapotranspiration, with the Thornthwaite formula. Explained in Paragraph 4.1.2;
5. Creation of the .txt files to feed to the model. Paragraph 2.5.2 describes this step.

4.1.1 Actual specific inflow

Evaporation

The evaporation flow (Q_E, m³/s) depends on the evaporation rates (reported for in Table 4.1, obtained from the Júcar River Basin Authority CHJ) and on the reservoir water surface. In line with the procedures used by the CHJ, Bellús evaporation losses have been considered as negligible. Evaporation is calculated by Eq. (4.1):

$$Q_E = R_E \frac{1}{86400 \cdot d} \cdot \frac{1}{1000} S$$

<div align="right">Eq. (4.1)</div>

Where R_E is the evaporation rate [mm m^{-2} month^{-1}], d is the number of days in the month of the interested day and S is the lake surface [m^2]. This latter has been computed using an empirical formula that ties surface to volume, in line with the process used by the CHJ (Eq. (4.2)):

$$S = aV^b$$

<div align="right">Eq. (4.2)</div>

Where V is the lake volume [Mm^3] and α and β empirical coefficients, reported in Table 4.2.

Table 4.1: evaporation rates for the Alarcón and Contreras reservoirs.

	Evaporation rates [mm m^{-2} month^{-1}]	
Month	Alarcón	Contreras
January	30	40
February	40	50
March	94	105
April	99	110
May	142	150
June	210	200
July	262	240
August	238	215
September	154	150
October	84	100
November	42	55
December	23	40

Table 4.2: a and b coefficients for the computation of the surface for the Alarcón and Contreras reservoirs.

Reservoir	a	b
Alarcón	49.94	0.70
Contreras	32.70	0.66

Infiltration

In line with the procedures used by the CHJ, infiltration is neglected in Alarcón and Bellús reservoirs, in the former due to its impervious lake bottom and in the latter due to the impossibility to establish a relationship between storage and infiltration by the lack of enough data. It depends on the lake volume, and is described by Eq. (4.3):

$$Q_I = \gamma V^\delta \frac{1}{86400 \cdot d}$$

<div align="right">Eq. (4.3)</div>

Where Q_I is the infiltration [m³/s], γ and δ empirical coefficients, reported in Table 4.3.

Table 4.3: γ and δ coefficients for the computation of the infiltration flow in the Contreras reservoir.

Reservoir	γ	δ
Contreras	1.15	0.44

Mass balance

After the calculation of infiltration and evaporation, it is possible to write a mass balance equation that computes the actual hydrological discharge (Q_R, m³/s) entering each reservoir (Eq. (4.4)):

$$Q_R = Q_{in} + Q_E + Q_I \qquad \text{Eq. (4.4)}$$

The model requires as input the surface-specific run-off (q_r), in mm/d: this can be obtained by Eq. (4.5):

$$q_r = \frac{Q_R}{A} \cdot 86400 \cdot 1000 \qquad \text{Eq. (4.5)}$$

Where A is the area of the basin draining into the reservoir. The area of the three draining basins are shown in Table 4.4.

Table 4.4: draining basin area for each of the reservoirs considered in the study.

Draining basin	Area [m²]
Alarcón	3.003 E09
Contreras	3.436 E09
Bellús	4.843 E08

4.1.2 Potential evapotranspiration

Evapotranspiration is the grouping o of two processes: evaporation and transpiration. Evaporation is the physical phenomenon for which water goes to the gaseous state. Transpiration is a biological process through which plants release water to the atmosphere. Since the two phenomena are difficult to measure separately and the interesting parameter is the total amount of water dispersed into the atmosphere, they are considered jointly.

Potential evapotranspiration (PET) is defined as the amount of evaporation that would occur if a sufficient water source were available, that is, in soils, when the soil porosity is completely saturated by water. Several methods have been proposed in literature to calculate it, among which the most popular are the Penman model (Penman, 1948) and the Thornthwaite equation (Thornthwaite, 1948). Despite the higher accuracy, the Penman model requires measurements that are often difficult to obtain: wind speed, relative humidity, hours of actual sunlight. Since they were not available for this study with proper spatial resolution, the Thornthwaite method was selected.

Since there is no simple relationship between the monthly temperature and the potential evapotranspiration, because it varies depending on the climate. A "climate index", depending on the number of hours daylight, is then introduced. The process is:

1. Calculation of a monthly heat index i, starting from the monthly average temperature T, illustrated in Eq. (4.6):

$$i = \left(\frac{T}{5}\right)^{1.514} \qquad \text{Eq. (4.6)}$$

2. Sum of the monthly indexes of each year, to get the yearly heat index I, as shown in Eq, (4.7):

$$I = \sum i \qquad \text{Eq. (4.7)}$$

3. Calculation of the daily potential evapotranspiration PET [mm/d] (Eq. (4.8)):

$$PET = 16\frac{L}{12}\frac{N}{30}\left(\frac{10T_d}{I}\right)^{\alpha} \qquad \text{Eq. (4.8)}$$

Where L is the number of hours of daylight, which depends on the location of each sub-basin, N is the number of days of the month, T_d is the average daily temperature and the coefficient α is obtained through Eq. (4.9):

$$\alpha = 6.75 \cdot 10^{-7} \cdot I^3 - 7.71 \cdot 10^{-5} \cdot I^2 + 1,792 \cdot 10^{-3} \cdot I + 0.49239 \qquad \text{Eq. (4.9)}$$

Note: when the average daily temperature T_d is equal or lower than zero, the potential evapotranspiration is set equal to null.

4.2 Model calibration and validation

Model calibration and validation consist in adjusting model parameters in such a way that the model output is as close as possible to reality. The way in which the parameter sets are generated will be explained below in this section. This process is necessary, because the model is of the conceptual type, and every case study is different from another. It is performed with the historical datasets, split into two subsets containing about 4/5 and 1/5 of the whole dataset, respectively for calibration and validation. The adopted principle is to find, among the parameters sets that perform best on the calibration subset, the best one for the validation subset.

The HBV-light software includes an automatic calibration tool, which has been exploited for the calibration of the sub-basins. This tool feats a system consisting of a genetic algorithm optimization followed by a Powell algorithm for local optimization (GAP tool), to find the best performing parameter set. The best performing parameter sets are found for the two datasets, compared between them, and modified till a compromise is found: a parameter set that performs close to optimum in calibration and that does not decrease validation performance

significantly. The selected objective function is the Nash-Sutcliffe Efficiency NSE, defined in Eq. (4.10).

$$NSE = 1 - \frac{\sum(Q_{obs} - Q_{sim})^2}{\sum(Q_{obs} - \overline{Q_{obs}})^2}$$

Eq. (4.10)

Where Q_{obs} is the observed discharge, Q_{sim} the simulated discharge and $\overline{Q_{obs}}$ the observed discharge mean value.

The Nash-Sutcliffe Efficiency provides a measure of how well observed flow is replicated by the model, based on how well the plot of observed versus simulated data fits the 1:1 line. Nash–Sutcliffe efficiency can range from $-\infty$ to 1. An efficiency of 1 corresponds to a perfect match of modelled discharge to the observed data. An efficiency of 0 indicates that the model predictions are as accurate as the mean of the observed data, whereas an efficiency less than zero occurs when the observed mean is a better predictor than the model or, in other words, when the residual variance (described by the numerator in the expression above), is larger than the data variance (described by the denominator). Essentially, the closer the model efficiency is to 1, the more accurate the model is. Threshold values to indicate a model of sufficient quality have been suggested between 0.5 < NSE < 0.65 (R. D. Harmel et al., 2013; Ritter & Muñoz-Carpena, 2013). This threshold holds respected in this study for every calibrated sub-basin.

4.2.1 Calibration

Model calibration consists in finding the best performing (according to the NSE function, Eq. (4.10)) parameter set for the calibration dataset. After setting the warm up period to 6 months, the GAP optimization is carried out, with the following process:

1. The first run is carried out with wide parameter search ranges and large number of model runs (each model run corresponds to the trial of one of the parameter sets of the population);

2. The model is run with the best performing parameter set found by the tool;

3. A new GAP-tool run is issued, setting new parameter ranges according to the previous result: the centre of the range will be the value of the best performing parameter in the previous iteration. If any of the parameters in the previous iteration is set to the range limit, the new search range will start from that value;

4. Multiple runs are executed, and the parameters values are then manually modified, in order to explore the interdependency of the parameters and the sensitivity of the model to them. Smaller number of model runs are set, since the best solution is promptly found, to save computational time;

5. Once a few stable solutions are found (since some parameters depend on each other, different combination can yield similar results), the parameter sets are saved.

4.2.2 Validation

The first part of the process is carried out similarly to calibration. Firstly, a number of best performing parameter sets are found, then they are compared with the ones found in calibration. If the results present important differences, the values are adjusted, in such a way to find a parameter set that fits well both the calibration and validation subsets. As a rule of thumb, a satisfactory *NSE* in validation can be up to 0.2 points lower than that obtained in calibration. However, if the performance is not close to that obtained with the ad-hoc parameter search, a better fit will be looked for. In fact, in some cases it is possible to significantly improve the validation performance, while only slightly worsening the calibration's one. The process is carried out with the help of the GAP tool, with the following procedure:

1. Find a few good parameter sets for the validation dataset, in the same way explained for the calibration process;
2. Try the parameter sets selected in the previous calibration process. If any of them provides a satisfactory result, i.e., sufficiently close (0.04-0.05) to that granted by the ad hoc validation subset parameter search, it is selected;

If the result is not satisfactory:

3. Fix, in the GAP search (that is, set the search range to 0), the parameters which have a similar value in both calibration and validation and to which the model does not appear to be sensitive, in order to concentrate the search on more "troublesome" parameters;
4. Try different combination of the left parameters, both in the calibration and validation subsets. A compromise solution should be found, which performs worse than the optimums found for each subset, but still provides a sufficiently good result for both subsets.

4.3 Bias correction of the forecasts

The bias correction process is exploited to remove systematic errors from the forecasts, using observations as reference. In this project, we have used a quantile mapping approach to bias-correct the seasonal forecasts. This process adjusts the cumulative distribution function of the forecasts to the observed one by applying mapping functions between the corresponding quantiles. A quantile mapping algorithm has been implemented with the software MATLAB, deemed a good choice, given the large amount of data to elaborate. In fact, considering the years 1995-2015, each with 25 ensemble members, 7-month lead reforecasts, yields a total of 1'290'000 entries, for each sub-basin, for each variable. The whole dataset will be split into

two subsets. For temperature, the subdivision was 1995-2008/2009-2015, for precipitation 1995-2005/2006-2015. The first subset will be used to find the correction function, the second one to test it. If the performance is not satisfactory in the second subset, different parameters in the correction function search will be tried. The procedure is carried out for each reservoir, for both precipitation and temperature, and it entails the following steps:

1. Data loading, complete with date: for each sub-basin and variable and for observations and reforecasts. The observation datasets include data gathered from 1980 to 2015, while the reforecasts datasets start in 1995 and end in 2015.
2. The highest and lowest value registered in the observations, for each month, are respectively the high and low thresholds for the corrected forecasts of that month;
3. Calculation of the quantile mapping coefficients to be applied to the forecasts;
4. First performance check, visual: creation of a graph of the CDFs of the observations, forecasts and corrected forecasts. The CDFs are referred to the same period of time;
5. Application of the correction to the second subset;
6. Verification of the performance by means of the monthly mean average error.

4.3.1 Data loading

All the data is loaded into the MATLAB workspace from EXCEL spreadsheets, with the help of a purposely made MATLAB function. The whole observation dataset is held by one single sheet, while the hindcasts are divided by year into different files. The hindcasts spreadsheets were adjusted to match the MATLAB format requirements.

The observation data is loaded as is from the spreadsheet, one column holding the date entries and the other the correspondent measured values.

The hindcasts data is loaded in two different modes, by month and by ensemble. In the "by month" matrix each column represents a month and holds all the forecasts targeted at that month and the exact date is not saved. The "by ensemble" matrix, differently, reports the data exactly as it is in the worksheet, with date of issue, target date and corresponding value for each ensemble. Each subsequent year of data is loaded below. Examples are shown for better clarity in Figure 4.2 and Figure 4.3. The first matrix (data by month) will be used to calibrate the bias correction algorithm, while the second will be used to apply the bias-correction and as performance comparison standard.

	1	2	3	4	5	6	7	8	9	10	11	12
1	9.5646	1.2757	3.9381	5.4027	4.1830	11.7788	19.0564	17.4120	17.6470	13.6064	3.8717	4.1205
2	3.7099	1.0896	3.6802	2.5593	5.4644	11.1983	20.5158	17.5114	17.6927	6.2255	12.0729	3.3844
3	6.4710	3.5292	3.8396	2.1663	5.4494	10.1579	17.8208	17.1563	16.0127	5.8442	9.3989	4.5767
4	6.9899	3.6878	1.1465	5.5018	13.7984	17.6754	17.2936	22.1452	14.8904	4.1856	8.2508	3.9813
5	5.8185	3.1925	3.0202	5.5822	13.7840	16.5872	17.9091	15.7138	4.7717	9.6099	3.9337	
6	4.2231	6.6202	1.2675	6.1701	8.6562	16.8868	14.4045	17.4914	16.5775	5.8139	10.1252	4.1290
7	2.3408	4.3565	0.6020	7.3235	10.5100	17.2393	16.1119	20.6854	16.6173	9.1162	9.4144	4.6583
8	3.8178	5.8972	2.5363	5.9496	11.1509	14.3985	13.4519	19.4186	15.7234	9.9587	12.5263	5.6275
9	2.0220	10.6485	1.3480	2.7678	8.1085	12.8382	10.5479	21.9554	16.5240	11.0717	10.3677	2.4954
10	3.1205	9.5326	3.8186	0.8585	5.1563	11.5445	15.9143	19.7174	16.3877	13.1354	8.8327	2.2247
11	3.2217	5.0399	5.3810	2.6730	8.5798	11.3604	17.3679	18.3592	16.9351	9.4077	12.5903	3.9380
12	2.9536	7.7859	1.0256	5.4940	7.7070	14.1797	18.8132	19.6781	17.0697	12.2101	7.2090	4.8854
13	1.4739	1.4298	1.8310	6.1645	10.5023	14.0199	17.2628	17.9623	14.9808	9.9661	6.1286	0.9855
14	5.0720	-0.0326	0.4939	7.6759	12.5856	13.9575	17.3417	16.4123	16.5615	11.0882	3.9235	3.7278
15	2.8869	5.3607	2.6917	7.3735	13.5065	14.5291	18.6847	16.9040	17.1427	7.7257	10.2669	6.3985
16	9.3643	2.2505	0.6435	5.6839	14.9714	14.8120	22.7054	17.7046	15.5610	8.9533	6.5079	7.3935
17	4.6643	3.6982	1.0462	7.2057	13.9250	15.5097	24.1270	19.8638	15.6255	8.7470	10.5632	8.6209
18	3.6861	4.0636	3.6245	5.5260	13.5046	14.0239	23.7100	18.6783	17.0816	11.8108	10.8958	10.7358

Figure 4.2: example of the forecasts data by month matrix loaded in MATLAB. Source: self-made.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	728660	728660	6.4143	7.4705	9.0136	7.5281	9.1929	9.0042	8.2012	6.2360	8.9680	10.0450	6.6934	9.1843	7.7223	6.5394
2	728660	728661	0.0286	0.0062	0.0492	0.0549	0.0232	0.0022	0.1547	0.0029	0.0704	0.0204	0.0100	0.0407	0.0154	0.0081
3	728660	728662	7.9137e-04	0.0450	7.9137e-04	0.0053	0.0511	0.0363	0.0200	0.0171	0.0081	0.0289	0.0062	0.0538	0.0116	0.0062
4	728660	728663	0.0336	0.2952	1.2971e-04	0.0611	0.0168	0.4861	0.5807	0.1754	1.2971e-04	0.0082	0.0042	2.8559	0	0.0860
5	728660	728664	0.6605	0.4171	0.2448	0.1704	0.7269	0.5245	0.4853	0.6982	0.0459	0	0	0.0198	0.0024	0.7046
6	728660	728665	3.9768	0.0830	0.7454	0.0144	3.3823	0.4948	0.7292	1.2749	1.4348	0.1440	0.4529	0.2590	0.0050	0.7513
7	728660	728666	0.0753	0.0311	0.1115	0.0140	0.0203	0.5782	0.0282	0.4761	0.0432	0.0097	0.3262	0.9406	0.0032	0.1828
8	728660	728667	0.0370	0.0952	0.2347	0.0484	0.0277	0.5441	0.1715	0.0039	0	0.0066	0.8719	0.0074	0.0556	0.4411
9	728660	728668	0.0098	0.1189	0.2415	0.0022	0.1921	1.9799	0.2370	0	0.1292	0.1534	1.3683		0.3882	0.0475
10	728660	728669	0.0389	0.0871	0.3666	0.0068	0.0035	0.0566	0.0555	0.0015	0.2071	0.1673	0.0044	0.1061	0.2071	0.3606
11	728660	728670	0.0105	0.0090	0.0555	0.0015	0.0020	0.0376	0.0159	0.0045	0.0036	0.0213	0.3452	0.0118	0.0538	0.1588
12	728660	728671	0	0.0144	0.0151	0	6.6166e-04	0.0097	0	0	0	0.0149	2.3018	0.0143	0.4698	1.7281
13	728660	728672	0.4983	0.0114	-0.0011	0.0106	0.0021	0	0.0038	0	0.4265	0.1530	3.4613	0	1.3872	0.0253
14	728660	728673	0.8561	0	0	0.7934	0.0024	0	0	0	0.6394	0.0124	0.1221	0	0.0024	0
15	728660	728674	11.7474	0.1025	0.0019	0	0.0310	7.9137e-04	0.0174	6.6166e-04	0.0346	3.2879	0.0045	0	0.2140	6.6166e-04
16	728660	728675	6.6937	0.3360	0.0039	7.9137e-04	1.2971e-04	0.7641	0.0019	0	0.2900	1.3249	7.9137e-04	0.0273	0.5651	0.0021
17	728660	728676	0.4443	0.6630	-0.0030	0	6.6166e-04	0.3352	1.1376e-04	0	0.0958	0.4415	0.0071	1.3119	0.9682	0
18	728660	728677	0.5945	0.1078	0	0.0051	7.9137e-04	0.0538	0.0044	0	0.0099	0.0060	0.0096	0.0944	2.7271	0
19	728660	728678	0.0080	0.9716	0.5479	0.0061	0.0412	0.0181	0.0023	0.0256	6.2504	0	0.0173	1.1376e-04	0.2034	0.0128

Figure 4.3: example of the forecasts data by ensemble member, as is in the original EXCEL sheet. The first column holds the issue date, the second the target date and the remaining 25 the 25 ensemble members. Source: self-made.

4.3.2 Bias correction algorithm calibration

After the data loading, the actual bias-correction function can be implemented. Camici et Al. (2013) applied two different scaling strategies, based on linear regression correction and Cumulative Density Function (CDF) matching, to remove systematic differences between satellite and site-specific soil moisture data (Camici, Brocca, Melone, & Moramarco, 2013). A CDF matching approach similar to that implemented by Camici et Al. was implemented in the present research, to bias-correct precipitation and temperature forecasts. It consists in three main processes: finding the correction algorithm, applying it to the forecasts, and measuring the corrected dataset performance, compared to the biased data. The procedure has been applied separately for each month, in order to adequately capture the in-year pattern of meteorological variables.

Search of the correction algorithm

The first part of the process consists in the following steps:

1. Find the lower and upper limits of the monthly CDFs: they correspond, respectively, to the lowest and highest value ever registered in the historical observations in that month, from 1995 to 2015;

2. Select the time range of the first dataset, i.e. the calibration subset;

Then, for each month (all the Januaries, all the Februaries, etc):

3. Sort the data in ascending order;

4. Associate each value of the sorted vectors x_{obs} (observations) and x_{for} (forecasts) to its cumulative probability, respectively f_{obs} and f_{for};

5. Divide the sorted vector into a pre-defined number of subsets n_{pr}. The division is operated by percentile range, e.g. every ten percentile points;

For each percentile range:

6. Linearly interpolate the forecast values x_{for} to the quantiles of the observed values f_{obs}, as shown by Eq. (4.11);

$$x_{int,i} = \frac{(x_{for,i+1} - x_{for,i})}{(f_{for,i+1} - f_{for,i})} f_{obs,i} + x_{for,i} \qquad f_{for,i} < f_{obs,i} < f_{for,i+1} \qquad \text{Eq. (4.11)}$$

7. Calculate the difference x_{diff} between the observed sorted values and the interpolated sorted forecast values;

8. Calculate the coefficients of an *n-th* degree polynomial, that fits the interpolated values x_{int} to the difference x_{diff}: (Eq. (4.12)). This is the end of the calibration process;

$$a_{i,opt} = \arg \min_{i=0}^{N} \left\| x_{diff} - (a_0 + a_1 x_{int} + a_2 x_{int}^2 + \cdots + a_n x_{int}^n) \right\| \qquad \text{Eq. (4.12)}$$

Note: since the first part of the sorted precipitation data is constant and equal to zero, a modification was introduced to the algorithm: when the range of the data is smaller than a pre-set minimum, the polynomial degree would be zero, i.e. the function approximation is a constant.

Apply the correction to the forecasts

The correction coefficients have been saved in a matrix, for each month there are $n_{pr} \cdot (n + 1)$ coefficients, each $(n + 1)$ tuple to be applied to the correspondent percentile range. The thresholds lt_i and ht_i (respectively low and high) of the percentile range i are calculated according to Eq. (4.13a) and Eq (4.13b):

$$lt_i = \min(x_{for,i}) \qquad \text{Eq. (4.13a)}$$

$$ht_i = \max(x_{for,i}) \qquad \text{Eq. (4.13b)}$$

Where $x_{for,i}$ is the subset of the sorted forecasts correspondent to the percentile range i.

The correction equation is reported in Eq. (4.14):

$$y_{corr} = y_{for} + a_0 + a_1 y_{for} + a_2 y_{for}^2 + \cdots + a_n y_{for}^n \qquad \text{Eq. (4.14)}$$

Where y_{for} represents the biased forecasts and the coefficients a_i the solutions found by Eq. (4.12) for the percentile range correspondent to that which the y_{for} value belongs to. As final step, a cut to the highest and lowest values is applied, to improve the efficiency at the extremes. The limits were set as the highest observed value increased by 20% and the lowest observed value decreased by 20%. This operation has been deemed necessary after reviewing the results: the CDF graphs were showing extremes of the corrected forecast well beyond the extremes of the observations.

The last operation, after applying the correction, is trying different modes of the correction algorithm: change the subdivision of the datasets, number of percentile ranges and polynomial degree. Finally, the best performing ones (or, in case of similar performance, the ones which require the lowest computational effort) are chosen.

Performance evaluation

The monthly MAE (Mean Absolute Error) indicator, with respect to the observations, is used to measure the performance of the bias-correction. The monthly time span is selected to avoid a bias towards rainy days for the precipitation variable. If measuring daily error instead of MAE, if a rainy day is wrongly forecasted as dry, or the other way around, the error shows more than dry days correctly forecasted: one bad forecast on a rainy day may offset several good ones on dry days. Before computing the actual MAE, the values – observations, hindcasts and bias-corrected hindcasts - had to be aggregated over each month of data, for each year. Each ensemble member has been treated separately. The MAE is calculated, afterwards, with the following formula, for the month i of the year j, for each ensemble member of the reforecasts and corrected reforecasts (Eq. 4.15)):

$$MAE_{ij} = \left| \overline{x_{obs,ij}} - \overline{x_{for,ij}} \right| \qquad \text{Eq. (4.15)}$$

Where $\overline{x_{obs,ij}}$ is the average of the observed variable through month i of the year j and $x_{for,ij}$ is the average of the forecasted variable through month i of the year j.

Next step is the averaging of the MAEs of the different ensemble members, in order to get a single array. The lower the MAE, the better, and the comparison between the MAEs, of the same month of biased and bias-corrected hindcasts is an indicator of the performance of the bias-correction. If the performance is deemed good enough, the next step is applying the bias correction algorithm for real-time forecasts, for which no observation can be used as comparison beforehand.

4.4 Testing the system

The previously calibrated HBV model is fed with the biased and bias-corrected forecasts, to produce new runoff simulations, which will be compared with the observed data. The model will be run multiple times: 25 times (one for each ensemble member of the forecasts) for each month of the investigated period, the drought of the years 2005-2008. Each run will be input with a dataset of 365 entries (one year of data), of which the first 5 months (150 days) are observation data and the last 7 months (215 days) are forecast data. The 5 months of observations will serve the purpose of model "warm up", i.e. finding the initial conditions, as the HBV light software does not allow to specify initial conditions. The actual simulation is then produced with the forecast data.

The complete process consists in, for each of the sub-basins:

1. Create the input files, for each month: these include the previous 5 months of observations, used in the model "warm up", to obtain the initial conditions, and 7 months of corrected forecasts, starting at the target month;

2. Repeat step 2 for the original forecasts: the input files with the original forecasts will be tested as well, to compare the performance of biased and bias-corrected forecasts;

3. Apply the Thornthwaite method, as described in paragraph 4.1.2, to calculate the potential evapotranspiration. The calculation is carried out with the temperature data of the input files (as in points 2 and 3), for each one of them;

4. Apply the parameters previously found in the calibration-validation process (illustrated in paragraph 4.2) and run the HBV model, feeding it with the precipitation, temperature and potential evapotranspiration time series;

5. Repeat step 5 for each input dataset created in steps 2 and 3.

CHAPTER 5
RESULTS

This chapter is devoted to a detailed illustration and discussion of the results, as obtained by the thorough and comprehensive application of the procedure described in Chapter 4. The order in which the results are reported follows that of the previous chapter, to achieve better readability and clarity. Results are shown with the help of tables, figures and graphs, and critically analysed.

5.1 Observation data processing

After the data collection, this is the first step. It has been chosen to carry out the procedure with the EXCEL software, given the relatively simplicity of the operations performed. A part of the whole dataset has been selected, from 01/01/1995 to 11/12/2013 for the Alarcón and Contreras reservoirs, from 13/01/1998 to 11/12/2013 for the Bellús reservoir.

5.1.1 Sub-basin inflow

The computation of the actual specific inflow requires the calculation of two quantities: evaporation and infiltration. The evaporation flow is derived from Eq. (4.1). Relevant statistical characteristics of the time series, for each reservoir, are reported in Table 5.1. Evaporation losses in Bellús have not considered significant, as explained in paragraph 4.1.1. Figure 5.1 shows an example of the graph of the evaporation time series, for the Alarcón reservoir. A strongly seasonal behaviour is evident.

Table 5.1: main statistical characteristics of the evaporation time series, by reservoir.

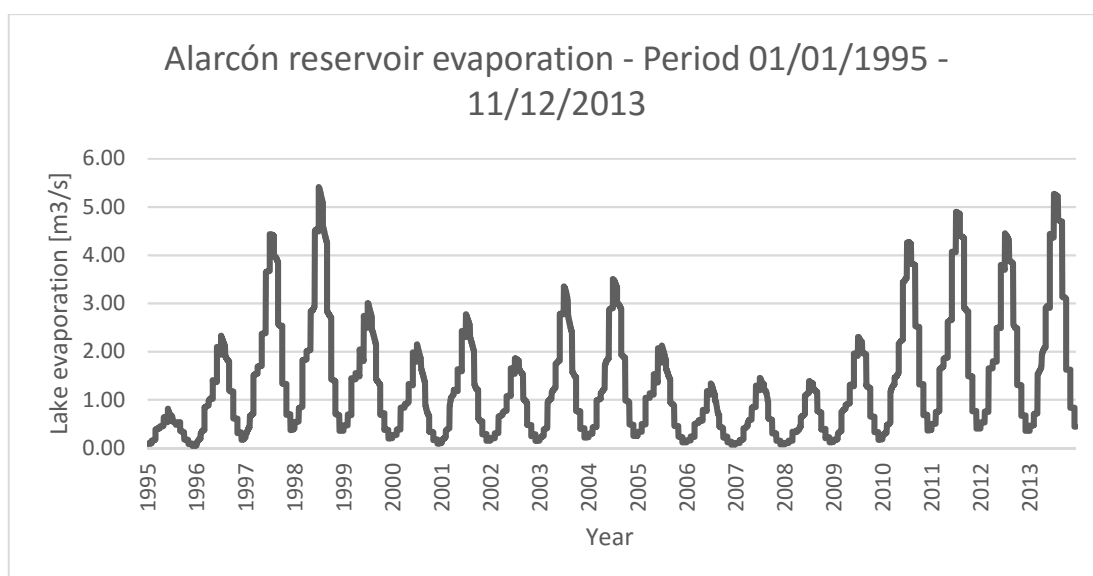| Parameter	Alarcón	Contreras	Bellús
Mean [m³/s]	1.308	0.412	-
Standard deviation [m³/s]	1.190	0.331	-
Skewness [-]	1.344	1.428	-

Figure 5.1: Alarcón reservoir evaporation time series, period 1995-2013. Source: self-made.

Eq (4.3) allows for the calculation of the infiltration. Infiltration losses have only been estimated in Contreras. In Table 5.2 the main statistical characteristics of the time series have been reported. Due to its insignificance (error < 10⁻³) when compared to the actual inflow and the evaporation contributions, the infiltration has been neglected.

Table 5.2: main statistical characteristic of the infiltration time series, for the Contreras reservoir.

Parameter	Value
Mean [m³/s]	3.77 E-06
Standard deviation [m³/s]	1.29 E-06
Skewness [-]	0.381

The application of the mass balance equation as reported in Eq. (4.4) and the normalization to the basin competent area (Eq. (4.5)), yields the corrected specific inflow time series. The graph for one of the reservoirs is shown as an example in Figure 5.2 and the statistical characteristics, as before, are reported in Table 5.3. The drought period of 2005-2008 can be clearly noted in the plot: during those years, the runoff is steadily very low.

Table 5.3: main statistical characteristics of the corrected specific inflow time series, by reservoir.

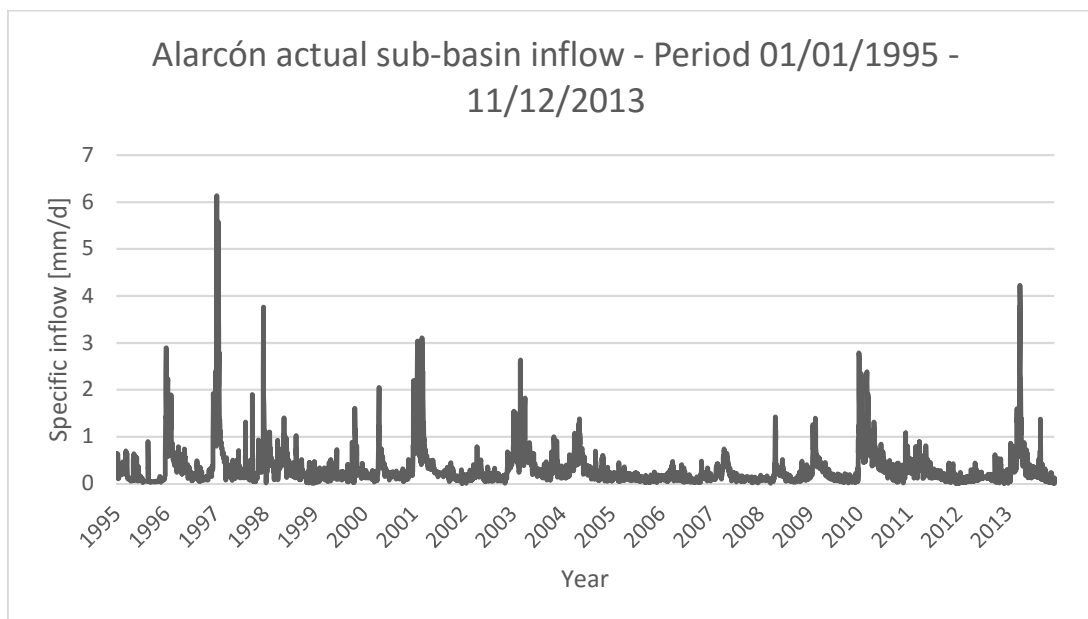| Parameter	Alarcón	Contreras	Bellús
Mean [mm/d]	0.294	0.150747	0.158
Standard deviation [mm/d]	0.367	0.202421	0.746
Skewness [-]	5.052	5.185368	25.162

Figure 5.2: Alarcón reservoir actual sub-basin inflow, period 1995-2013. Source: self-made.

5.1.2 Potential evapotranspiration

The calculation of the potential evapotranspiration (PET) time series, as described in paragraph 4.1.2, requires several steps. The two variables concurring on its calculation are daylight duration and daily temperature.

The final potential evapotranspiration is calculated at the daily scale, for each reservoir, following Eq. (4.8). Main statistical characteristics are reported in Table 5.4 and an example of the time series plot in Figure 5.3 for the Alarcón reservoir.

Table 5.4: main statistical characteristics of the potential evapotranspiration time series, by reservoir.

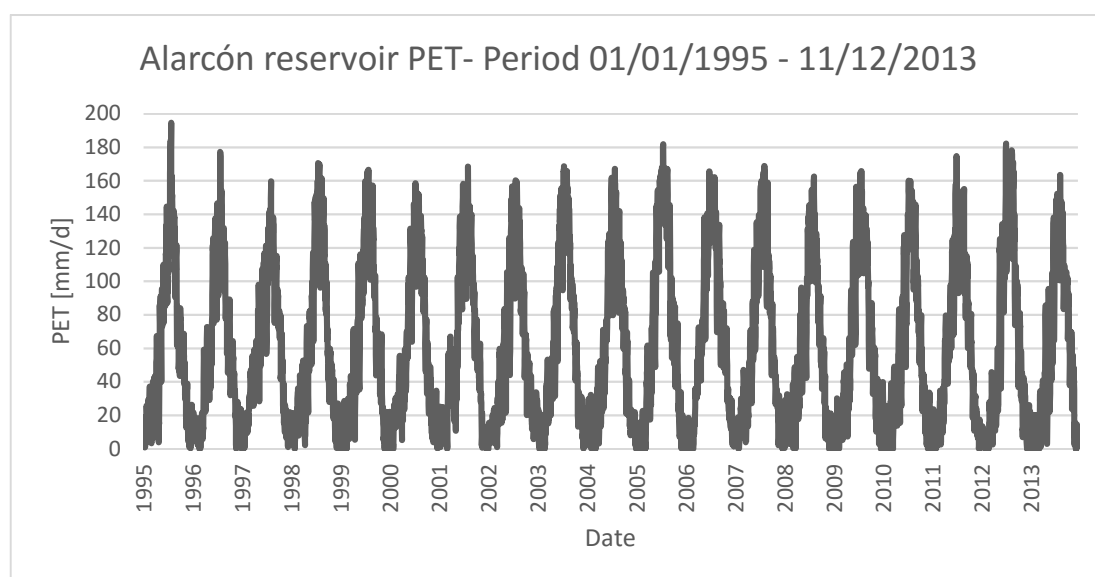| Parameter	Alarcón	Contreras	Bellús
Mean [mm/d]	59.599	54.229	73.473
Standard deviation [mm/d]	48.118	43.444	53.716
Skewness [-]	0.552	0.484	0.564

Figure 5.3: Alarcón sub-basin potential evapotranspiration, period 1995-2013. Source: self-made.

5.2 Model calibration and validation

Table 5.5: final parameter sets selected for each reservoir after the calibration-validation procedure with the GAP-tool of the HBV-light software.

Parameter	Routine	Sensitivity range	Alarcón	Contreras	Bellús
TT		-	-0.540	3.000	1.880
CFMAX		-	2.840	4.533	4.750
SP	Snow	-	1.000	1.000	1.000
SFCF		-	1.000	0.884	0.010
CFR		-	0.050	0.050	0.050
CWH		-	0.100	0.100	0.100
FC		10	350.000	570.000	450
LP	Soil moisture	0.05	0.999	0.999	0.300
BETA		0.02	0.390	0.917	0.920
PERC		0.13	0.860	0.864	5.600
UZL		1÷2	20.000	39.016	8.000
K0	Soil response	0.06	0.700	0.202	0.500
K1		0.01	0.079	0.170	0.500
K2		0.007	0.018	0.012	0.012
MAXBAS	Triangular routing	0.1	3.660	4.227	3.070

Once every dataset necessary to run the model has been produced in the required format, the calibration and validation procedures, as described in paragraph 4.2, can take place. The snow routine parameters did not show a very important effect in the model calibration: in fact, snow is not an important factor in the selected reservoirs, and is rarely present. The K1, K2 and MAXBAS appeared to have a single optimum value range: other explored combination of values would not be performing even close to the optimum. The parameters FC, LP and BETA showed a high interdependency: there were different set of values which returned a similar performance in the simulation, although with different simulation results. The FC value could be varied by 50 or more units, adjusting the others would compensate the loss of performance, but a right combination had to be found. This property was exploited during the calibration-validation process, when there was high disagreement among the calibration and validation parameters. The FC parameter was fixed, and the optimal combination was searched through the GAP tool, until a good compromise was reached. The same occurred with the PERC, UZL and K0 parameters, although on a smaller range of variability: the PERC value could change about by 5 units. The final selected parameter sets are reported, for each reservoir, in Table 5.5. The "sensitivity range" column shows a sensitivity analysis of the parameters: varying the parameter by the respective indicated value would decrease the Nash-Sutcliffe efficiency of about 1%.

The HBV-light software plots:

- The observed temperature;
- The observed precipitation and simulated snow;
- The simulated vs. observed runoff and their total cumulated difference.

A screen capture of the results is shown in Figure 5.4 for the Alarcón sub-basin.

The simulated-observed runoff plot gives a first idea of the goodness of the model: the trend of the observed runoff is very well followed by the simulated time series, despite the noisy signal of the observed data. The peaks are often not very well matched, but this was to be expected: they are single occurrences, with an important random component that can't be expected to be well reproduced. Anyway, there is no apparent dissociation between the general trend: even if high peaks are not well represented, they are simulated with a smaller runoff peak. On the other hand, the dry periods are very well represented: there is no case in which a low observed runoff corresponds to a high simulated one. This means that the model can have a significant importance in simulation and prediction of drought events, and is able to simulate less water-stressed periods.
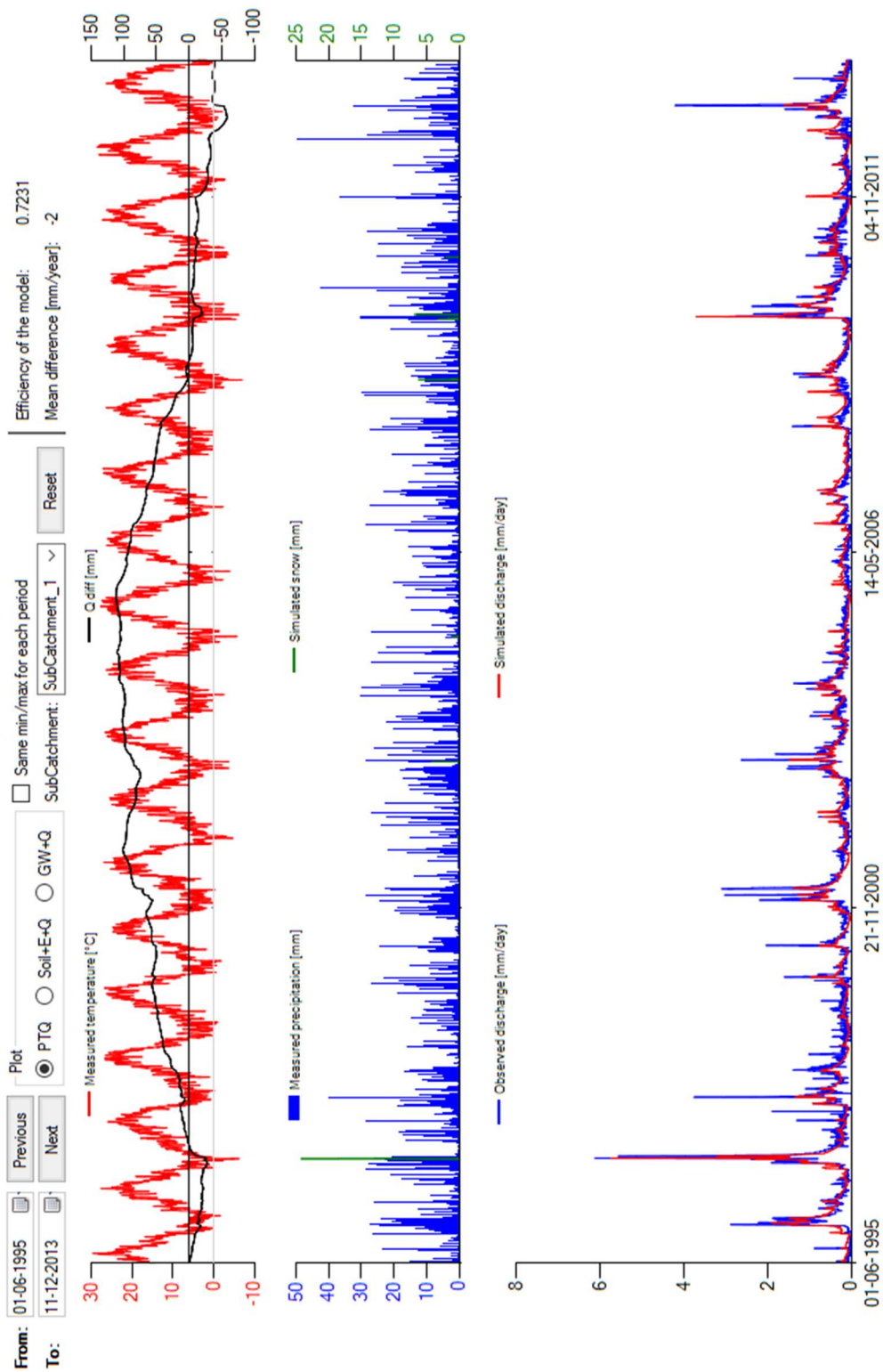
Figure 5.4: screen capture of the results produced by the HBV-light software. Alarcón sub-basin. Source: self-made.

Finally, the HBV-light software produces a report of the performance of the simulation, according to several different objective functions, explained below (Krause, Boyle, & Bäse, 2005; R. D. Harmel et al., 2013):

- Coefficient of determination (R^2): it is defined as the squared value of the coefficient of correlation according to Bravais-Pearson. R^2 can also be expressed as the squared ratio between the covariance and the multiplied standard deviations of the observed and predicted values. Therefore, it estimates the combined dispersion against the single dispersion of the observed and predicted series. The range of R^2 lies between 0 and 1. A value of zero means no correlation at all whereas a value of 1 means that the dispersion of the prediction is equal to that of the observation. A "satisfactory" value for watershed-scale model is R^2>0.6. The observation data is marked as y_i, its mean as $\bar{y}$, the simulated results as f_i and its mean as $\bar{f}$. It is computed through Eq. (5.1):

$$R^2 = \frac{\left(\sum_i (y_i - \bar{y})(f_i - \bar{f})\right)^2}{\sum_i (y_i - \bar{y})^2 \sum_i (f_i - \bar{f})^2}$$

Eq. (5.1)

- Nash–Sutcliffe model efficiency coefficient (NSE): as described in paragraph 4.2. The NSE is calculated according to Eq. (4.10);
- Nash-Sutcliffe efficiency with logarithmic values ($\ln NSE$): to reduce the problem of the squared differences and the resulting sensitivity to extreme values the Nash-Sutcliffe efficiency NSE is often calculated with logarithmic values of y and f. Through the logarithmic transformation of the runoff value the peaks are flattened and the low flows are kept more or less at the same level. As a result, the influence of the low flow values is increased in comparison to the flood peaks resulting in an increase in sensitivity of $\ln NSE$ to systematic model over- or underprediction.
- Flow-weighted Nash efficiency ($wNSE$): all criteria described above quantify the difference between observation and prediction by the absolute values. As a result, an over- or underprediction of higher values has, in general, a greater influence than those of lower values, even if this effect is lower for the $\ln NSE$ indicator. As a solution to this problem, the efficiency can be measured on relative deviations, as in Eq. (5.3):

$$wNSE = 1 - \frac{\sum_i \left(\frac{y_i - f_i}{y_i}\right)^2}{\sum_i \left(\frac{y_i - \bar{y}}{\bar{y}}\right)^2}$$

Eq. (5.3)

Through this modification, the differences between the observed and predicted values are quantified as relative deviations which reduce the influence of the absolute differences during high flows significantly. In addition, the influence of the absolute lower differences during low flow periods are enhanced because they are significant if

looked at relatively. As a result, the relative forms are more sensitive on systematic over- or underprediction, in particular during low flow conditions.

The HBV program is fed with the calibration and validation datasets, and produced the simulation results sought. In Table 5.6 and Table 5.7 the performance obtained by the model is summarized, according to four different objective functions. A significant goodness of fit, for every objective function, is obtained, and the thresholds for a sufficient to good fit are respected in for the NSE (NSE>0.5) and R^2 (R^2>0.6) according to Moriasi et Al. (2015) and Ritter et Al. (2013). As was expected, the flow weighted efficiency shows higher values: the influence of the high flow values is reduced, and since the model simulates very well low flows, they gain a higher importance in the performance assessment. The performance lowers in system validation, as expected, but it does not decrease more than 0.15, indicating a good agreement of the conditions between the calibration and validation subsets.

Table 5.6: goodness of fit functions evaluated for the three considered sub-basins. Calibration subset.

Objective function	Alarcón	Contreras	Bellús
R²	0.769	0.773	0.643
NSE	0.753	0.770	0.594
lnNSE	0.512	0.546	0.256
wNSE	0.818	0.847	0.663

Table 5.7: goodness of fit functions evaluated for the three considered sub-basins. Validation subset.

Objective function	Alarcón	Contreras	Bellús
R²	0.643	0.762	0.620
NSE	0.639	0.727	0.512
lnNSE	0.463	0.508	0.216
wNSE	0.665	0.832	0.599

5.3 Meteorological forecasts bias correction

In this section, the results of the bias correction algorithm described in paragraph 4.3 are illustrated, with the help of plots generated by the MATLAB code. The parameters explored for the bias-correction were:

- Subdivision break-point of the dataset;
- Fitting polynomial degree;
- Number of subdivisions of the whole percentile range.

The calibration algorithm was launched multiple times, trying different combination of these parameters. Every time, the performance was evaluated, comparing the adjusted dataset with the observations, according to the MAE indicators. As a rule of thumb, the simpler the system, the better: a more complicated solution (i.e., higher polynomial degree and/or more percentile ranges) has to be justified with a clear performance improvement.

- Polynomial degree: it has been found that increasing the polynomial degree over two created problems in the fitting, resulting in divergence. The precipitation dataset performed best with a second degree, while for the temperature a simpler first-degree regression was chosen, because no relevant difference existed between both choices;
- Number of percentile ranges: for the selection of the number of percentile ranges, holds the same said for the polynomial degree choice. Moreover, it has been found that splitting the percentile range in more than 20 did not make any difference, and in some cases would even worsen the performance of the system;
- The dataset split seemed to have a more important impact on the overall performance of the system: changing the range by one year would noticeably, even if not dramatically, change the performance.

The final choice for said parameters, which provided the best fit of the corrected time series to the observation, is reported in Table 5.8, for precipitation and temperature.

Table 5.8: final choice of parameters for the bias-correction algorithm.

Parameter	Precipitation	Temperature
Subdivision (first / second)	1995-2005 / 2006-2015	1995-2008 / 2009-2015
Polynomial degree	2	1
Number of percentile ranges	20	10

Figure 5.5 and Figure 5.6 show an example of the biased and bias-corrected temperature CDFs in the same plot as that of the observations, for the two subsets. A significant improvement can be noticed in the matching of the CDFs. There seem to be systematic errors in the original forecasts: they tend to underestimate temperature. The correction algorithm properly adjusts the forecasts to the observation, with an optimal performance in the matching of the CDFs. A slightly lower performance can be observed in the test dataset for the months of February and March: in particular, in the month of February, the original forecasts seems to have a similar performance to that of the bias-corrected ones. However, in the remaining ten months, the bias-correction algorithm keeps a good performance in the test subset.

In Figure 5.7 and Figure 5.8 an example is displayed for the precipitation variable. In this case, as was expected, there seem to be no improvement after the application of the correction algorithm: the CDF of the original forecasts already shows a good match to that of the observations.
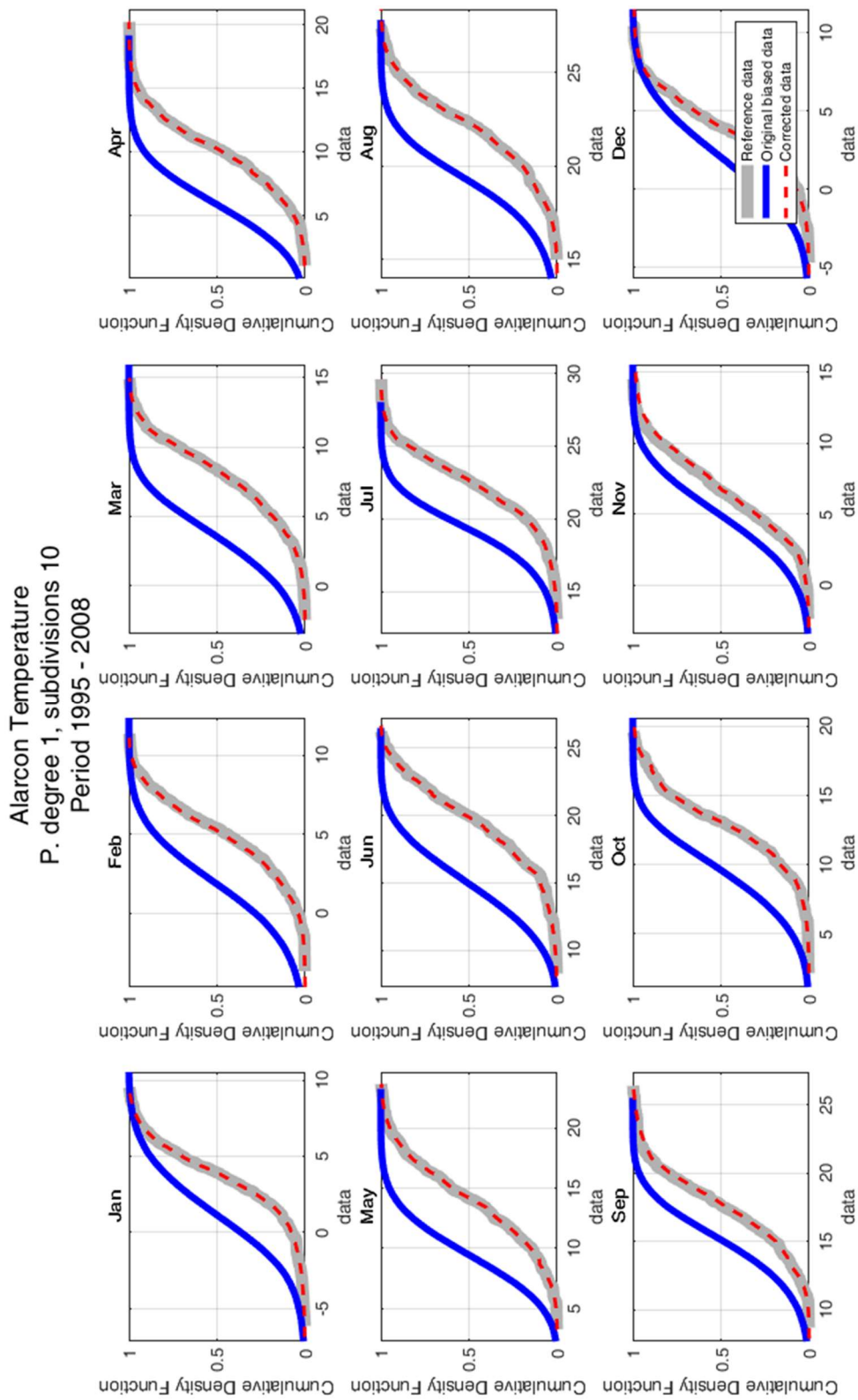
Figure 5.5: CDFs of the observations (grey) compared to the biased (blue) and corrected (dotted red) forecasts. Alarcón reservoir, first temperature subset. Source: self-made.
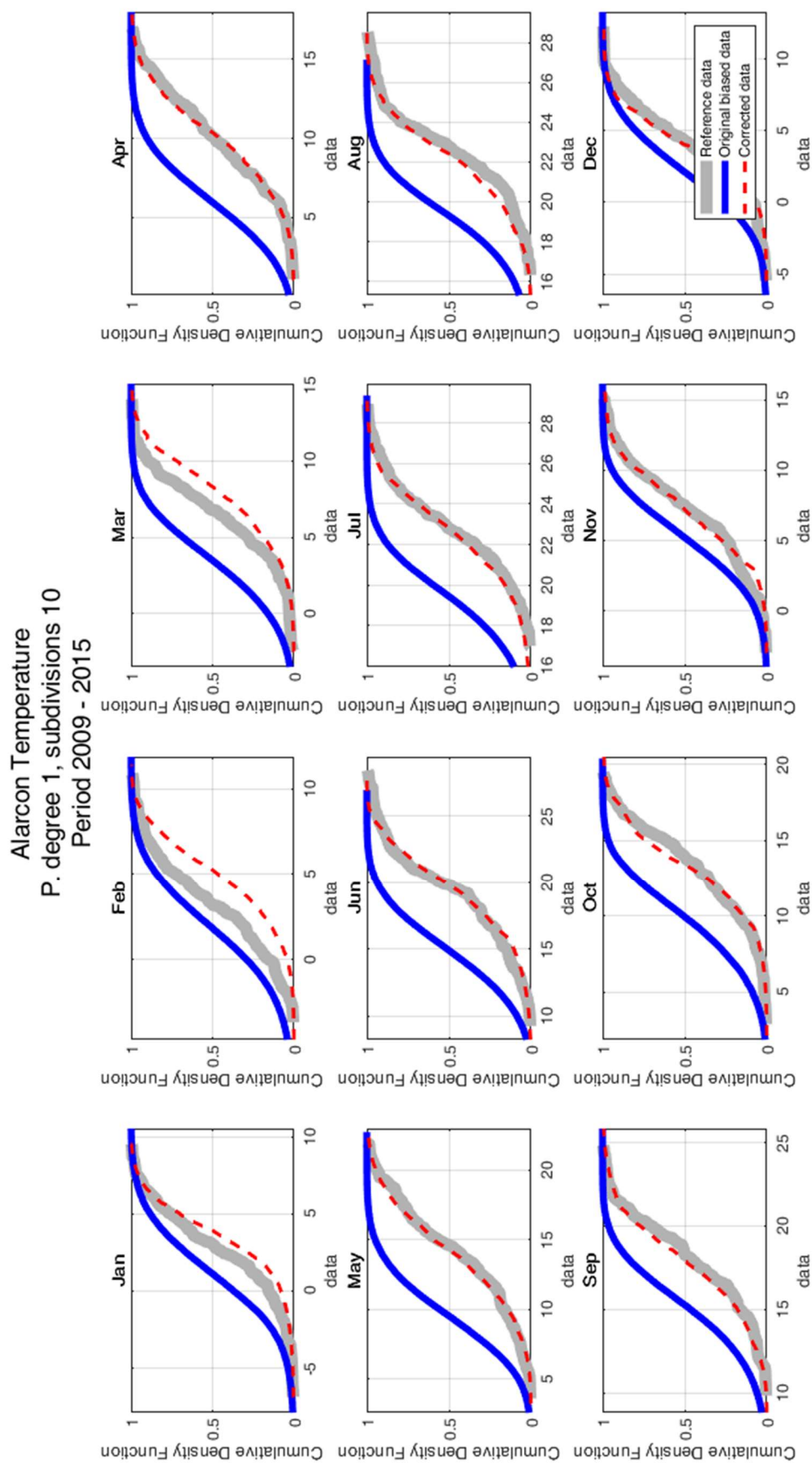
Figure 5.6: CDFs of the observations (grey) compared to the biased (blue) and corrected (dotted red) forecasts. Alarcón reservoir, second temperature subset. Source: self-made.
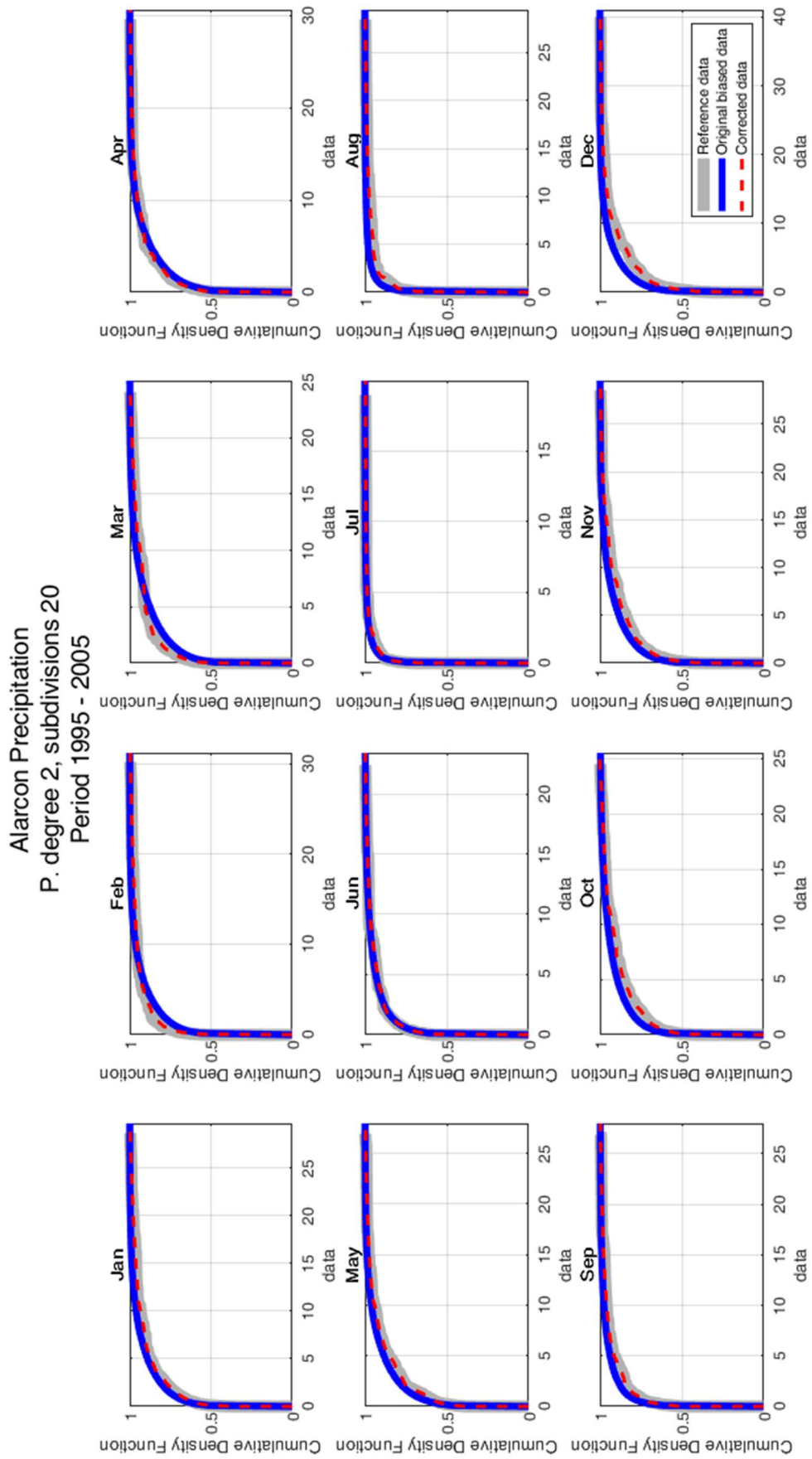
Figure 5.7: CDFs of the observations (grey) compared to the biased (blues) and corrected (dotted red) forecasts. Alarcón reservoir, first precipitation subset. Source: self-made.
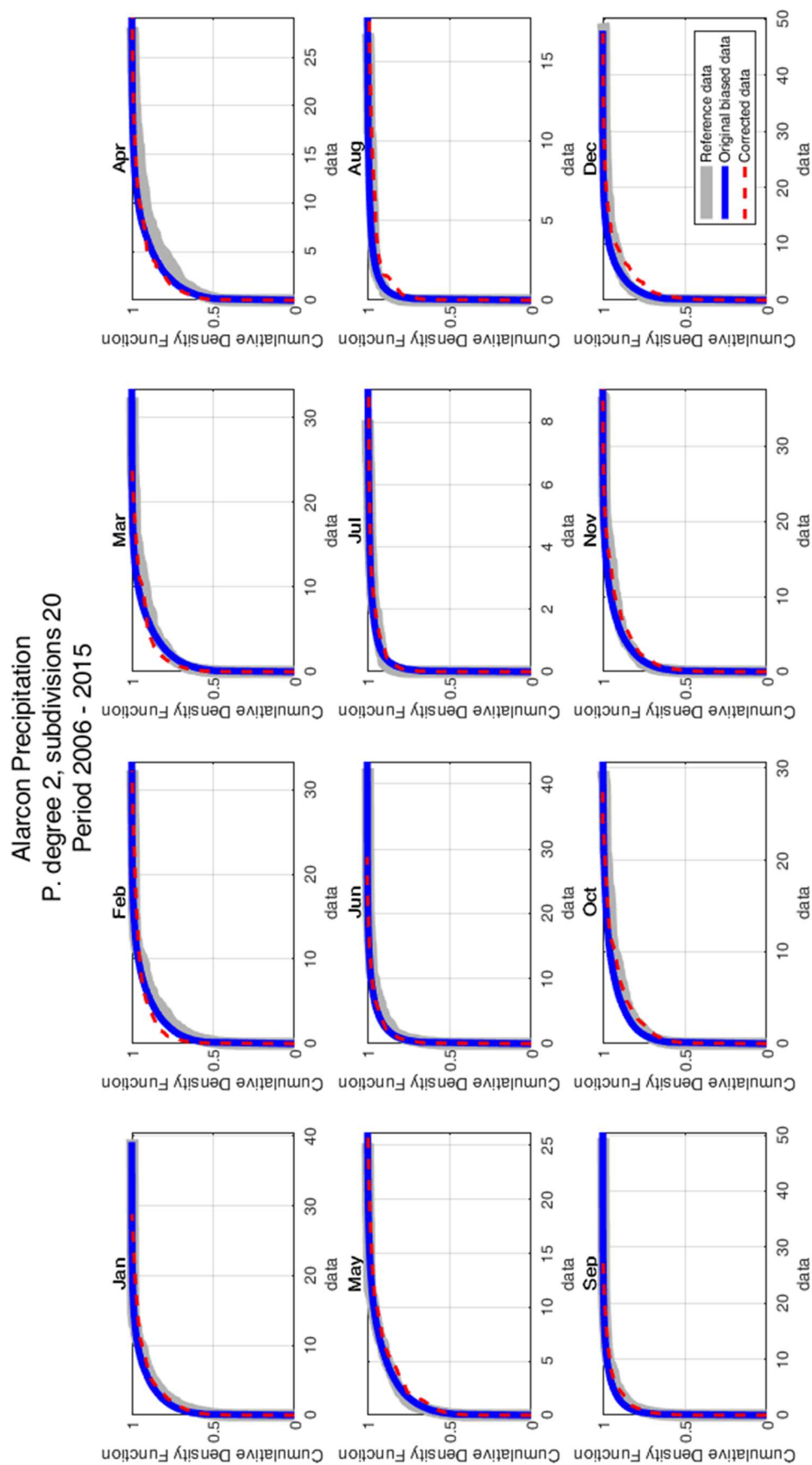
Figure 5.8: CDFs of the observations (grey) compared to the biased (blues) and corrected (dotted red) forecasts. Bellús reservoir, second precipitation subset. Source: self-made.

The bias correction algorithm found is then applied to the data. The resulting corrected forecasts are plotted against the original forecasts and observation in Figure 5.9 and Figure 5.10. From the graphs, it is clear that the adjusted temperature forecasts follow the observations better than the original biased data, while no definitive judgement can be made looking at the precipitation graph: there are high observed peaks which are covered by neither the original nor the corrected forecasts, nevertheless, the corrected forecasts increase more than the original biased data when these peaks occur. However, even if these are good premises, nor the graphical observation of the plots, nor the simple improvement of the CDFs are definitive proofs to tell if the algorithm is actually working: to really evaluate the performance of the bias correction algorithm, the mean average error indicator was employed. It has been calculated according to the following patterns:

- Yearly sum: sum of the monthly MAE of each month of the year. An example is reported in Figure 5.11;
- Month average: average of the MAEs of every month. Figure 5.12 shows an example.

From the examination of the bar plots, several considerations can be made.

As hinted in the analysis of the CDFs, an overall good performance of the algorithm for the temperature correction is achieved, while the precipitation correction does not deliver the same results. In fact, as an unconditional method, QM does not preserve the connection between each pair of forecast and observation values. Thus, QM might sometimes adjust the raw forecasts to the wrong direction for some forecast values and cannot provide satisfying results as conditional methods. Moreover, Zhao et al. found that although QM is able to correct the bias, it cannot ensure the reliability and coherence of forecasts ('coherence' here means forecasts are at least as skilful as climatology). The reason is that QM does not consider the correlation between raw forecasts and observations.

More in detail, the performance seems to vary significantly, by year and by month: for example, the precipitation correction usually yields a performance similar to that of the original forecasts. Looking at the temperature correction performance, in Figure 5.11 the picture is clear: the algorithm improves the predictive ability of the forecasts: nevertheless, in the year 2010 the performance of the corrected forecasts is worse than that of the original forecasts. Also, looking at Figure 5.12 it can be seen that the performance changes throughout the year, with lower values during the summer months.
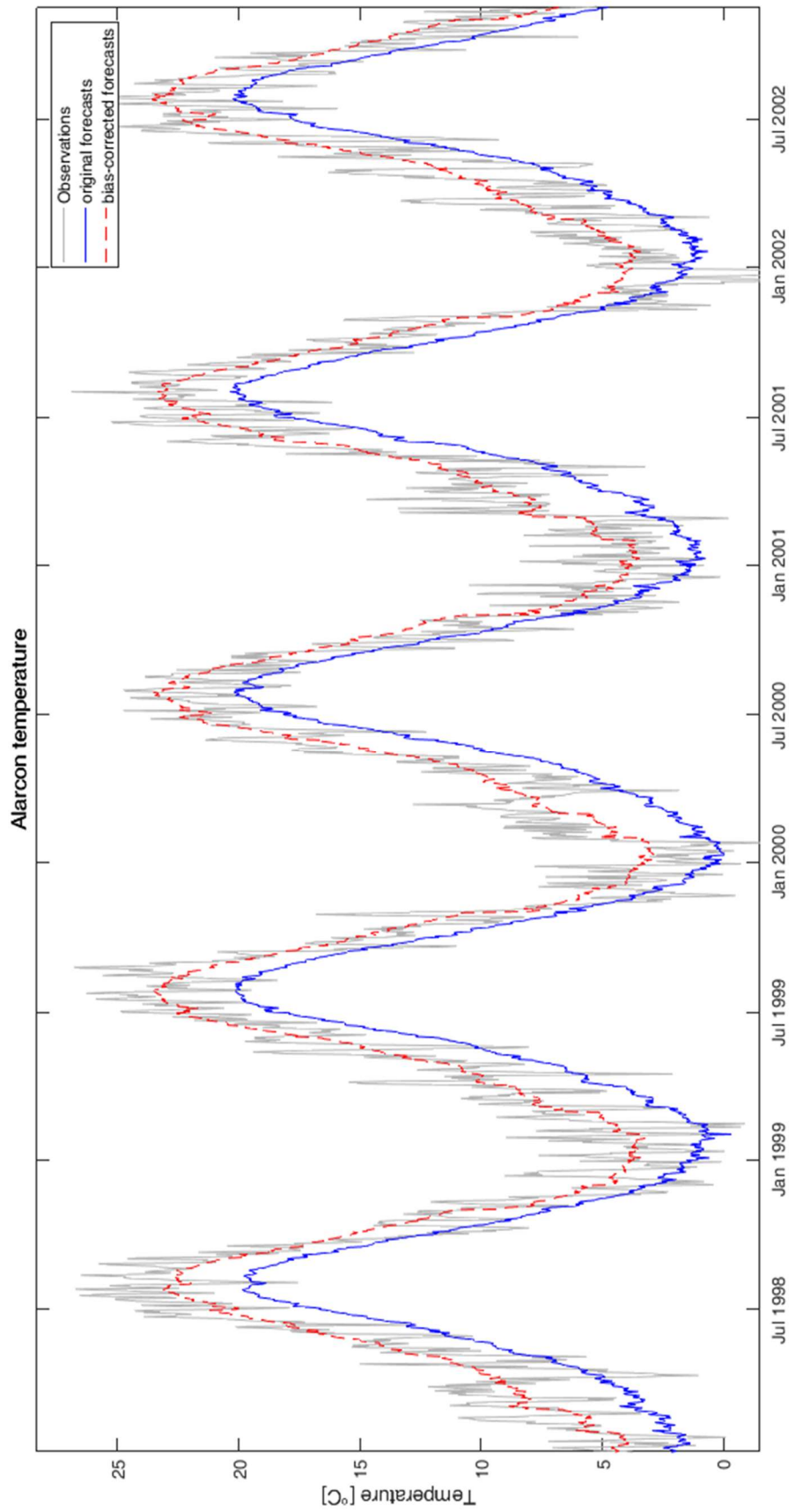
Figure 5.9: extract of the plot of the temperature biased and corrected hindcasts against observed values for the Alarcón reservoir. Source: self-made.

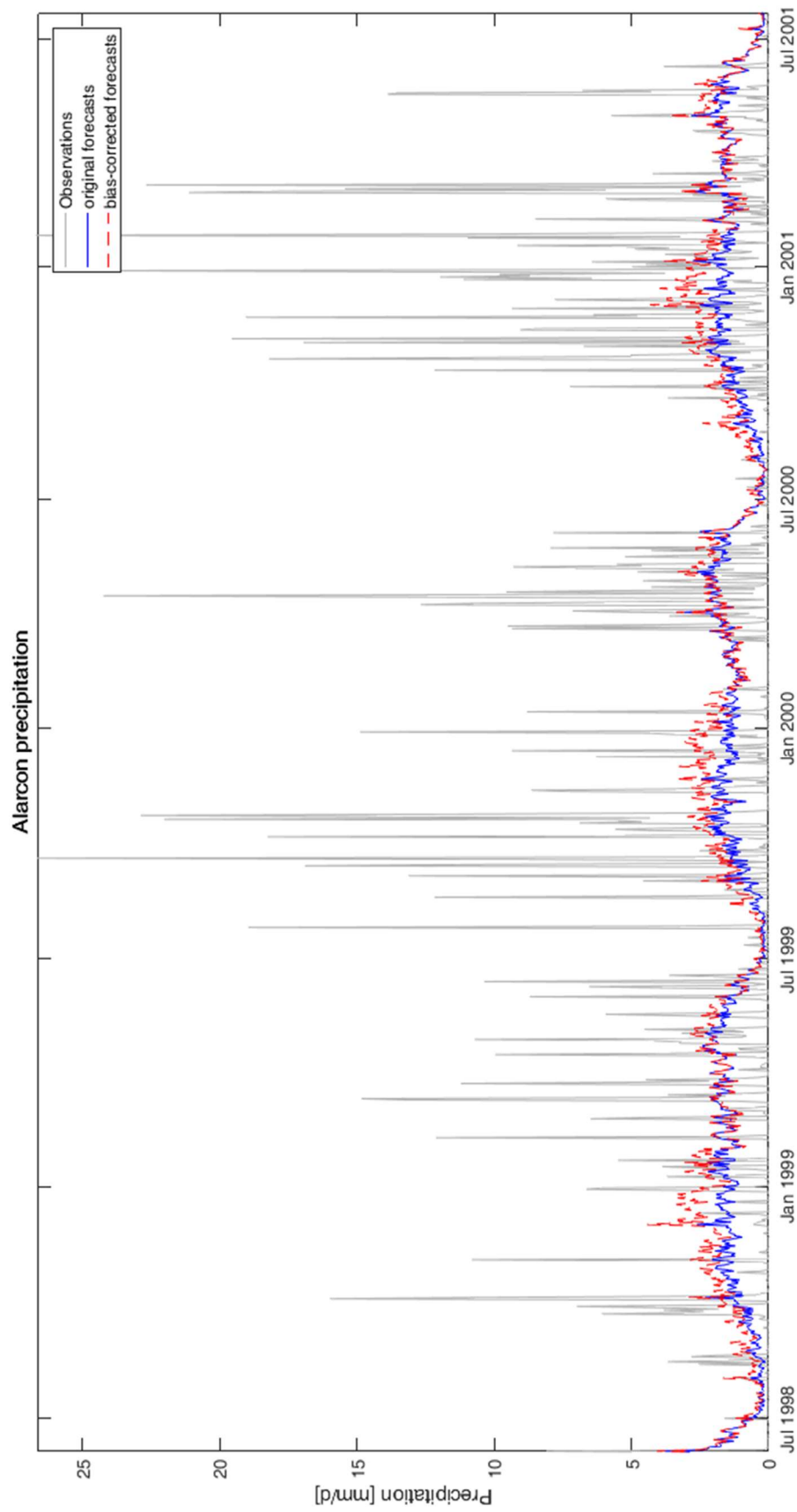Figure 5.10: extract of the plot of the precipitation biased and corrected hindcasts against observed values for the Alarcón reservoir. Source: self-made.
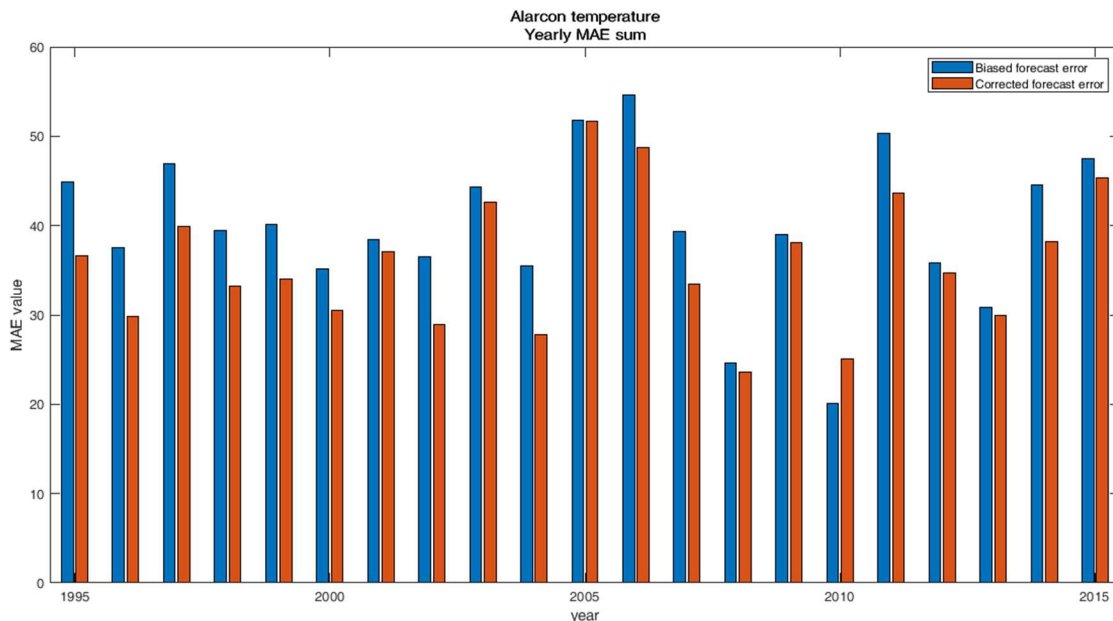
Figure 5.11: Yearly sum of the MAE indicator of the biased and bias-corrected temperature time series. Alarcón sub-basin. Source: self-made.
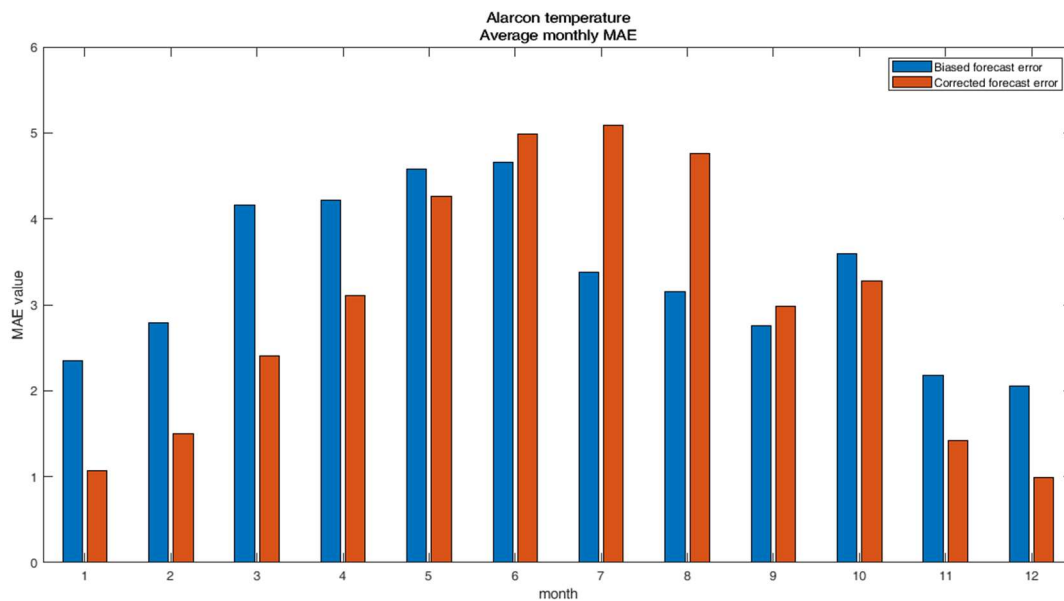

Figure 5.12: average MAE of every month of the biased and bias-corrected temperature time series. Alarcón sub-basin. Source: self-made.

5.4 Hydrological forecasts generation

Hydrological forecasts for the Júcar sub-basins of Alarcón, Contreras and Bellús have been generated for the period 2005-2008, which corresponds to a severe drought episode, to test the skill of the hydrological models and meteorological forecasts described before to provide comprehensive hydrological forecasts. The model will be run once for every ensemble member of the corrected forecast. Each plot corresponds to 25 model runs. The title of each plot indicates the month in which the forecasts were issued (the forecasts are issued on the first day of the month), and shows the simulation of the next 215 days. Five time series are plotted in the same graph: one corresponds to the observed runoff and the other four correspond to:

- The 25th percentile of the runoff simulation ensemble fed with the corrected forecasts;
- The 75th percentile of the runoff simulation ensemble;
- The median of the runoff simulation ensemble;

Such an organization has been decided in order to retain and show the statistical information given by the 25 ensemble members of the forecasts, while keeping the plot readable without filling it with 27 different time series.

5.4.1 Alarcón sub-basin

Figure 5.13, Figure 5.14, Figure 5.15 and Figure 5.16 show the simulated vs observed plots for the Alarcón sub-basin, for the period 2005-2008. A very good agreement between the trends of observation and corrected simulations can be observed in the whole dataset, excepting some isolated cases. The general trend seems to highlight that the first quartile of the simulations run with the corrected forecasts matches the observations better than all the other simulated time series. On occasions (forecasts issued on May 2006, May and June 2007 and July and November 2008), the initial conditions are not properly set by the HBV model, resulting in an initial error after which the simulated time series tends to get closer to the observed one. HBV light does not allow to introduce the initial conditions, so a warm-up period of 5 months, fed with observations, was used. In some cases, this period was not enough to secure that initial conditions were adequately reproduced and, as a result, the runoff for the beginning of the forecasting period was wrong (for example, Alarcon in May 2006). Moreover, wrong initial conditions could result in a completely prejudiced forecast.

The general trend of a low runoff, for the whole period, is correctly simulated by the system. The driest periods, corresponding to the summer months of the years 2005 and 2006 and the second part of 2007, have been simulated correctly, with months of anticipation. More in details:

- The dry periods of the summer-fall of 2005, 2006 and 2007 were predicted already in the first month of the same year, and consistently simulated in the following months: there were clear clues for an incoming particularly dry spell months in advance;
- The recuperation moment in the end of 2006-spring 2007 was forecasted as well: the simulations showed a steady increase of the runoff, several months before, and they were consistently, month after month, predicting an increase of the water flow;
- The forecasts issued in the months of July, August and September 2007 wrongly indicated a less dry period in the winter 2007-spring 2008. However, starting from October 2007, the forecasts started to correctly indicate the following dry period;
- The dry summer of 2008 was forecasted already starting from January of the same year, and the more wet fall and winter were predicted as well.

From these observations, it is evident that the dry periods are consistently reproduced several months before their occurrence. Moreover, there have been no "false positives": if a period was forecasted to be dry, then it actually was dry. The system made some mistakes on the prediction of more wet periods. Some of them were predicted to occur before they actually happened, others were not forecasted. The very wet spring of 2009 was not predicted by the system.

Figure 5.13: simulation results for the year of issued forecasts 2005. Alarcón sub-basin. Source: self-made.

Figure 5.14: simulation results for the year of issued forecasts 2006. Alarcón sub-basin. Source: self-made.

Figure 5.15: simulation results for the year of issued forecasts 2007. Alarcón sub-basin. Source: self-made.

Figure 5.16: simulation results for the year of issued forecasts 2008. Alarcón sub-basin. Source: self-made.

5.4.2 Contreras sub-basin

The simulation results are shown in Figure 5.17, Figure 5.18, Figure 5.19 and Figure 5.20
For the Contreras sub-basin, the same analysis has been carried out. In this case, the system
performs even better: the simulations show a tighter correspondence to the observations, even
if they do not reproduce the noise of the observed runoff signal. Particularly in the first tested
year (Figure 5.17) the model is working exceptionally well: the maximum difference between
the trends of the observations and simulations is 0.05 to 0.1 mm/d. In the other years a very
good correspondence to the trend is observed as well, with the same characteristics highlighted
before for the Alarcón sub-basin: the system reproduces dry periods very well, but it lacks
predictive capacity for more wet periods (Figure 5.19 and Figure 5.20), and misses the peaks,
even if it shows an increase in the simulated runoff in correspondence of the peaks. Anyway,
this effect may be because some of the peaks correspond to convective phenomena that cannot
be adequately captured by the SEAS5 model, as they happen in specific locations, while the
model has worldwide coverage. This phenomenon cannot be predicted with more than few
days in advance (and in some case even hours in advance), so it was not expected to be
reproduced by 7-month lead forecasts. This case, in the same way of the previous, confirms
that the 25th percentile is the one that gets the closest to the observations trend.
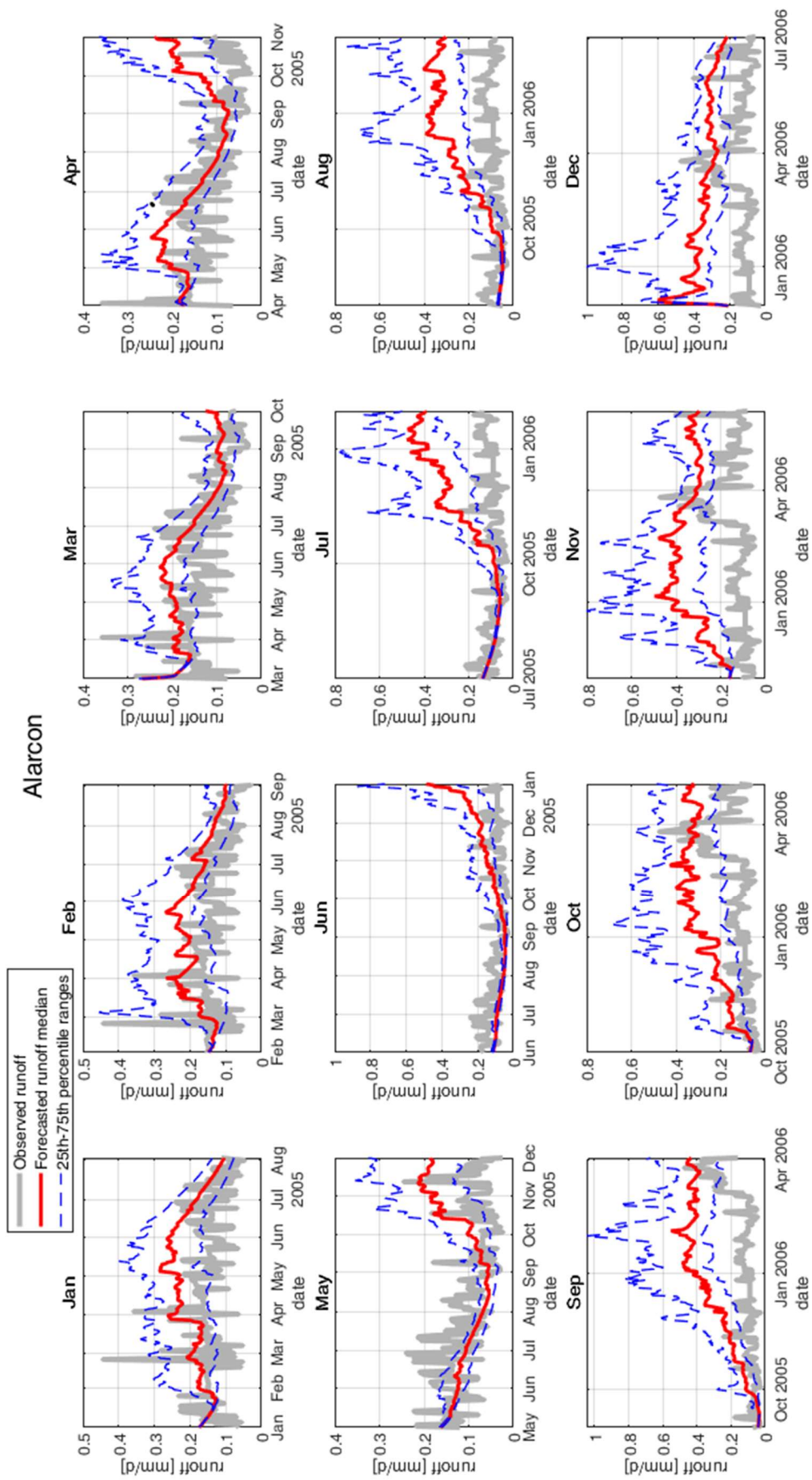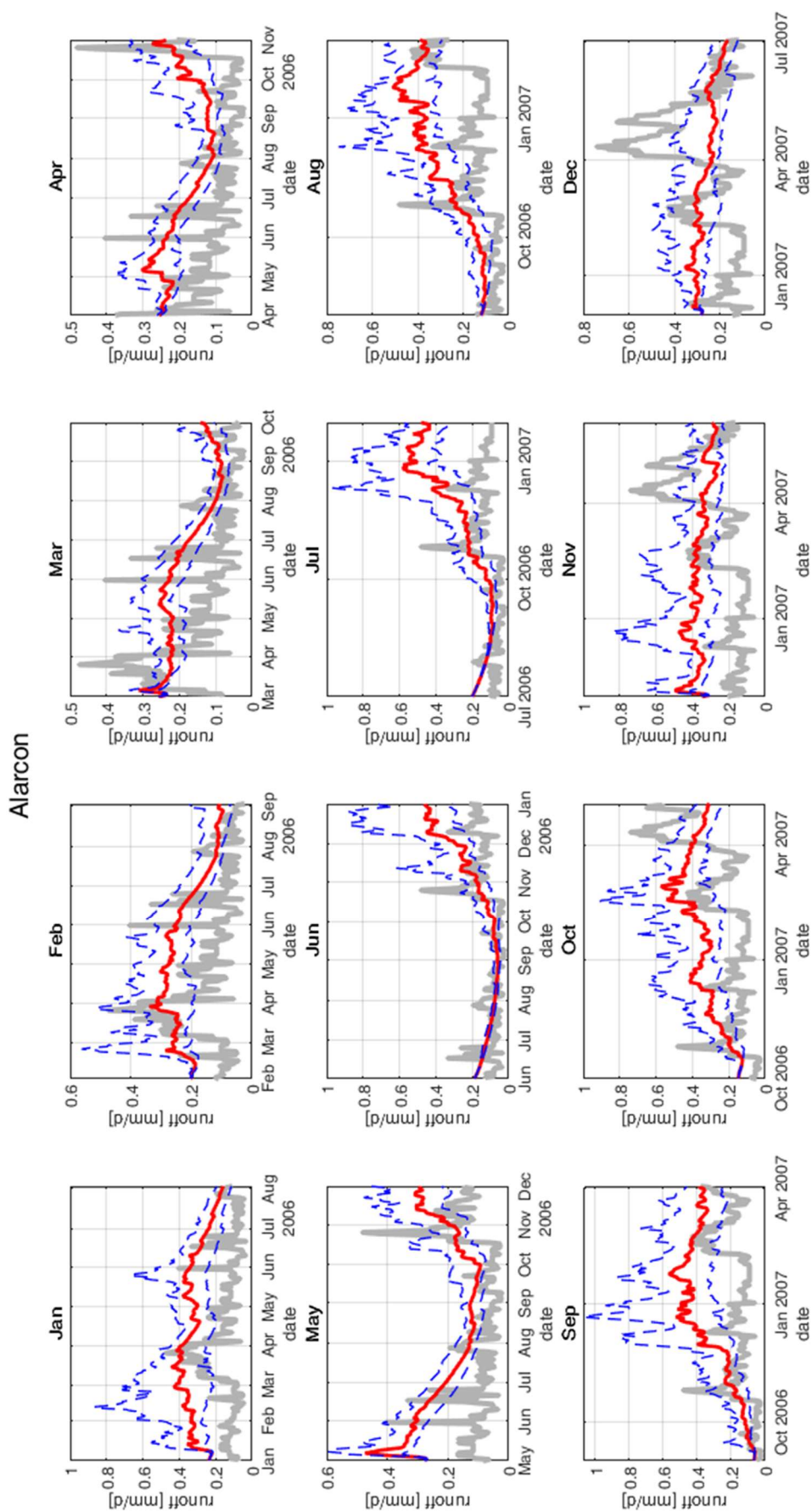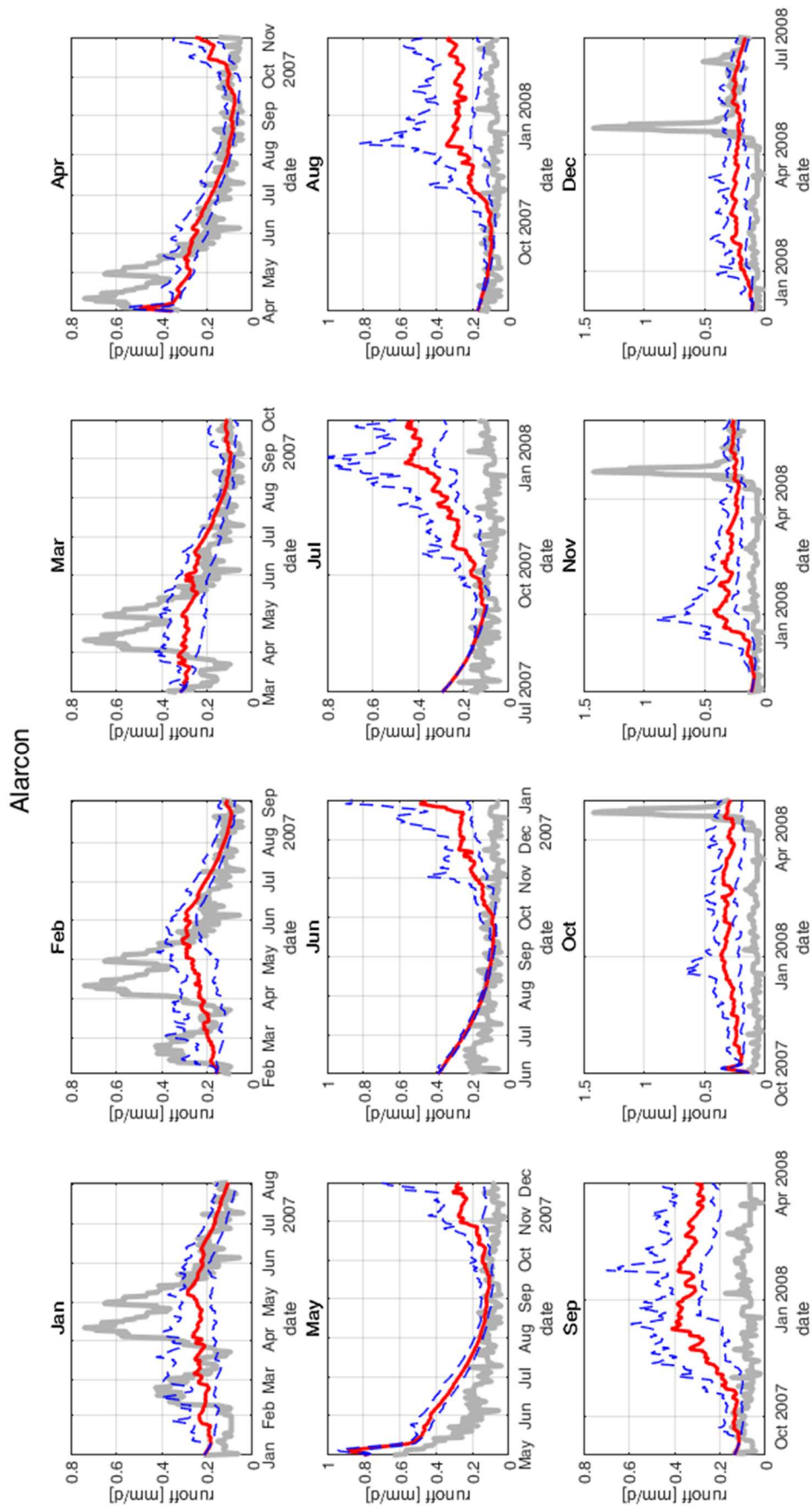
Figure 5.17: simulation results for the year of issued forecasts 2005. Contreras sub-basin. Source: self-made.

Figure 5.18: simulation results for the year of issued forecasts 2006. Contreras sub-basin. Source: self-made.

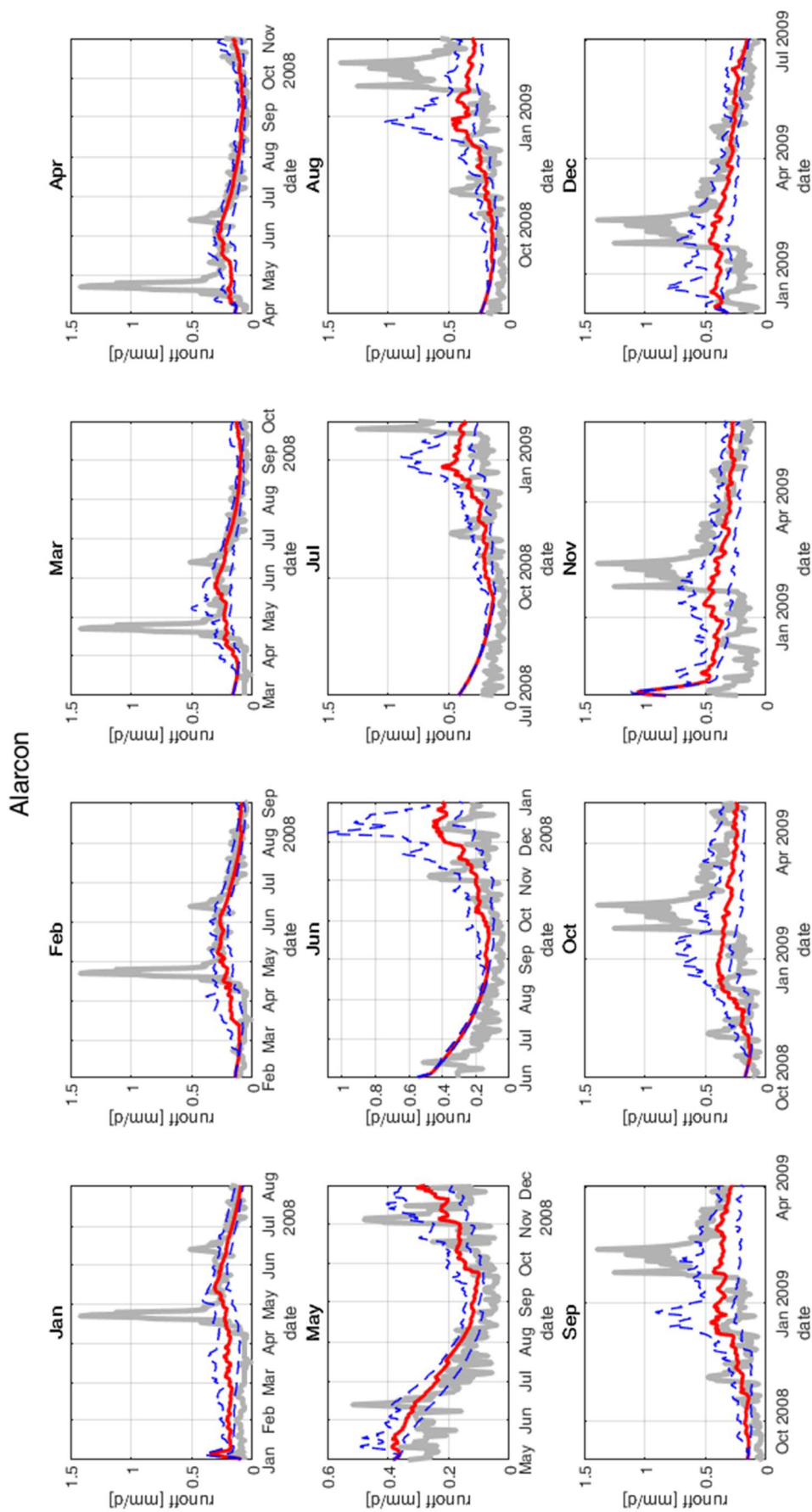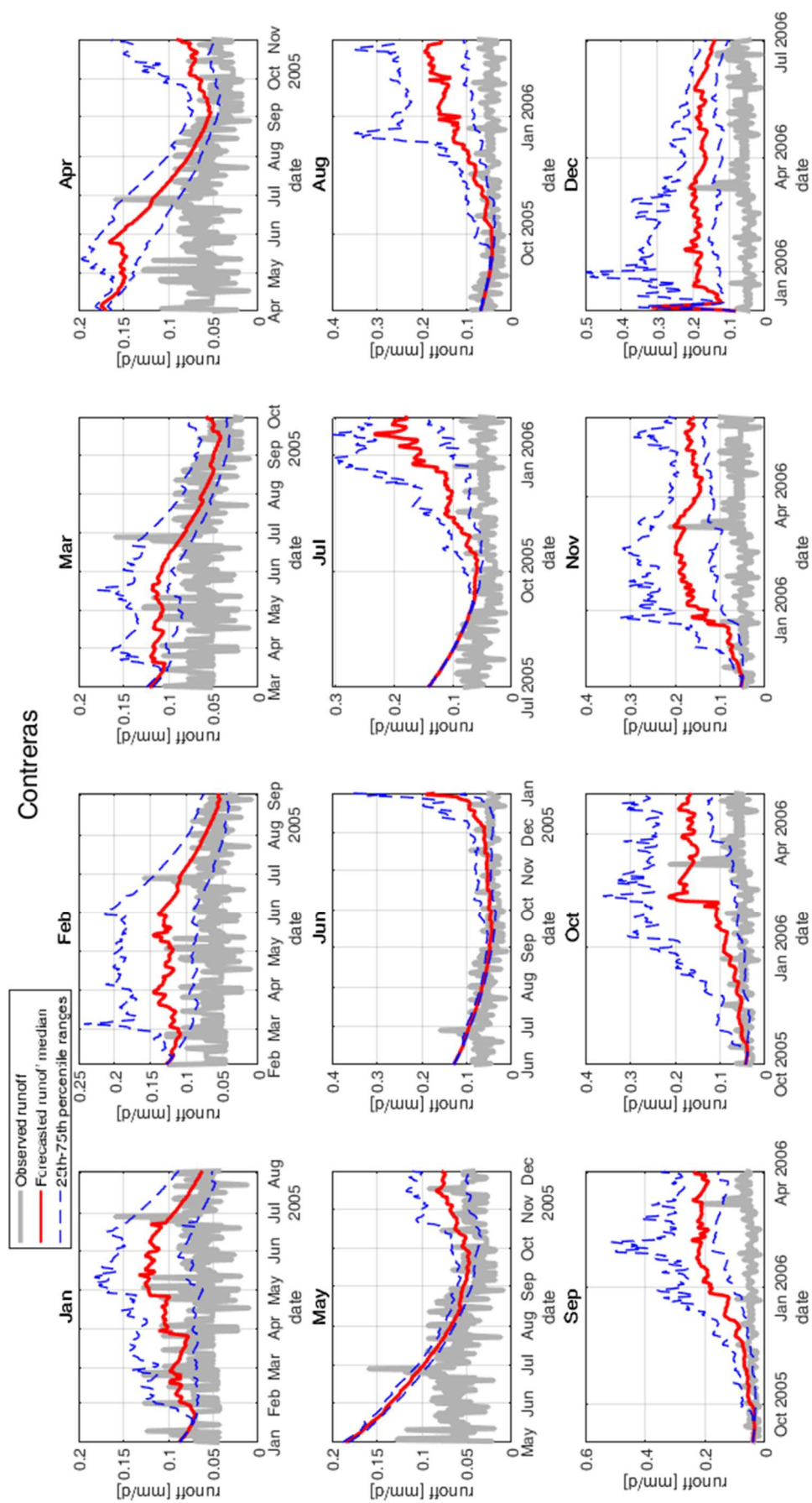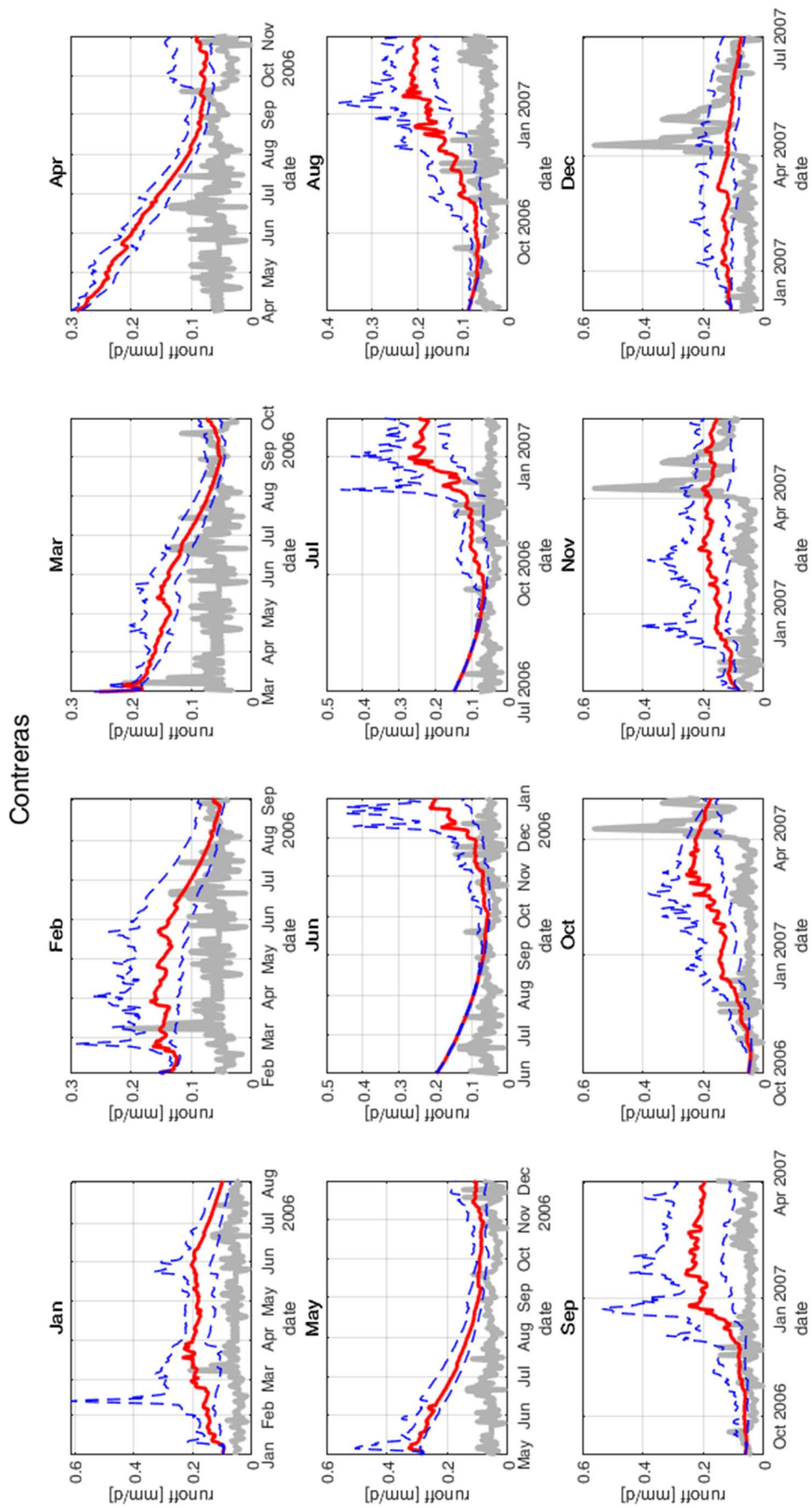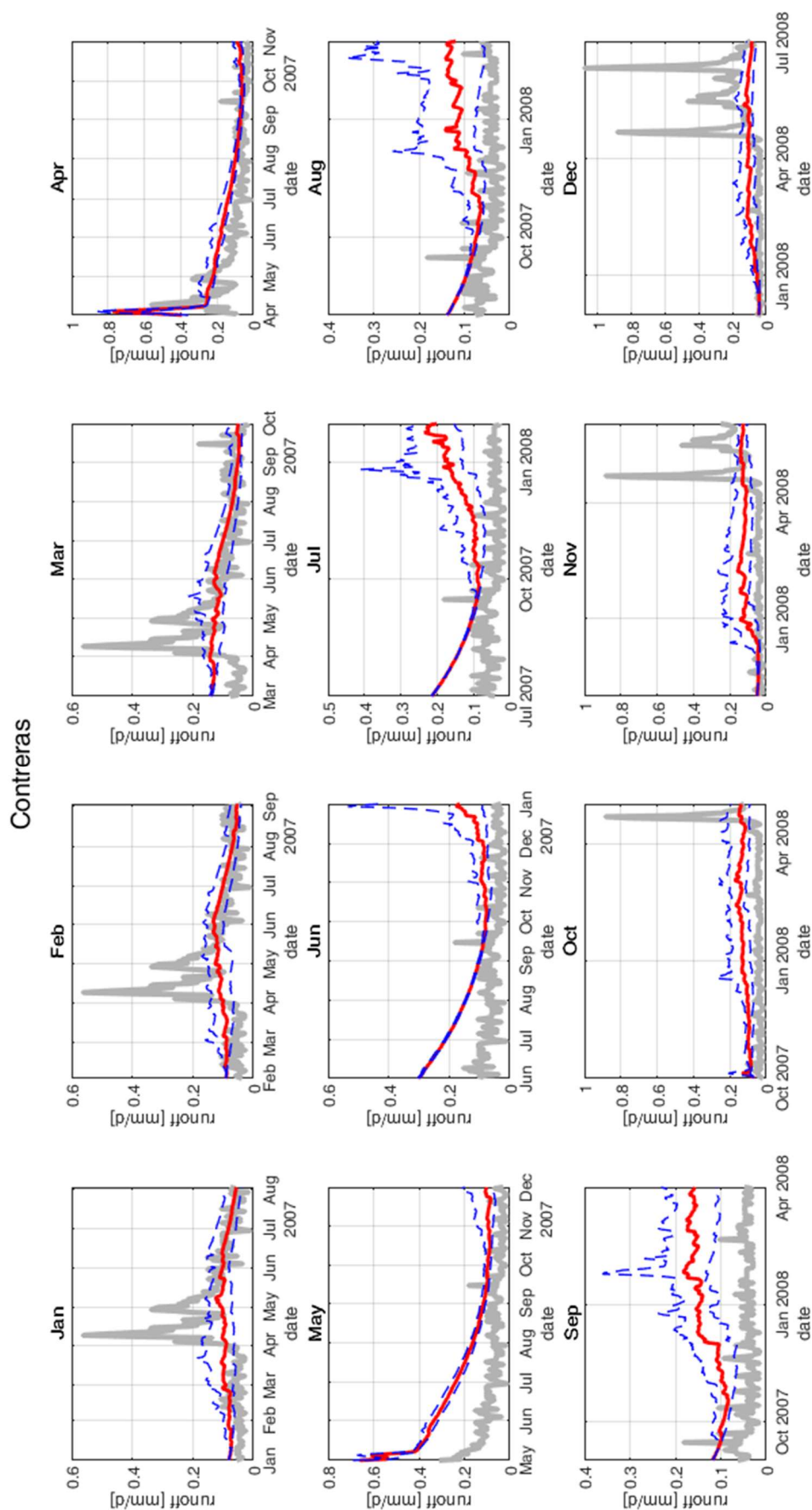Figure 5.19: simulation results for the year of issued forecasts 2007. Contreras sub-basin. Source: self-made.

Figure 5.20: simulation results for the year of issued forecasts 2008. Contreras sub-basin. Source: self-made.

5.4.3 Bellús sub-basin

The simulations plots for the years from 2005 to 2008 are reported in Figure 5.21, Figure 5.22, Figure 5.23 and Figure 5.24. The plots are cut to the y value of 0.8, to better highlight the low values: high peaks of observed runoff reach up to 30 mm/d, making the lowest (and most occurring) values less readable if kept in the scale. The smaller scale of this basin makes hydrological forecasts more difficult. In this case, the peaks in the observations are barely "noticed" by the simulations, and wetter stages between the dry periods are not very well simulated. This happens for the same reasons explained in the previous paragraph 5.4.2 for the Contreras sub-basin. Nevertheless, there is a noticeable increase in the 75th percentile of the ensembles, in these cases, indicating a higher uncertainty of the weather forecasts, as opposed to the very little variations present in more dry moments. Sufficient results are obtained in dry periods, with the ensemble median of the simulations being the closest to the observations. There is a consistent error in the setting of the initial conditions: it can be observed in the fact that simulations, at the beginning of the plot, are not very describing of the observations, and tend to get closer with the advancing of the simulation time. The lower grade of coincidence of the simulations to the observations is due to the lower performance attained by the previous HBV calibration and by the small surface of the sub-basin, which produces high variability in the signal of the observed runoff. The lower skill may also be caused by the fact that the Bellús sub-basin presents a very small area, compared to that of Alarcón and Contreras, and thus its inner variability cannot be adequately capture by the meteorological forecasting model SEAS5, whose global coverage makes that very few pixels cover this sub-basin.
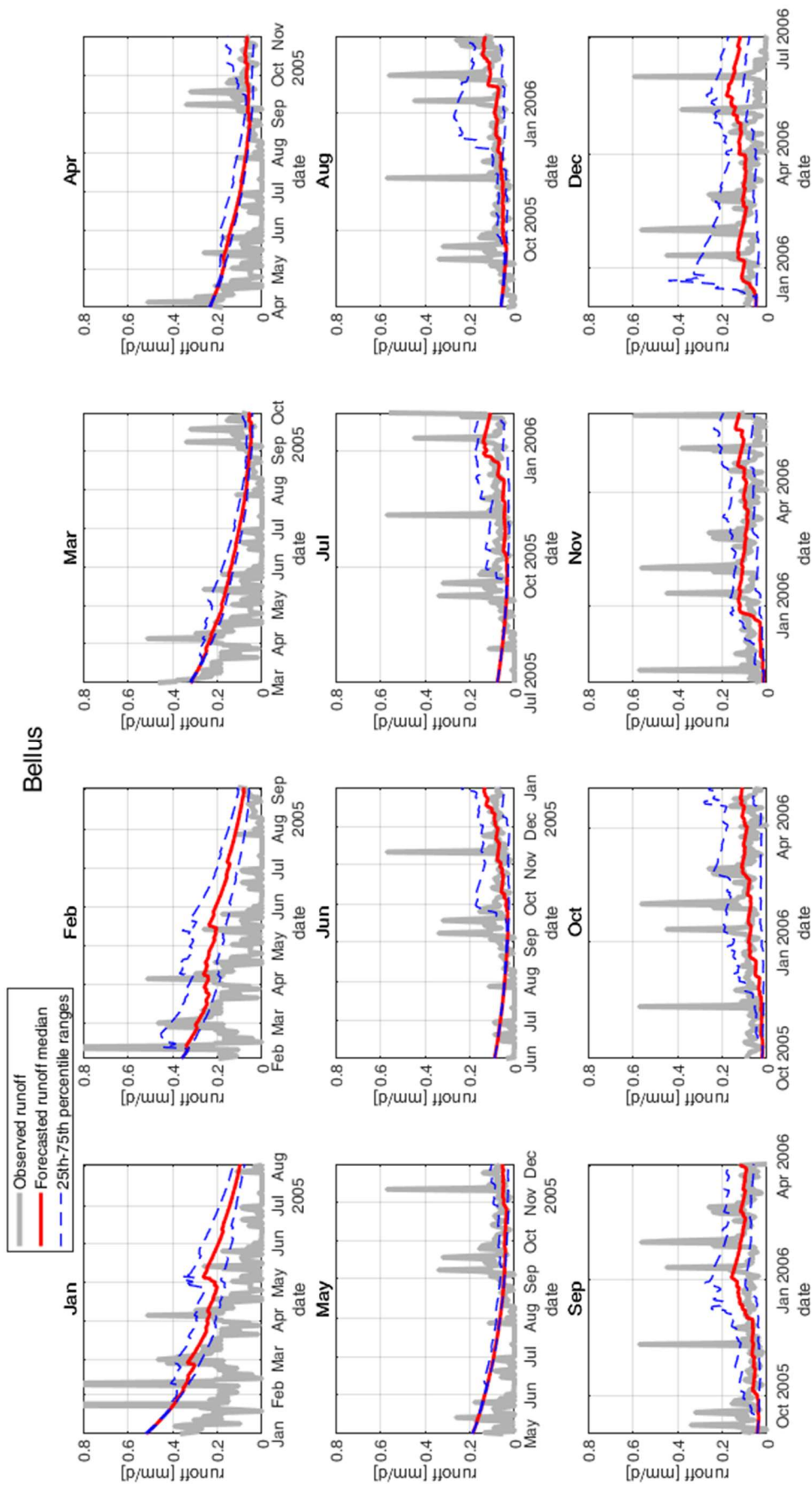
Figure 5.21: simulation results for the year of issued forecasts 2005. Bellús sub-basin. Source: self-made.
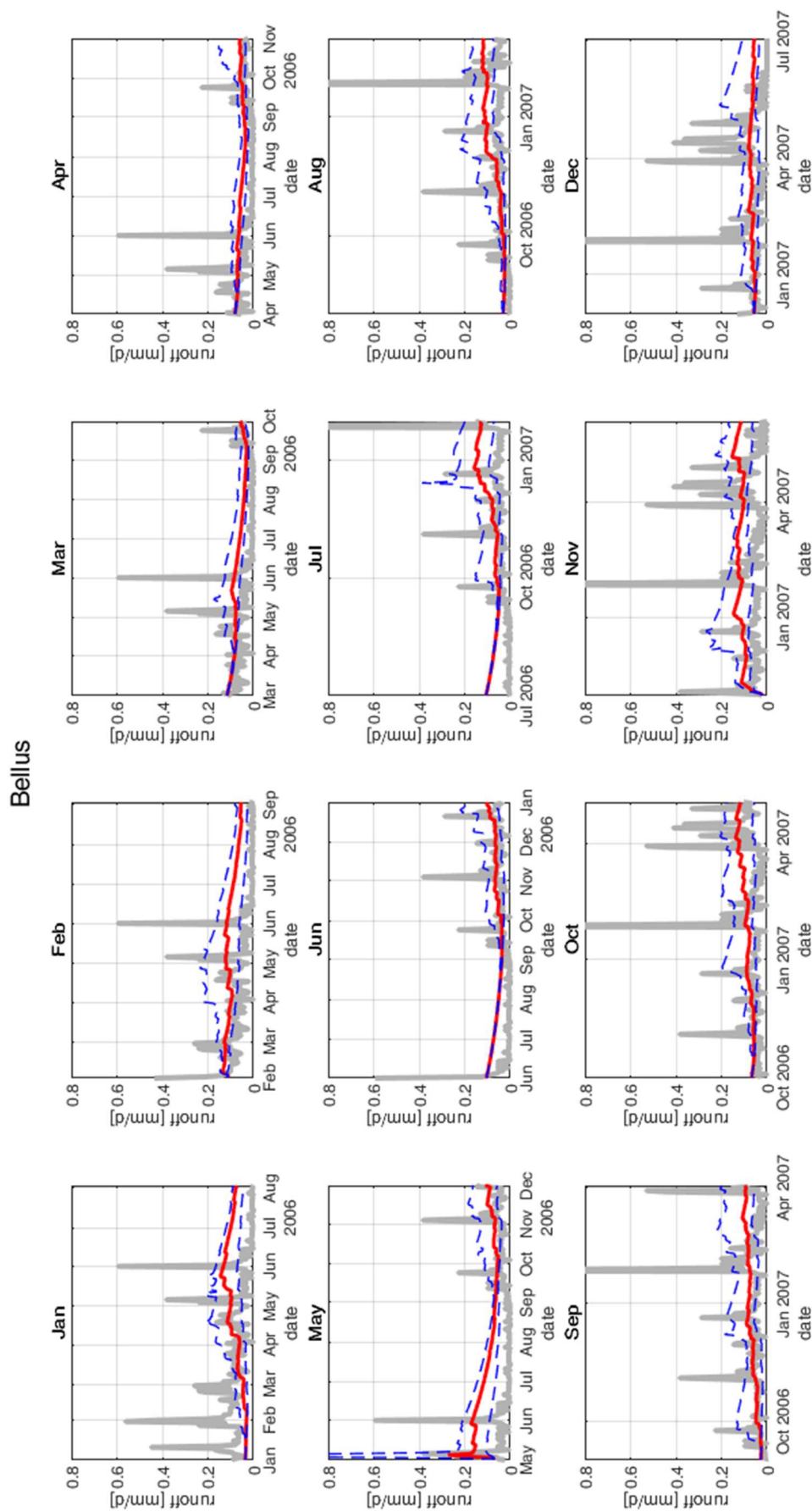
Figure 5.22 simulation results for the year of issued forecasts 2006. Bellús sub-basin. Source: self-made.

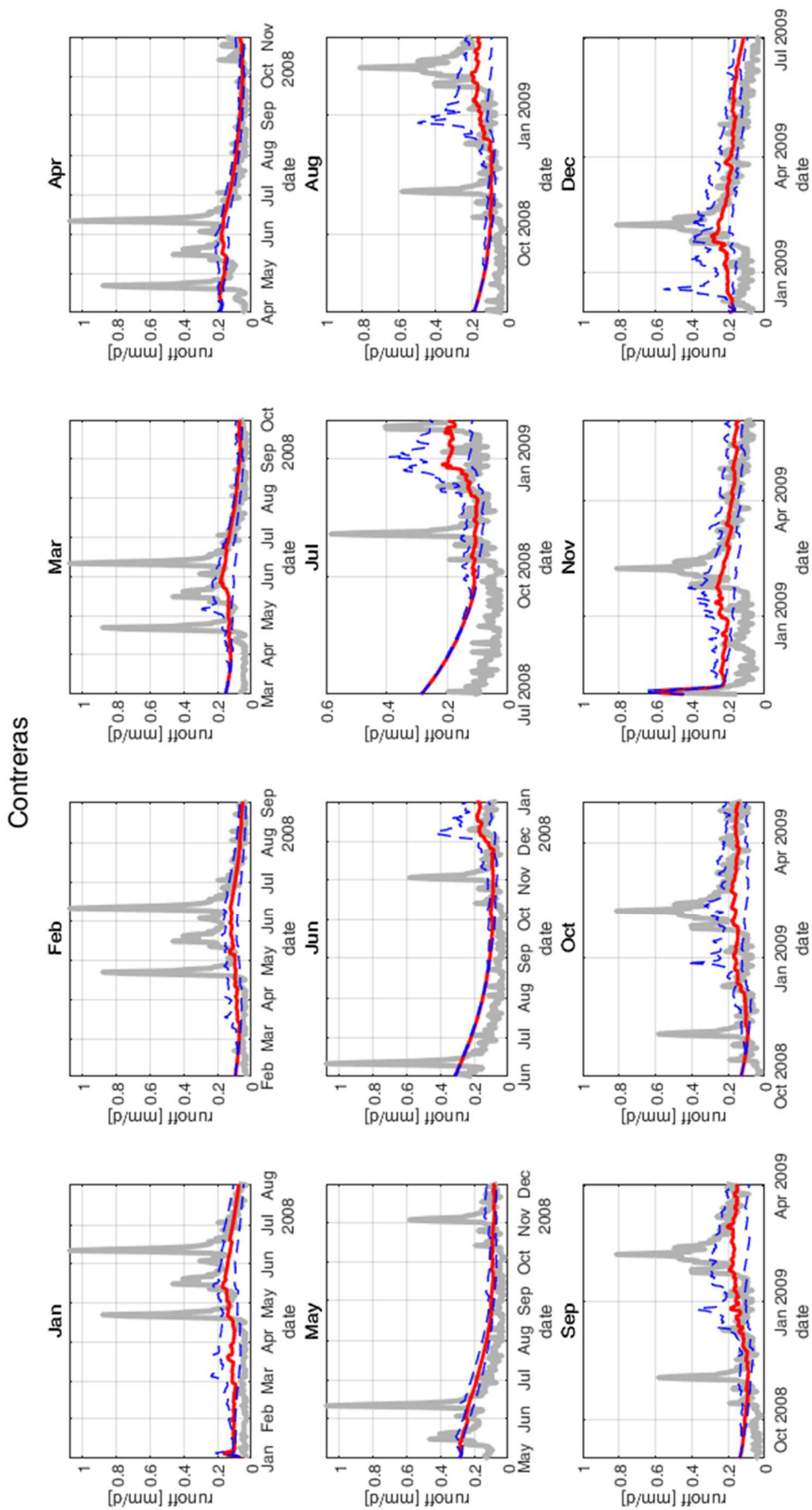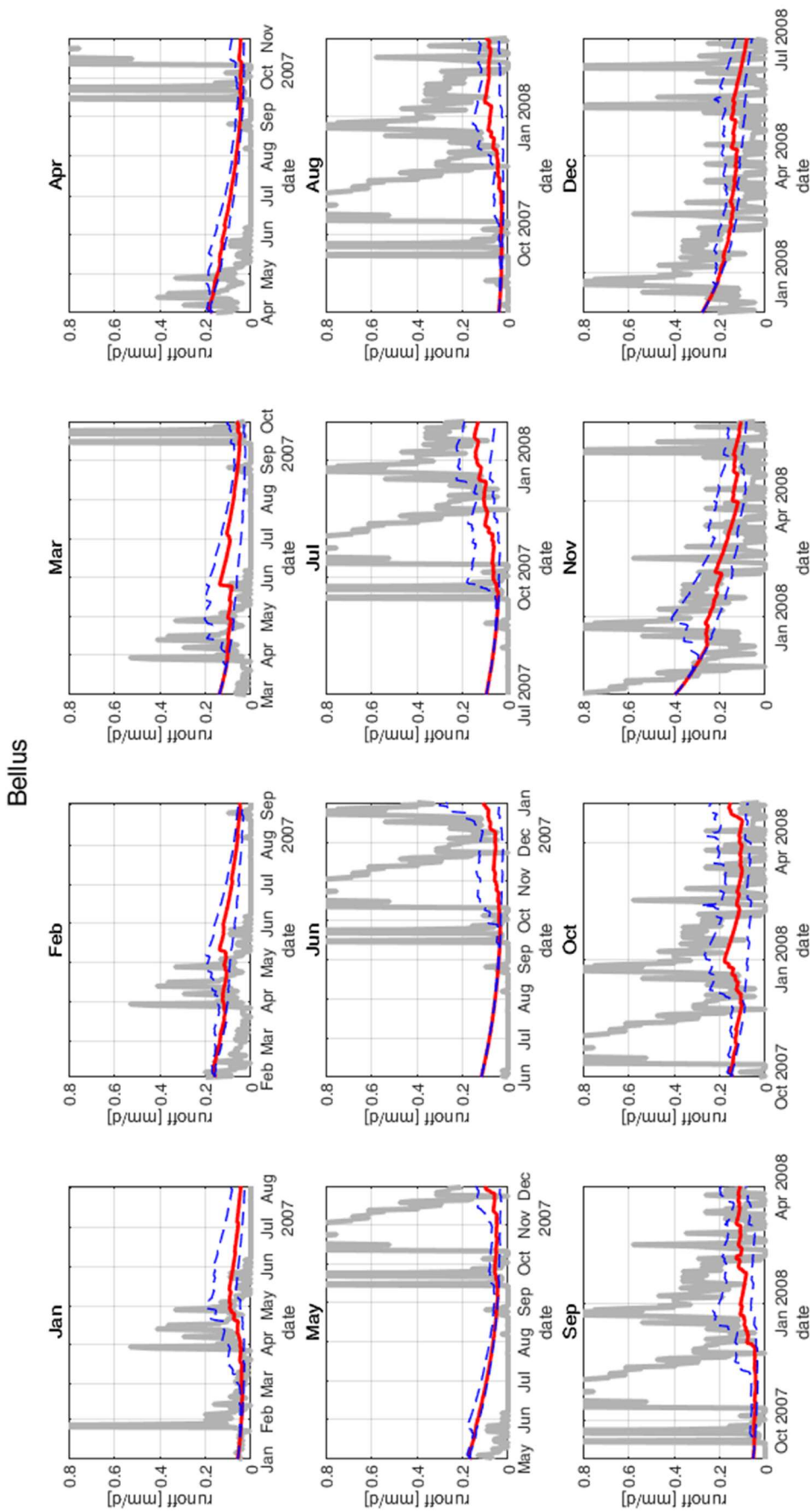Figure 5.23 simulation results for the year of issued forecasts 2007. Bellús sub-basin. Source: self-made.
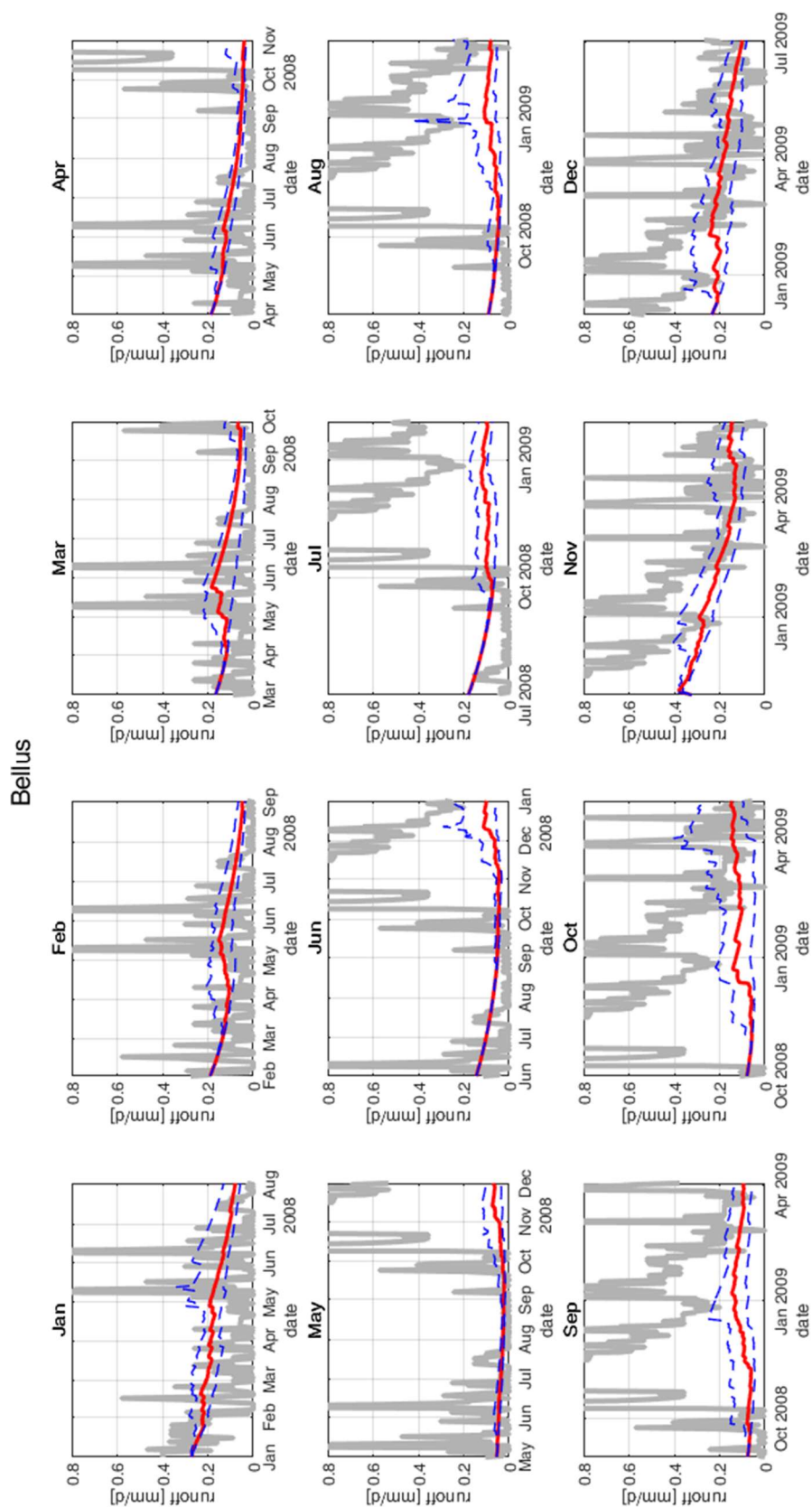
Figure 5.24 simulation results for the year of issued forecasts 2008. Bellús sub-basin. Source: self-made.

CHAPTER 6

CONCLUSIONS

This is the last chapter of the work, and it is devoted to a brief summary of the work, critical conclusions and possible future lines of research.

6.1 Brief summary of the analysis

This work focused on the creation of a hydrological forecasting system, specifically calibrated on 3 sub-basins of the Júcar river basin. To do so, several different tools have been exploited. The hydrological model providing the simulations is the HBV model, already widely and successfully tested in a variety of different studies. To calibrate the model to the local conditions, observation data were used, obtained from the Spain02 v5 dataset. The weather hindcasts (forecasts generated on past periods), downloaded through the Copernicus Climate Data Store Service, were generated by the ECMWF SEAS5 system, which relies on an analysis of the initial state of the ocean, a global coupled ocean-atmosphere general circulation model, and a post-processing suite. Said forecasts were then processed with a quantile mapping bias-correction method, to account for and correct systematic errors present in the original ensemble.

The first step, after obtaining the necessary data, was the preparation of the input to the HBV model, operation carried out with the EXCEL software. The potential evapotranspiration was calculated with the Thornthwaite formula, deemed the best option in this case: the Penman formula required data which were not readily available.

Then, the HBV model was calibrated, for every studied sub-basin, with the help of the GAP (Genetic Algorithm-Powell) parameter search tool included in the HBV-light software. Sufficient to good results were obtained in the model calibration and no systematic errors were observed in the simulated time series. The Bellús sub-basin showed the least satisfying, even

if still sufficient, results: this is due to the small areal dimension of the sub-basin, and the fact that it is subject to swift and unpredictable changes of the runoff, which lead to very high peaks, difficult to correctly simulate by any model. Anyway, the base flow was very well simulated by the model.

The third action to be carried out was the bias-correction of the forecasts. Given the large amount of data to process, it was preferred to use the MATLAB software, given its capacity of dealing conveniently with datasets of important dimensions and automating repetitive calculations. The application of the correction method provided good results with the temperature variable, while a less convincing performance was obtained for the precipitation. This is due to the fact that the quantile mapping technique is an unconditional method: it does not preserve the connection between each pair of forecast and observation values. Since precipitation forecasts suffer from not only bias but also reliability and coherence problems, an unconditional method did not provide satisfactory results.

Finally, the bias-corrected forecasts were input to the previously calibrated HBV model, to test the predictive ability of the system. The dry period 2005-2008 was studied for each of the sub-basins. The resulting simulations indicated a good agreement to the trend of the observations, especially during dry periods. Stages with a higher runoff were not simulated with the same accuracy, even if an increasing trend in the simulations matched the observations in these cases. The forecasting potential of the system seem to be very interesting: dry events were predicted, consistently, several months before their occurrence.

6.2 Conclusions

The implemented system is able to return reliable forecasts of the trend of the future state of the upper Júcar river basin. It showed a particularly good performance in consistently predicting, months in advance, incoming dry periods. However, it does not reproduce the relatively high variability of the runoff, and showed lower performance in predicting wetter periods. This was expected: wet periods have a very unpredictable pattern in the Mediterranean climate, and are usually brief moments of heavy precipitations.

Moreover, the presented system suffers from the typical and inherent issues which trouble every mathematical model: mistakes in the measures, mistakes in model structure, mistakes in model calibration, initial conditions, boundary conditions and simplifying assumptions contribute to a not perfect correspondence of the simulations to the reality.

In particular for this case, the weak points encountered have been:

- The computation of the actual inflow to the reservoir, which is very difficult to accurately calculate, given the vast surface of the basin and the variety of mechanisms

involved. In addition, the considered evaporation and infiltration phenomena were described by empirical relationships, which are good enough, but not very accurate;

- The precipitation and temperature measures are interpolated from different measuring points and sub-basin averages have been considered, rather than higher resolution grids;

- The 7 month-lead forecasts, despite the recent considerable improvements, still are not as reliable as short-term forecasts, for which alternative (but yet not open-access) products exist;

- The bias correction method exploited, while showing a good performance for the temperature forecasts, does not offer significant skill improvements to the precipitation forecasts;

- The calibration procedure relies on the idea of stationarity, which is not strictly respected: land use change and climate change can affect the definition of the model coefficients.

Future research can try to remediate these problems. Still, despite all these issues, the study showed a rather good overall performance of the implemented system in the studied area.

6.3 Further investigation lines

The present research was the first to implement a bias-correction algorithm to daily-scale seasonal forecasts to be input to a hydrological model in the Júcar river basin: no such a scheme has been seen before. The results are promising, but more research needs to be done. Different bias-correction methods can be tested, in order to select the one which best fits each variable, especially for the precipitation forecasts, which were not satisfactorily corrected with the quantile mapping bias correction adopted, due to their problems of reliability and coherence. Different techniques have been tried in other studies, which showed better performance.

The system needs to be furtherly tested, on different sub-basins of the same river basin and on different river basins, in Spain and in other countries, to furtherly confirm its efficacy. The various possible adjustable parameters and methods have not been completely explored in this research: for example, a different, similarly performing parameter set, could have been chosen for the HBV model calibration, or a non-linear fitting function could be used to fit the forecasts quantiles to the observations' ones.

Different sources of forecasts could be used as input to the system: just one of the many forecasts products of the Copernicus programme has been used.

Different hydrological models could be tested, to simulate the runoff of the sub-basins.

Finally, the system could be coupled with a water resources management model and tested to understand the improvements it would bring to the administration of the water resources of the river basin.

CHAPTER 7
REFERENCES

Addor, N., Clark, M. P., & Nijssen, B. (2016). Adapting hydrological model structure to catchment characteristics: A large-sample experiment. In *EGU General Assembly Conference Abstracts* (Vol. 18, p. 10250).

Arsenault, R., Poulin, A., Côté, P., & Brissette, F. (2014). Comparison of Stochastic Optimization Algorithms in Hydrological Model Calibration. *Journal of Hydrologic Engineering*. https://doi.org/10.1155/2013/427908

Bartholmes, J. C., Thielen, J., Ramos, M. H., & Gentilini, S. (2009). The european flood alert system EFAS" Part 2: Statistical skill assessment of probabilistic and deterministic operational forecasts. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-13-141-2009

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*. https://doi.org/10.1162/153244303322533223

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. In *NIPS Proceedings*.

Bergström, S. (1976). Development and Application of a Conceptual Runoff Model for Scandinavian Catchments. *Smhi*. https://doi.org/10.1093/bjsw/bcn100

Camici, S., Brocca, L., Melone, F., & Moramarco, T. (2013). Impact of Climate Change on Flood Frequency Using Different Climate Models and Downscaling Approaches. *Journal of Hydrologic Engineering*. https://doi.org/10.1061/(asce)he.1943-5584.0000959

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Basic Concepts. In *Evolutionary Algorithms for Solving Multi-Objective Problems*. https://doi.org/10.1007/978-0-387-36797-2_1

Gneiting, T., & Katzfuss, M. (2014). *Probabilistic Forecasting*. *SSRN*. https://doi.org/10.1146/annurev-statistics-062713-085831

Gneiting, T., Raftery, A. E., Westveld, A. H., & Goldman, T. (2005). Calibrated Probabilistic

Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation. *Monthly Weather Review*. https://doi.org/10.1175/mwr2904.1

Haro-Monteagudo, D., Solera, A., & Andreu, J. (2017). Drought early warning based on optimal risk forecasts in regulated river systems: Application to the Jucar River Basin (Spain). *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2016.11.022

Harou, J. J., Pulido-Velazquez, M., Rosenberg, D. E., Medellín-Azuara, J., Lund, J. R., & Howitt, R. E. (2009). Hydro-economic models: Concepts, design, applications, and future prospects. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2009.06.037

Heinz, I., Pulido-Velazquez, M., Lund, J. R., & Andreu, J. (2007). Hydro-economic modeling in river basin management: Implications and applications for the European water framework directive. *Water Resources Management*. https://doi.org/10.1007/s11269-006-9101-8

Jakeman, A. J., & Hornberger, G. M. (1993). How much complexity is warranted in a rainfall-runoff model? *Water Resources Research*. https://doi.org/10.1029/93WR00877

Krause, P., Boyle, D. P., & Bäse, F. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*. https://doi.org/10.5194/adgeo-5-89-2005

Li, W., Duan, Q., Miao, C., Ye, A., Gong, W., & Di, Z. (2017). A review on statistical postprocessing methods for hydrometeorological ensemble forecasting. *Wiley Interdisciplinary Reviews: Water*. https://doi.org/10.1002/wat2.1246

Lindström, G., Johansson, B., Persson, M., Gardelin, M., & Bergström, S. (1997). Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology*. https://doi.org/10.1016/S0022-1694(97)00041-3

Luber, G., & McGeehin, M. (2008). Climate Change and Extreme Heat Events. *American Journal of Preventive Medicine*. https://doi.org/10.1016/j.amepre.2008.08.021

Merritt, W. S., Letcher, R. A., & Jakeman, A. J. (2003). A review of erosion and sediment transport models. In *Environmental Modelling and Software*. https://doi.org/10.1016/S1364-8152(03)00078-1

Moore, R. J. (2007). The PDM rainfall-runoff model. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-11-483-2007

Navarro-Ortega, A., Acuña, V., Batalla, R. J., Blasco, J., Conde, C., Elorza, F. J., Barceló, D. (2012). Assessing and forecasting the impacts of global change on Mediterranean rivers. The SCARCE Consolider project on Iberian basins. *Environmental Science and Pollution Research*. https://doi.org/10.1007/s11356-011-0566-5

Paredes-Arquiola, J., Andreu-Álvarez, J., Martín-Monerris, M., & Solera, A. (2010). Water Quantity and Quality Models Applied to the Jucar River Basin, Spain. *Water Resources Management*. https://doi.org/10.1007/s11269-010-9578-z

Penman, H. L. (1948). Evaporation from open water, bare soils and grass. *Proceedings of the*

Royal Society of London. Series A. Mathematical and Physical.

Piani, C., Haerter, J. O., & Coppola, E. (2010). Statistical bias correction for daily precipitation in regional climate models over Europe. *Theoretical and Applied Climatology*. https://doi.org/10.1007/s00704-009-0134-9

Powell, M. J. D. (2005). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*. https://doi.org/10.1093/comjnl/7.2.155

R. D. Harmel, T. L. Veith, J. G. Arnold, M. W. Van Liew, R. L. Bingner, & D. N. Moriasi. (2013). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. *Transactions of the ASABE*. https://doi.org/10.13031/2013.23153

Ritter, A., & Muñoz-Carpena, R. (2013). Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2012.12.004

Seibert, J. (1997). Estimation of parameter uncertainty in the HBV model. *Nordic Hydrology*. https://doi.org/10.2166/nh.1997.015

Thielen, J., Bartholmes, J., Ramos, M. H., & De Roo, A. (2009). The European flood alert system" part 1: Concept and development. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-13-125-2009

Thornthwaite, C. W. (1948). An Approach toward a Rational Classification of Climate. *Geographical Review*. https://doi.org/10.2307/210739

Thrasher, B., Maurer, E. P., McKellar, C., & Duffy, P. B. (2012). Technical Note: Bias correcting climate model simulated daily temperature extremes with quantile mapping. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-16-3309-2012

Vaze, J., Post, D. A., Chiew, F. H. S., Perraud, J. M., Viney, N. R., & Teng, J. (2010). Climate non-stationarity - Validity of calibrated rainfall-runoff models for use in climate change studies. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2010.09.018

Verkade, J. S., Brown, J. D., Reggiani, P., & Weerts, A. H. (2013). Post-processing ECMWF precipitation and temperature ensemble reforecasts for operational hydrologic forecasting at various spatial scales. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2013.07.039

Yang, T., Zhou, X., Yu, Z., Krysanova, V., & Wang, B. (2015). Drought projection based on a hybrid drought index using Artificial Neural Networks. *Hydrological Processes*. https://doi.org/10.1002/hyp.10394

Zappa, M., Beven, K. J., Bruen, M., Cofiño, A. S., Kok, K., Martin, E., Rossa, A. (2010). Propagation of uncertainty from observing systems and NWP into hydrological models: COST-731 working group 2. *Atmospheric Science Letters*. https://doi.org/10.1002/asl.248

ANNEX

This annex contains the code written in the MATLAB software for the bias-correction algorithm, divided by function, and the code written in VBA for the arrangement of the EXCEL sheets of the original forecasts.

MATLAB source codes: scripts

These source codes create and save variables directly in the current working directory.

assemble_matrices.m

```
% Assemble matrices to load for the results: after running
% temperature_correction and precipitation_correction (in this
order)
run temperature_correction ;
run precipitation_correction ;
% ALARCÓN
% prec
OBS_prec_Alarcón = [OBS_Alarcónp ; OBS_Alarcónp_v] ;
prec_Alarcón_ = [prec_Alarcón_c ; prec_Alarcón_v ] ;
prec_ens_Alarcón_ = [prec_ens_Alarcón_c ; prec_ens_Alarcón_v ] ;
Alarcón_prec_corr = QMAPP(prec_Alarcón_, prec_ens_Alarcón_, 1,
COEFF_Alarcónp, paramp) ;
Alarcón_pmean_corr = dailyavg(Alarcón_prec_corr) ;
Alarcón_pmean = dailyavg(prec_ens_Alarcón_) ;
% temp
OBS_temp_Alarcón = [OBS_Alarcónt ; OBS_Alarcónt_v] ;
temp_Alarcón_ = [temp_Alarcón_c ; temp_Alarcón_v ] ;
temp_ens_Alarcón_ = [temp_ens_Alarcón_c ; temp_ens_Alarcón_v ] ;
Alarcón_temp_corr = QMAPP(temp_Alarcón_, temp_ens_Alarcón_, 2,
COEFF_Alarcónt, paramt) ;
Alarcón_tmean_corr = dailyavg(Alarcón_temp_corr) ;
Alarcón_tmean = dailyavg(temp_ens_Alarcón_) ;
% CONTRERAS
% prec
OBS_prec_Contreras = [OBS_Contrerasp ; OBS_Contrerasp_v] ;
```

```matlab
prec_Contreras_  = [prec_Contreras_c ; prec_Contreras_v ] ;
prec_ens_Contreras_  = [prec_ens_Contreras_c ; prec_ens_Contreras_v ]
;
Contreras_prec_corr= QMAPP(prec_Contreras_, prec_ens_Contreras_, 1,
COEFF_Contrerasp, paramp) ;
Contreras_pmean_corr = dailyavg(Contreras_prec_corr) ;
Contreras_pmean = dailyavg(prec_ens_Contreras_) ;

% temp
OBS_temp_Contreras = [OBS_Contrerast ; OBS_Contrerast_v] ;
temp_Contreras_  = [temp_Contreras_c ; temp_Contreras_v ] ;
temp_ens_Contreras_  = [temp_ens_Contreras_c ; temp_ens_Contreras_v ]
;
Contreras_temp_corr= QMAPP(temp_Contreras_, temp_ens_Contreras_, 2,
COEFF_Contrerast, paramt) ;
Contreras_tmean_corr = dailyavg(Contreras_temp_corr) ;
Contreras_tmean = dailyavg(temp_ens_Contreras_) ;

% BELLÚS
% prec
OBS_prec_Bellús = [OBS_Bellúsp ; OBS_Bellúsp_v] ;
prec_Bellús_  = [prec_Bellús_c ; prec_Bellús_v ] ;
prec_ens_Bellús_  = [prec_ens_Bellús_c ; prec_ens_Bellús_v ] ;
Bellús_prec_corr= QMAPP(prec_Bellús_, prec_ens_Bellús_, 1,
COEFF_Bellúsp, paramp) ;
Bellús_pmean_corr = dailyavg(Bellús_prec_corr) ;
Bellús_pmean = dailyavg(prec_ens_Bellús_) ;

% temp
OBS_temp_Bellús = [OBS_Bellúst ; OBS_Bellúst_v] ;
temp_Bellús_  = [temp_Bellús_c ; temp_Bellús_v ] ;
temp_ens_Bellús_  = [temp_ens_Bellús_c ; temp_ens_Bellús_v ] ;
Bellús_temp_corr= QMAPP(temp_Bellús_, temp_ens_Bellús_, 2,
COEFF_Bellúst, paramt) ;
Bellús_tmean_corr = dailyavg(Bellús_temp_corr) ;
Bellús_tmean = dailyavg(temp_ens_Bellús_) ;
```

precipitation_correction.m

```matlab
% this script loads the original data and computes the coefficients
of the quantile
% mapping correction algorithm for the precipitation variable
[Alarcónp,date] =
xlsread('pr_Spain02_v50_daily.xlsx','A10964:B24112');
dtt = datetime(date,'InputFormat','dd/MM/yyyy');
date = datenum(dtt) ;
Alarcóndatap = [date,Alarcónp] ;
[paramp.mind, paramp.maxd] = minmax(Alarcóndatap, 0) ;
% resevoir: 1 for Alarcón, 2 for contreras, 3 for Bellús;
% sheetnr: 1 for precipitations, 2 for temp;
% start = starting year, fin = end year
% ISPREC = 1 treats the data as precipitation
start  = 1995 ;
fin    = 2005 ;
```

```
startv = fin+1 ;
finv   = 2015 ;
paramp.degree = 2;
paramp.nrpct = 20 ;
paramp.limit = 10^-3 ;
paramp.minrange = 10^-3 ;
sheetnr = 1 ;
ISPREC = sheetnr ;

% ALARCÓN: CALIBRATION
reservoir = 1 ;
% get the forecasts: by month and by ensemble, with date
[prec_Alarcón_c, prec_ens_Alarcón_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
% get the vector of dates of the calibration dataset
datec = date(year(dtt)<=fin & year(dtt)>=start) ;
% select the observations correspondent to the calibration dataset
OBS_Alarcónp = [datec Alarcónp(year(dtt)<=fin & year(dtt)>=start)] ;
% get the coefficients
COEFF_Alarcónp =
Quantilemapping_calibration(OBS_Alarcónp,prec_Alarcón_c, 1, paramp,
reservoir, 0) ;

% ALARCÓN : VALIDATION
% get forecasts data
[prec_Alarcón_v, prec_ens_Alarcón_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the date array
datev = date(year(dtt) <= finv & year(dtt) > fin) ;
% get the observation data
OBS_Alarcónp_v = [datev Alarcónp(year(dtt)<=finv & year(dtt) > fin)]
;

% CONTRERAS: CALIBRATION
Contrerasp = xlsread('pr_Spain02_v50_daily.xlsx','D10964:D24112');
Contrerasdatap = [date,Contrerasp] ;
[paramp.mind, paramp.maxd] = minmax(Contrerasdatap, 0) ;
reservoir = 2 ;
% get the forecasts: by month and by ensemble, with date
[prec_Contreras_c, prec_ens_Contreras_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
% select the observations correspondent to the calibration dataset
OBS_Contrerasp = [datec Contrerasp(year(dtt)<=fin &
year(dtt)>=start)] ;
% get the coefficients
COEFF_Contrerasp =
Quantilemapping_calibration(OBS_Contrerasp,prec_Contreras_c, 1,
paramp, reservoir, 0) ;

% CONTRERAS : VALIDATION
% get forecasts data
[prec_Contreras_v, prec_ens_Contreras_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the observation data
```

```matlab
OBS_Contrerasp_v = [datev Contrerasp(year(dtt)<=finv & year(dtt) >
fin)] ;

% BELLÚS: CALIBRATION
Bellúsp = xlsread('pr_Spain02_v50_daily.xlsx','H10964:H24112');
Bellúsdatap = [date,Bellúsp] ;
[paramp.mind, paramp.maxd] = minmax(Bellúsdatap, 0) ;
reservoir = 3 ;
% get the forecasts: by month and by ensemble, with date
[prec_Bellús_c, prec_ens_Bellús_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
% select the observations correspondent to the calibration dataset
OBS_Bellúsp = [datec Bellúsp(year(dtt)<=fin & year(dtt)>=start)] ;
% get the coefficients
COEFF_Bellúsp =
Quantilemapping_calibration(OBS_Bellúsp,prec_Bellús_c, 1, paramp,
reservoir, 0) ;

% BELLÚS : VALIDATION
% get forecasts data
[prec_Bellús_v, prec_ens_Bellús_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the observation data
OBS_Bellúsp_v = [datev Bellúsp(year(dtt)<=finv & year(dtt) > fin)] ;
```

temperature_correction.m

```matlab
% this script loads the original data and computes the coefficients
of the quantile
% mapping correction algorithm for the temperature variable
[Alarcónt,date] = xlsread('tas_Spain02_v50_daily.xlsx','A6:B13154');
dtt = datetime(date,'InputFormat','dd/MM/yyyy');
date = datenum(dtt) ;
Alarcóndatat = [date,Alarcónt] ;
[paramt.mind, paramt.maxd] = minmax(Alarcóndatat, 0) ;
% resevoir: 1 for Alarcón, 2 for contreras, 3 for Bellús;
% sheetnr: 1 for tempipitations, 2 for temp;
% start = starting year, fin = end year
% ISPREC = 1 treats the data as tempipitation
start  = 1995 ;
fin    = 2008 ; %performance is ~ constant for Alarcón and Bellús,
for contreras best is with fin = 2008
startv = fin+1 ;
finv   = 2015 ;
paramt.degree = 1; % doesn't change results much, so i select the
easiest
paramt.nrpct = 10 ; % doesn't change results much, so i select the
easiest
paramt.limit = 10^-3 ;
paramt.minrange = 10^-3 ;
sheetnr = 2 ;
ISPREC = sheetnr ;

% ALARCÓN: CALIBRATION
```

```matlab
reservoir = 1 ;
% get the forecasts: by month and by ensemble, with date
[temp_Alarcón_c, temp_ens_Alarcón_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
% get the vector of dates of the calibration dataset
datec = date(year(dtt)<=fin & year(dtt)>=start) ;
% select the observations correspondent to the calibration dataset
OBS_Alarcónt = [datec Alarcónt(year(dtt)<=fin & year(dtt)>=start)] ;
% get the coefficients
COEFF_Alarcónt =
Quantilemapping_calibration(OBS_Alarcónt,temp_Alarcón_c, 2, paramt,
1, 0) ;

% ALARCÓN : VALIDATION
% get forecasts data
[temp_Alarcón_v, temp_ens_Alarcón_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the date array
datev = date(year(dtt) <= finv & year(dtt) > fin) ;
% get the observation data
OBS_Alarcónt_v = [datev Alarcónt(year(dtt)<=finv & year(dtt) > fin)]
;

% CONTRERAS: CALIBRATION
Contrerast = xlsread('tas_Spain02_v50_daily.xlsx','D6:D13154');
Contrerasdatat = [date,Contrerast] ;
[paramt.mind, paramt.maxd] = minmax(Contrerasdatat, 0) ;
reservoir = 2 ;
% get the forecasts: by month and by ensemble, with date
[temp_Contreras_c, temp_ens_Contreras_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
% select the observations correspondent to the calibration dataset
OBS_Contrerast = [datec Contrerast(year(dtt)<=fin &
year(dtt)>=start)] ;
% get the coefficients
COEFF_Contrerast =
Quantilemapping_calibration(OBS_Contrerast,temp_Contreras_c, 2,
paramt, 2, 0) ;

% CONTRERAS : VALIDATION
% get forecasts data
[temp_Contreras_v, temp_ens_Contreras_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the observation data
OBS_Contrerast_v = [datev Contrerast(year(dtt)<=finv & year(dtt) >
fin)] ;

% BELLÚS: CALIBRATION
Bellúst = xlsread('tas_Spain02_v50_daily.xlsx','H6:H13154');
Bellúsdatat = [date,Bellúst] ;
[paramt.mind, paramt.maxd] = minmax(Bellúsdatat, 0) ;
reservoir = 3 ;
% get the forecasts: by month and by ensemble, with date
[temp_Bellús_c, temp_ens_Bellús_c] =
get_forecasts(sheetnr,reservoir,start,fin) ;
```

```
% select the observations correspondent to the calibration dataset
OBS_Bellúst = [datec Bellúst(year(dtt)<=fin & year(dtt)>=start)] ;
% get the coefficients
COEFF_Bellúst =
Quantilemapping_calibration(OBS_Bellúst,temp_Bellús_c, 2, paramt, 3,
0) ;


% BELLÚS : VALIDATION
% get forecasts data
[temp_Bellús_v, temp_ens_Bellús_v] =
get_forecasts(sheetnr,reservoir,startv,finv) ;
% get the observation data
OBS_Bellúst_v = [datev Bellúst(year(dtt)<=finv & year(dtt) > fin)] ;
```

results.m

```
% this is the file which provides the results of each subbasin
% Contents:
% Graphs of the CDFs of corrected and biased forecasts vs
observations;
% Observed vs simulated (with the bias-corrected forecasts) flow;
% Confrontation of the MAE of the biased vs corrected forecasts.
% for each resrvoir
% next command will take around to 20 minutes to complete
run Assemble_matrices
% if the workspace generated by the previous command have been
saved, load
% it with the following, without running the first one
load('final workspace.mat')
% ALARCÓN
reservoir = 1 ;
% Graphs of the CDFs of corrected and biased forecasts vs
observations:
% prec
figure ;
[~] = Quantilemapping_calibration(OBS_Alarcónp,prec_Alarcón_c, 1,
paramp, reservoir, 1) ;
figure ;
Quantilemapping_validation(OBS_Alarcónp_v, prec_Alarcón_v, 1,
COEFF_Alarcónp, paramp, reservoir, 1)
% temperature
figure ;
[~] = Quantilemapping_calibration(OBS_Alarcónt,temp_Alarcón_c, 2,
paramt, reservoir, 1) ;
% try the correction function on the validation dataset
figure ;
Quantilemapping_validation(OBS_Alarcónt_v, temp_Alarcón_v, 2,
COEFF_Alarcónt, paramt, reservoir, 1)
% Observed vs simulated flow
% watch out! put the parameter, clarea and simulation files before
starthbv=2005; finhbv = 2008;
for i = starthbv:finhbv
    figure;
```

```matlab
    HBV_files(Alarcóndatap, Alarcóndatat, prec_ens_Alarcón_,
temp_ens_Alarcón_, Alarcón_prec_corr, Alarcón_temp_corr, i, i,
reservoir)
end
% MAE temp
mae_data (1995, 2015, date, dtt, Alarcónt,
temp_ens_Alarcón_c(:,2:end), temp_ens_Alarcón_v(:,2:end),
temp_Alarcón_c, temp_Alarcón_v, 2, COEFF_Alarcónt, paramt,
reservoir)
% MAE prec
mae_data (1995, 2015, date, dtt, Alarcónp,
prec_ens_Alarcón_c(:,2:end), prec_ens_Alarcón_v(:,2:end),
prec_Alarcón_c, prec_Alarcón_v, 1, COEFF_Alarcónp, paramp,
reservoir)

% view the corrected vs biased time series
% prec
figure;
plot(datetime(OBS_prec_Alarcón(:,1),'ConvertFrom','datenum'),OBS_pre
c_Alarcón(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Alarcón_pmean(:,1),'ConvertFrom','datenum'),
Alarcón_pmean(:,2),'b-') ;
plot(datetime(Alarcón_pmean_corr(:,1),'ConvertFrom','datenum'),
Alarcón_pmean_corr(:,2),'r--') ;
title('Alarcón precipitation') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Precipitation [mm/d]') ;
% temp
figure;
plot(datetime(OBS_temp_Alarcón(:,1),'ConvertFrom','datenum'),OBS_tem
p_Alarcón(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Alarcón_tmean(:,1),'ConvertFrom','datenum'),
Alarcón_tmean(:,2),'b-') ;
plot(datetime(Alarcón_tmean_corr(:,1),'ConvertFrom','datenum'),
Alarcón_tmean_corr(:,2),'r--') ;
title('Alarcón temperature') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Temperature [°C]') ;

% CONTRERAS
reservoir = 2 ;
% Graphs of the CDFs of corrected and biased forecasts vs
observations:
% prec
figure ;
[~] = Quantilemapping_calibration(OBS_Contrerasp,prec_Contreras_c,
1, paramp, reservoir, 1) ;
figure ;
Quantilemapping_validation(OBS_Contrerasp_v, prec_Contreras_v, 1,
COEFF_Contrerasp, paramp, reservoir, 1)
% temperature
figure ;
[~] = Quantilemapping_calibration(OBS_Contrerast,temp_Contreras_c,
2, paramt, reservoir, 1) ;
figure ;
```

```matlab
Quantilemapping_validation(OBS_Contrerast_v, temp_Contreras_v, 2,
COEFF_Contrerast, paramt, reservoir, 1)
% Observed vs simulated flow
starthbv=2005; finhbv = 2008;
for i = starthbv:finhbv
    figure;
    HBV_files(Contrerasdatap, Contrerasdatat, prec_ens_Contreras_,
temp_ens_Contreras_, Contreras_prec_corr, Contreras_temp_corr, i, i,
reservoir)
end
% MAE temp
mae_data (1995, 2015, date, dtt, Contrerast,
temp_ens_Contreras_c(:,2:end), temp_ens_Contreras_v(:,2:end),
temp_Contreras_c, temp_Contreras_v, 2, COEFF_Contrerast, paramt,
reservoir)
% MAE prec
mae_data (1995, 2015, date, dtt, Contrerasp,
prec_ens_Contreras_c(:,2:end), prec_ens_Contreras_v(:,2:end),
prec_Contreras_c, prec_Contreras_v, 1, COEFF_Contrerasp, paramp,
reservoir)

% view the corrected vs biased time series
% prec
figure;
plot(datetime(OBS_prec_Contreras(:,1),'ConvertFrom','datenum'),OBS_p
rec_Contreras(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Contreras_pmean(:,1),'ConvertFrom','datenum'),
Contreras_pmean(:,2),'b-') ;
plot(datetime(Contreras_pmean_corr(:,1),'ConvertFrom','datenum'),
Contreras_pmean_corr(:,2),'r--') ;
title('Contreras precipitation') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Precipitation [mm/d]') ;
% temp
figure;
plot(datetime(OBS_temp_Contreras(:,1),'ConvertFrom','datenum'),OBS_t
emp_Contreras(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Contreras_tmean(:,1),'ConvertFrom','datenum'),
Contreras_tmean(:,2),'b-') ;
plot(datetime(Contreras_tmean_corr(:,1),'ConvertFrom','datenum'),
Contreras_tmean_corr(:,2),'r--') ;
title('Contreras temperature') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Temperature [°C]') ;

% BELLÚS
reservoir = 3 ;
% Graphs of the CDFs of corrected and biased forecasts vs
observations:
% prec
figure ;
[~] = Quantilemapping_calibration(OBS_Bellúsp,prec_Bellús_c, 1,
paramp, reservoir, 1) ;
figure ;
Quantilemapping_validation(OBS_Bellúsp_v, prec_Bellús_v, 1,
COEFF_Bellúsp, paramp, reservoir, 1)
```

```matlab
% temperature
figure ;
[~] = Quantilemapping_calibration(OBS_Bellúst,temp_Bellús_c, 2,
paramt, reservoir, 1) ;
figure ;
Quantilemapping_validation(OBS_Bellúst_v, temp_Bellús_v, 2,
COEFF_Bellúst, paramt, reservoir, 1)
% Observed vs simulated flow
starthbv=2009; finhbv = 2012;
for i = starthbv:finhbv
    figure;
    HBV_files(Bellúsdatap, Bellúsdatat, prec_ens_Bellús_,
temp_ens_Bellús_, Bellús_prec_corr, Bellús_temp_corr, i, i,
reservoir)
end
% MAE temp
mae_data (1995, 2015, date, dtt, Bellúst,
temp_ens_Bellús_c(:,2:end), temp_ens_Bellús_v(:,2:end),
temp_Bellús_c, temp_Bellús_v, 2, COEFF_Bellúst, paramt, reservoir)
% MAE prec
mae_data (1995, 2015, date, dtt, Bellúsp,
prec_ens_Bellús_c(:,2:end), prec_ens_Bellús_v(:,2:end),
prec_Bellús_c, prec_Bellús_v, 1, COEFF_Bellúsp, paramp, reservoir)

% view the corrected vs biased time series
% prec
figure;
plot(datetime(OBS_prec_Bellús(:,1),'ConvertFrom','datenum'),OBS_prec
_Bellús(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Bellús_pmean(:,1),'ConvertFrom','datenum'),
Bellús_pmean(:,2),'b-') ;
plot(datetime(Bellús_pmean_corr(:,1),'ConvertFrom','datenum'),
Bellús_pmean_corr(:,2),'r--') ;
title('Bellús precipitation') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Precipitation [mm/d]') ;
% temp
figure;
plot(datetime(OBS_temp_Bellús(:,1),'ConvertFrom','datenum'),OBS_temp
_Bellús(:,2),'Color',0.7*[1,1,1]) ; hold on ;
plot(datetime(Bellús_tmean(:,1),'ConvertFrom','datenum'),
Bellús_tmean(:,2),'b-') ;
plot(datetime(Bellús_tmean_corr(:,1),'ConvertFrom','datenum'),
Bellús_tmean_corr(:,2),'r--') ;
title('Bellús temperature') ; legend('Observations', 'original
forecasts', 'bias-corrected forecasts','location','Northeast');
ylabel('Temperature [°C]') ;
```

test_parameters.m

```matlab
% this script is created to conveniently test the parameters of the
% bias-correction algorithm
start  = 1995 ;
fin    = 2006 ;
startv = fin+1 ;
```

```
finv    = 2015 ;
param.degree = 1; % doesn't change results much, so i select the
easiest
param.nrpct = 10 ; % doesn't change results much, so i select the
easiest
param.limit = 10^-3 ;
param.minrange = 10^-3 ;
sheetnr = 2 ;
ISPREC = sheetnr ;

% Alarcón
reservoir = 1 ;
COEFF_Alarcón =
Quantilemapping_calibration(OBS_Alarcón,temp_Alarcón_c,ISPREC,
param, reservoir, 0) ;
mae_data (start, finv, date, dtt, Alarcón, temp_ens_Alarcón_c,
temp_ens_Alarcón_v, temp_Alarcón_c, temp_Alarcón_v, ISPREC,
COEFF_Alarcón, param, reservoir)

% Contreras
reservoir = 2 ;
COEFF_Contreras =
Quantilemapping_calibration(OBS_Contreras,temp_Contreras_c,ISPREC,
param, reservoir, 0) ;
mae_data (start, finv, date, dtt, Contreras, temp_ens_Contreras_c,
temp_ens_Contreras_v, temp_Contreras_c, temp_Contreras_v, ISPREC,
COEFF_Contreras, param, reservoir)

% Bellús
reservoir = 3 ;
COEFF_Bellús =
Quantilemapping_calibration(OBS_Bellús,temp_Bellús_c,ISPREC, param,
reservoir, 0) ;
mae_data (start, finv, date, dtt, Bellús, temp_ens_Bellús_c,
temp_ens_Bellús_v, temp_Bellús_c, temp_Bellús_v, ISPREC,
COEFF_Bellús, param, reservoir)
```

MATLAB source codes: functions

These source codes need to be called from regular scripts, they create the variables in their
own workspace. They are useful when a same task has to be repeated for different cases.

dailyavg.m

```
% this function computes the average forecasted value by day: first,
it
% computes the ensemble mean, then the average forecast of each day
function [avg_by_day] = dailyavg(ens)
mean_ =[ens(:,2) mean(ens(:,3:27),2) ];
mean_ = sortrows(mean_,1) ;
starti = mean_(1,1) ;
```

```matlab
endi = mean_(end,1) ;
avg_by_day = nan(endi-starti,2) ;
for i = starti:endi
    idx = mean_(:,1) == i ;
    avg_by_day(i-starti+1,2) = mean(mean_(idx,2));
    avg_by_day(i-starti+1,1) = i ;
end
end
```

get_flow.m

```matlab
% this function reads the data contained in the PTQ file to get the
% observed flow time series
function [Q] = get_flow(reservoir)
sizeA = [4 6920];
if reservoir ==1
    name = "PTQ_Alarcón_c.txt" ;
    namee ="Alarcón_EVAP.txt" ;
elseif reservoir ==2
    name = "PTQ_Contreras_c.txt" ;
    namee ="Contreras_EVAP.txt" ;
else
    name = "PTQ_Bellús_c.txt" ;
    namee ="Bellús_EVAP.txt" ;
    sizeA = [4 5812] ;
end
% open PTQ file
fileID = fopen(name) ;
% specify file format for the fscanf function
formatSpec = '%f %f %f %f';
% read the file and assign it to a variable
Q = (fscanf(fileID,formatSpec,sizeA))' ;
fclose(fileID) ;
% get the date
S = num2str( Q(:,1));
% first 4 characters are the year, then 2 for month and 2 for day
y = S(:,1:4) ;
m = S(:,5:6) ;
d = S(:,7:8) ;
% assemble them together in the ddmmyyyy format
dmy = [ d m y ] ;
% conversion to the desired format
date = datetime(dmy,'InputFormat','ddMMyyyy') ;
date = datenum(date) ;
% definitive Q variable which will be used
Q = [date Q(:,4) ] ;
%EVAP = load ('Alarcón_EVAP.txt','-ascii') ;
%EVAP = [date EVAP] ;
end
```

get_forecasts.m

```matlab
% this function reads the forecasts from the EXCEL worksheets. It
returns
% the values as they are in the worksheet and arranged by month
```

```matlab
function [y_m,ensemble] = get_forecasts(sheetnr,reservoir,start,fin)
% create first part of name string
name = 'ECMWF_forecast_values_daily_' ;
% select appropriate range
if reservoir == 1
    range = 'C8:AA2587' ;
elseif reservoir == 2
    range = 'BA8:BY2587' ;
else
    range = 'EW8:FU2587' ;
end
% preallocate variables for speed
forecast1 = nan(2580,25) ;
forecast = nan(2580,25) ;
y_m = nan(5500*(fin-start+1),12) ;
ensemble = nan(2580*(fin-start+1),27) ;
for i = start:fin
i_str    = num2str(i-1) ;
i_str1   = num2str(i) ;
name_i   = strcat(name,i_str,'.xlsx') ; %creation of the name string
of year i-1
name_i1  = strcat(name,i_str1,'.xlsx') ; %creation of the name
string of year i
% read the file of year i-1
forecast = xlsread(name_i,sheetnr,range) ;
[~,dt]   = xlsread(name_i,sheetnr,'B8:B2587') ;   % reading the date
of year i-1
dt       = datetime(dt,'Inputformat','dd/MM/yyyy') ;
% read the file of year i
forecast1 = xlsread(name_i1,sheetnr,range) ;
[~,dt1] = xlsread(name_i1,sheetnr,'A8:B2587') ; % reading the date
of year i
dt1 = datetime(dt1,'Inputformat','dd/MM/yyyy');
% create the ensemble matrix:
% convert datetime to datenum
dtn1 = datenum(dt1) ;
% create the definitive ensemble file
ensemble(2580*(i-start)+1:2580*(i-start+1),1:2) = dtn1 ;
ensemble(2580*(i-start)+1:2580*(i-start+1),3:27) = forecast1 ;
for j = 1:12
    % in this loop, get all the values correspondent to the MONTH J
and
    % YEAR I
    % get all the values of the month j of year i from the 1st
dataset
    mont = forecast((month(dt) == j & year(dt) == i ), :)  ;
    % put them in an array
    mont = mont(:) ;
    % get all the values of the month j of year i from the second
dataset
    mont1 = forecast1((month(dt1(:,2)) == j & year(dt1(:,2)) == i ),
:) ;
    mont1 = mont1(:) ;
    % put all the values together
    mont2 = [mont; mont1] ;
```

```
    % fill the matrix of forecasts: first data is first forecasts
issued in
    % year i-1 for Jan 1st of year i
    % the max length is 5500, but not every month has 5500
forecasts: fill
    % the rest with nan
    y_m(1+5500*(i-start):5500*(1+i-start),j) = [mont2;nan(5500-
length(mont2),1)] ;
    % filter: if sheetnr = 1 (precipitation) put to 0 negative
values
    if sheetnr == 1
    y_m(y_m < 0) = 0 ;
    end
end
end
end
```

HBV_files.m

```
% this function creates the files needed to run the HBV model and
runs the
% simulations. It returns the plots containing the simulations
obtained by
% feeding the model with the original forecasts ensemble mean and
the 25th
% and 75th percentile and the median of the simulations obtained
feeding
% the model with the corrected forecasts ensemble members.
function [] = HBV_files(OBSp, OBSt, forp, fort, corr_forp,
corr_fort, start, fin, reservoir)
% create string of names for the graph
if reservoir ==1
    name = "Alarcon" ;
elseif reservoir ==2
    name = "Contreras" ;
else
    name = "Bellus" ;
end
ptqheader = [name ; "Date    P[mm/d] T[°C]   Q[mm/d]"] ; % create the
header of the file to be printed

% load observations
D_OBS = OBSp(:,1); %date of the obs
OBSdatat = OBSt(:,2) ; %values of the temp obs
OBSdatap = OBSp(:,2) ; %values of the prec obs
D_OBS1 = datetime(D_OBS,'ConvertFrom','datenum');
% select the values from start to fin
OBS_idx = year(D_OBS1)>=(start-1) & year(D_OBS1)<=fin ; %indicator
to select values within start-1
% (because jan of the first year of forecasts requires data of the
previous year)and fin years
D_OBS = D_OBS(OBS_idx);        % obs date limited to indicator
OBSdatap = OBSdatap(OBS_idx) ; % prec obs values limited to
indicator
```

```matlab
OBSdatat = OBSdatat(OBS_idx) ; % temp obs values limited to
indicator

%load forecasts
D_FOR = corr_forp(:,1) ; % date of issued forecasts
FORdatap = corr_forp(:,2:end) ; % values of prec forecasts with
target date
FORdatat = corr_fort(:,2:end) ; % values of temp forecasts with
target date
D_FOR1 = datetime(D_FOR,'ConvertFrom','datenum');
% select the values from start to fin
FOR_idx = year(D_FOR1)>=start & year(D_FOR1)<=fin ; %indicator to
select values within start and fin years
D_FOR = D_FOR(FOR_idx);        % for issue date limited to indicator
FORdatap = FORdatap(FOR_idx, :) ; % for prec values limited to
indicator
FORdatat = FORdatat(FOR_idx, :) ; % for temp values limited to
indicator

%load flow
Q = get_flow(reservoir) ;
D_Q = Q(:,1) ; % date of flow
Qdata = Q(:,2:end) ; % values of flow
D_Q1 = datetime(D_Q,'ConvertFrom','datenum');
% select the values from start to fin
Q_idx = year(D_Q1)>=(start-1) & year(D_Q1)<=fin+1 ; %indicator to
select values within start and fin years
D_Q = D_Q(Q_idx);       % Q date limited to indicator
Qdata = Qdata(Q_idx) ; % Q values limited to indicator

% load daylight duration file for EVAP file creation
daydur = load('daylight duration.txt', '-ascii') ;
% temp = [FORdatat(1:215,1) mean(FORdatat(1:215,2:end),2)] ;
% EVAP = thornthwaite(daydur,temp) ;
%evapname = compose(formatSpecd,name, '_EVAP.txt') ;
%EVAP = load(evapname, '-ascii') ;
%EVAPdata = EVAP(Q_idx) ; %select the values in the same position of
the Q. They correspond to the same date

%create the new folder:
%folder name:
formatSpecb = "%s%s" ;
newfolder = compose(formatSpecb,name, "\data\") ;
%catchment folder direction:
catchment = char(compose(formatSpecb,
'C:/Users/avefe/Desktop/tesi/hbv/', name)) ;

% preallocation
resultsdata = nan(215,25) ;

for i = 1:12*(fin-start+1)
      % get the dates
    printtargetdate = FORdatap(1+(i-1)*215:i*215,1) ; %select the
target dates of forecasts issued in month i
```

```matlab
    printissuedatei = D_FOR(1+(i-1)*215:i*215) ;      %select the
issue date with position correspondent to the month i

    % assemble the matrix to create the prec + temp variables of the
PTQ file: 6 months of obs and 7 of
    % forecasts
    idx = D_OBS<printissuedatei(1) & D_OBS>=(printissuedatei(1)-150)
; %150 days of warm up period, till the last day before forecasts
    OBSprint = [ OBSdatap(idx) OBSdatat(idx)] ; % first 5 months of
warm up
    totdate = [ D_OBS(idx) ; printtargetdate ] ; % returns the
complete date column
    idxq = D_Q<=max(totdate) & D_Q>=min(totdate) ; % index that
selects the dates = totdate for the observed flow
    Qprint = Qdata(idxq) ; %select the observed flow values
correspondent to the totdate index
    totdatedt = datetime(totdate,'ConvertFrom','datenum') ;
    totdate1 = datestr(totdatedt,'yyyymmdd') ; % put the date in the
format needed by the HBV program
    resultsdate = datetime(printtargetdate,'ConvertFrom','datenum')
;

    for k = 2:26
    printforp = FORdatap(1+(i-1)*215:i*215,k) ;      %select the prec
values with position correspondent to the month i, for the ensemble
k
    printfort = FORdatat(1+(i-1)*215:i*215,k) ;      %select the temp
values with position correspondent to the month i, for the ensemble
k

    FORprint = [ printforp printfort] ; % 7 months forecasts

    % assemble the matrix to create the temp file to feed to the
thornthwaite function: 365 days of data
    evapprint = [OBSdatat(idx) ; printfort] ; % first 150 days of
obs + 215 days of forecasts

    % call the function and create the EVAP array for the EVAP file
    printevap = thornthwaite(daydur,[totdate evapprint]) ;

    % Create the PTQ file
    print1 = [OBSprint ; FORprint] ; % first 6 months of
observations for warm up, then 215 days of forecasts

    % create the name of the PTQ file

    ptqfilename =
char(compose(formatSpecb,datestr(datetime(printissuedatei(1),'Conver
tFrom','datenum'),'yyyy-mm-dd'), "_PTQ.txt")) ;
    evapfilename =
char(compose(formatSpecb,datestr(datetime(printissuedatei(1),'Conver
tFrom','datenum'),'yyyy-mm-dd'), "_EVAP.txt")) ;

    % create the name of the destination folder
    formatSpecc = "%s%s%s" ;
    % direction of the PTQ file with the forecasts
```

```matlab
    ptqfiledir =
compose(formatSpecc,"C:/Users/avefe/Desktop/tesi/hbv/",
newfolder,ptqfilename) ;

    % direction of the EVAP file
    destevap =
compose(formatSpecc,"C:/Users/avefe/Desktop/tesi/hbv/",
newfolder,evapfilename) ;

    % create the PTQ file
    % create the matrix to be printed in the file: first column is
the
    % date, 2nd is the prec, 3rd temp, 4th observed runoff
    formatSpec = "%s\t%f\t%f\t%f" ;
    ptq = compose(formatSpec,totdate1, print1(:,1), print1(:,2) ,
Qprint) ;
    ptq = [ptqheader;ptq] ;  % total matrix needed
    fileID = fopen(ptqfiledir,'w');
    fprintf(fileID,"%s\r\n",ptq) ;
    fclose(fileID) ;
    clear fileID

    % create the EVAP file with the corrected forecasts
    evapfile = num2str(printevap) ;
    evapfile = [name ; evapfile] ;
    fileID1 = fopen(destevap,'w');
    fprintf(fileID1,"%s\r\n",evapfile) ;
    fclose(fileID1) ;
    clear fileID1

    % after the creation of the HBV files, run the program itself
and save
    % the results
    % set parameters: Save the simulation results and set start and
end of the
    % simulation period
    simstart =
datestr(datetime(printtargetdate(1),'ConvertFrom','datenum'),'yyyymm
dd') ; %format required
    simend =
datestr(datetime(printtargetdate(end),'ConvertFrom','datenum'),'yyyy
mmdd') ;
    resultname =
char(compose(formatSpecb,datestr(datetime(printissuedatei(1),'Conver
tFrom','datenum'),'yyyy-mm-dd'), "_Res")) ;
    summaryname =
char(compose(formatSpecb,datestr(datetime(printissuedatei(1),'Conver
tFrom','datenum'),'yyyy-mm-dd'), "_summary.txt")) ;
  eval(['!"C:/Program Files (x86)/HBV-light/HBV-light-CLI.exe"
Settings ',catchment,' /SaveResults /Start:', simstart,' /End:',
simend,' /ptq:',ptqfilename,' /evap:',evapfilename]);
  eval(['!"C:/Program Files (x86)/HBV-light/HBV-light-CLI.exe" Run
',catchment,' SingleRun Results /results:',resultname,'
/ptq:',ptqfilename,' /evap:',evapfilename,'
/summary:',summaryname]);
```

```matlab
    % load results file:
    formatSpece = "%s%s%s%s" ;
    resultspath = compose(formatSpece,
catchment,"/Results/",resultname,".txt") ;
    fileID2 = fopen(resultspath,'r') ;
    % get the cell array of date, simulated and observed flow
    results = textscan(fileID2,'%s %f %f %*[^\n]', 'headerLines',1)
;
    % conversion into matrix
    % first column is the simulated runoff, second is the observed
runoff

    resultsdata(:,k-1) = cell2mat(results(2)) ;
    fclose(fileID2) ;

    end

%plot
 set(gcf,'position',[ 530, 190, 1111, 794])
    subplot(3,4,i)
    plot(resultsdate,observedrunoff,'Color',0.7*[1,1,1],
'linewidth',3)
    hold on
    plot(resultsdate,median(resultsdata,2), 'r-','linewidth',2)
    plot(resultsdate,prctile(resultsdata,25,2), 'b--','linewidth',1)
    plot(resultsdate,prctile(resultsdata,75,2), 'b--','linewidth',1)
    xlabel('date'), ylabel('runoff [mm/d]')
    xlim([min(resultsdate) max(resultsdate)]) ;
    ylim([0 0.8])
    monthTitle
=['Jan';'Feb';'Mar';'Apr';'May';'Jun';'Jul';'Aug';'Sep';'Oct';'Nov';
'Dec'];
    title(monthTitle(i,:),'fontweight','bold','fontsize',10), grid
on
    if i==12
    sgtitle(name) ;
    end
end
end
```

MAE.m

```matlab
% this function calculates the Mean Average error with respect to
the
% observations of the original forecasts and of the bias-corrected
forecasts
function [BIASmae, CORRmae] = MAE(OBS, BIAS, CORR)
%organize observations
D_OBS = OBS(:,1);
D_OBS = datetime(D_OBS,'ConvertFrom','datenum');
OBSdata=(OBS(:,2));
%organize biased forecasts
```

```matlab
D_BIAS = BIAS(:,1) ;
D_BIAS = datetime(D_BIAS,'ConvertFrom','datenum');
BIASdata = BIAS(:,2:end) ;


D_CORR = CORR(:,1) ;
D_CORR = datetime(D_CORR,'ConvertFrom','datenum');
CORRdata = CORR(:,2:end) ;
% variables preallocation
OBSmean = nan(12,1) ;
BIASmean = nan(12,size(BIASdata,2)) ;
CORRmean = nan(12,size(BIASdata,2)) ;
BIASmae = nan(12,1) ;
CORRmae = nan(12,1) ;
BIASabserr = nan(size(BIASdata,2),1) ;
CORRabserr = nan(size(BIASdata,2),1) ;
for i = 1:12
    % mean value of the observation for the month i
    OBSmean(i) = mean(OBSdata(month(D_OBS)==i)) ;
    for j = 1:size(BIASdata,2)
        % select the ensemble member j
        BIASdataj = BIASdata(:,j) ;
        CORRdataj = CORRdata(:,j) ;
        % mean value of the forecasts for the ensemble j and month i
    BIASmean(i,j) = mean(BIASdataj(month(D_BIAS)==i)) ;
    CORRmean(i,j) = mean(CORRdataj(month(D_CORR)==i)) ;
        % absolute error of the forecasts for the ensemble j and month
i
    BIASabserr(j) = abs(BIASmean(i,j)-OBSmean(i)) ;
    CORRabserr(j) = abs(CORRmean(i,j)-OBSmean(i)) ;
    end
    BIASmae(i) = mean(BIASabserr) ;
    CORRmae(i) = mean(CORRabserr) ;

end
end
```

mae_data.m

```matlab
% this function creates the bar plots of the MAE: sum by year, month
% average and monthly
function [] = mae_data (start, finv, date, dtt, OBS, ens_c, ens_v,
for_c, for_v, ISPREC, COEFF, param, reservoir)
% create string of names for the graph
if ISPREC ==1
    variable = 'precipitation' ;
else
    variable = 'temperature' ;
end
if reservoir ==1
    name = "Alarcón" ;
elseif reservoir ==2
    name = "Contreras" ;
else
    name = "Bellús" ;
end
```

```matlab
formatSpec = "%s%s%s" ;
titlestring = compose(formatSpec,name, " ", variable) ;
% MAE
sumBIAS = nan(finv-start+1,1) ;
sumCORR = nan(finv-start+1,1) ;
BIASmae = nan(finv-start+1,12) ;
CORRmae = nan(finv-start+1,12) ;
% "assemble" the biased forecast ensembles: date
date_ens1 = ens_c(:,1) ;
date_ens1 = datetime(date_ens1,'ConvertFrom','datenum');
date_ens2 = ens_v(:,1) ;
date_ens2 = datetime(date_ens2,'ConvertFrom','datenum');
% get the corrected forecasts
for_ = [for_c;for_v] ;
for yr = start:finv
% get the observations correspondent to the year yr
OBS_ = [date(year(dtt) == yr) OBS(year(dtt) == yr)] ;
% get the forecasts correspondent to the year yr and pile them
together:
% ensemble
prec_ens = [ens_c(year(date_ens1) == yr, :) ; ens_v(year(date_ens2)
==yr, :)] ;
% by month
for_yr = for_(1+5500*(yr-start):5500*(yr-start+1),:) ;
% get the corrected forecasts
prec_corr = QMAPP(for_yr, prec_ens, ISPREC, COEFF, param) ;

[BIASmae(yr-start+1,:), CORRmae(yr-start+1,:)] = MAE(OBS_, prec_ens,
prec_corr) ;
sumBIAS(yr-start+1) = sum(BIASmae(yr-start+1,:)) ;
sumCORR(yr-start+1) = sum(CORRmae(yr-start+1,:)) ;
end
if finv ~=start
%plot the yearly MAE sum
figure ;
bar([(start:finv)' (start:finv)'],[sumBIAS, sumCORR]);
title([titlestring,'Yearly MAE sum']) ;
legend('Biased forecast error', 'Corrected forecast error');
xlabel('year') ; ylabel('MAE value') ;
%plot the average MAE for all the Jan, Feb, March, etc.
figure ;
bar([(mean(BIASmae))', (mean(CORRmae))']);
title([titlestring,'Average monthly MAE']) ;
legend('Biased forecast error', 'Corrected forecast error');
xlabel('month') ; ylabel('MAE value') ;
end
% plot the monthly MAE
% get the month-year array
idx = year(dtt)>=start & year(dtt)<=finv ;
y = year(dtt(idx)) ;
m = month(dtt(idx),'shortname') ;
m_y = [char(m) num2str(y)];
m_y = (unique(m_y(:,:),'stable','rows')) ;
m_y = cellstr(m_y) ;


% transpose the matrix
```

```matlab
BIASmae1 = BIASmae' ;
CORRmae1 = CORRmae' ;
figure ;
% BIASmae(:) rearranges the matrix into a vector, by column. for
this
% reason it was necessary to transpose before
bar([BIASmae1(:), CORRmae1(:)]); title([titlestring,'Monthly MAE'])
;
xticks(1:length(m_y)) ;
xticklabels(m_y) ;
legend('Biased forecast error', 'Corrected forecast error');
xlabel('date') ; ylabel('MAE value') ;
end
```

minmax.m

```matlab
% this function returns the minimum and maximum value of each month
of data
% input
function [mindata, maxdata] = minmax(OBS,percincrease)
data = OBS(:,2) ;
date = OBS(:,1) ;
date = datetime(date,'ConvertFrom','datenum');
maxdata = nan(12,1) ;
mindata = nan(12,1) ;
for i = 1:12
    monthdata=(data(month(date)==i));
    maxdata(i) = max(monthdata)*(1+percincrease) ;
    mindata(i) = min(monthdata)*(1-percincrease) ;
    if mindata(i)<0
    mindata(i) = min(monthdata)*(1+percincrease) ;
    end
end
```

QMAPP.m

```matlab
% this function corrects the forecasts with the coefficients
previously
% found
function [ens_corr_ok] = QMAPP(FORECASTS, ENSEMBLE, ISPREC, COEFF,
param)
%----------------------------------------------------------------------
-------
% quantile mapping function

% parameters
limit = param.limit ;
nrpct = param.nrpct ;
degree = param.degree ;
% filters
lw = param.mind ;
hi = param.maxd ;
%initialize for speed
thrsh = nan(12,nrpct) ;
thrsl = nan(12,nrpct) ;
```

```matlab
% get date and the ensemble mean
ens_date = datetime(ENSEMBLE(:,2), 'ConvertFrom','datenum') ;
ens_date1 = ENSEMBLE(:,1) ;
ens_corr = ENSEMBLE(:,3:end) ;
ens_corr_ok = ens_corr ;
if ISPREC ==1
ens_corr(ens_corr<=limit) = 0 ;
ens_corr_ok(ens_corr_ok<=limit) = 0 ;
end

for i=1:12      % for each month of the year
    %remove nan and select column correspondent to month i
    FORmont = rmmissing(FORECASTS(:,i)) ;
    % set forecasts lower than the limit to 0(if data is
precipitation)
    if ISPREC == 1
    FORmont(FORmont<limit) = 0 ;
    end
    % calculate the cumulative probability
    PFOR = (1:length(FORmont))'./(length(FORmont)+1);
    %sort forecasts
    sortFOR = sort(FORmont) ;

   % find an equation for each 10th percentile
   for j = 1:nrpct
    % find the lower and higher thresholds for each percentile range
    thrsh(i,j) = max(sortFOR(PFOR<=j*(1/nrpct) & PFOR>j*(1/nrpct)-
(1/nrpct))) ;
    thrsl(i,j) = min(sortFOR(PFOR<=j*(1/nrpct) & PFOR>j*(1/nrpct)-
(1/nrpct))) ;
    % correction
    % conditions: lower than the higher threshold,
    % higher than the lower threshold, in the month i
    % correction applied to each ensemble member
    for k =1:size(ens_corr,2)
    idx = (ens_corr(:,k)<=thrsh(i,j) & ens_corr(:,k)>=thrsl(i,j) &
month(ens_date)==i ) ;
    ens_corr_ok(idx,k) = polyval(COEFF(i,(j-1)*(degree+1)+1:(j-
1)*(degree+1)+1+degree), ens_corr(idx,k))+ens_corr(idx,k) ;
    end
   end
   %if ISPREC == 1
   % filter for low values (of each month)
   ens_corr_ok(ens_corr_ok<=lw(i) & month(ens_date)==i ) = lw(i) ;
   % filter for high values (of each month)
   ens_corr_ok(ens_corr_ok>=hi(i) & month(ens_date)==i) = hi(i) ;
   %end
end

  if ISPREC == 1
  ens_corr_ok(ens_corr_ok<0) = 0 ;
  end
  ens_date = datenum(ens_date) ;
  ens_corr_ok =  [ens_date1 ens_date ens_corr_ok] ;
```

```
end
```

quantilemapping_calibration.m

```matlab
% this function calculates the coefficients of the quantile mapping
% correction to be applied to the forecasts
function [COEFF_ok] =
Quantilemapping_calibration(OBS,FORECASTS,ISPREC, param, reservoir,
graph)
%------------------------------------------------------------------
-------
%                            CDF matching approach
% The function allows to rescale the input data (FOR) to match the
% Cumulative Distribution Function (CDF) of benchmark data (OBS).
% create string of names for the graph
if reservoir ==1
    name = "Alarcón" ;
elseif reservoir ==2
    name = "Contreras" ;
else
    name = "Bellús" ;
end
if ISPREC ==1
    name2 = "Precipitation" ;
else
    name2 = "Temperature" ;
end
% prepare observation data
D_OBS = OBS(:,1);
D_OBS = datetime(D_OBS,'ConvertFrom','datenum');
OBSdata=(OBS(:,2));
% load parameters: equation degree and percentile subdivisions
degree = param.degree ;
nrpct = param.nrpct ;
% lower boundary for the values(precipitation): values below are set
= 0
limit = param.limit ;
% minrange: supposedly the measurement error
minrange = param.minrange ;
% filters
lw = param.mind ;
hi = param.maxd ;
%preallocate for speed
COEFF_ok = nan(12,(degree+1)*nrpct) ;
FORCDF = nan(length(FORECASTS),1) ;
for i=1:12      % for each month of the year
    OBSmont = OBSdata(month(D_OBS)==i);
    % account for measurement error: observations lower than limit
    % approximated to 0. This applies only with precipitation data
    if ISPREC == 1
    OBSmont(OBSmont<limit) = 0 ;
    end
    %remove nan and select column correspondent to month i
    FORmont = rmmissing(FORECASTS(:,i)) ;
    % set forecasts lower than the limit to 0(if data is
precipitation)
```

```matlab
    if ISPREC == 1
    FORmont(FORmont<limit) = 0 ;
    end
    % calculate the cumulative probabilities
    POBS = (1:length(OBSmont))'./(length(OBSmont)+1);
    PFOR = (1:length(FORmont))'./(length(FORmont)+1);
    %sort forecast and observed values
    sortFOR = sort(FORmont) ;
    sortOBS = sort(OBSmont) ;
    % values correspondent to high and low percentiles filters
    % find an equation for each 10th percentile
  for j = 1:nrpct
    % cumulative probabilities of each percentile range
    POBSj = POBS(POBS<=j*(1/nrpct) & POBS>j*(1/nrpct)-(1/nrpct)) ;
    PFORj = PFOR(PFOR<=j*(1/nrpct) & PFOR>j*(1/nrpct)-(1/nrpct)) ;
    sortFORj = sortFOR(PFOR<=j*(1/nrpct) & PFOR>j*(1/nrpct)-
(1/nrpct)) ;
    sortOBSj = sortOBS(POBS<=j*(1/nrpct) & POBS>j*(1/nrpct)-
(1/nrpct)) ;
    % Interpolation of input data (sortFORj), correspondent to the
    % percentiles PFORj, to the same percentiles (POBSj) of
benchmark data (OBS)
    FORint   = interp1(PFORj,sortFORj,POBSj,'linear','extrap');

    % Computation of the differences between the CDFs of OBS and SAT
data
    DIFF = sortOBSj-FORint;

    % Fitting of a polynomial curve to DIFF: if the observed values
are all
    % within a small range, then approximate with the average,
otherwise
    % use a (degree) degree polynomial
    if range(sortOBSj) <= minrange
        COEFF = polyfit(FORint,DIFF, 0);
    else
        COEFF= polyfit(FORint,DIFF, degree);
    end
    % fill the COEFFj vector: it contains the coefficients of each
    % percentile range for the month i
  COEFFj(1+(j-1)*(degree+1):j*(degree+1)) = COEFF ;
    % Evaluation of the polynomial curve to SAT data
  FORCDF(FORmont<=max(sortFORj)& FORmont>=min(sortFORj))=
polyval(COEFF,FORmont(FORmont<=max(sortFORj)&
FORmont>=min(sortFORj)))+FORmont(FORmont<=max(sortFORj)&
FORmont>=min(sortFORj));
    if ISPREC == 1
        FORCDF(FORCDF<0) = 0 ;
    end

    end
      % filter for low values (only temperature)
    %if ISPREC == 1
    FORCDF(FORCDF<=lw(i)) = lw(i) ;
    % filter for high values
    FORCDF(FORCDF>=hi(i)) = hi(i) ;
```

```matlab
  % end
   % fill the COEFF matrix, each row represents the coefficients of
a month
    COEFF_ok(i,:) = COEFFj ;
    if graph ==1
    % graphs
    % Comparison of cdf curves estimated for: benchmark data
(OBSmont),
    % data to modify (FORmont), corrected data (FORCDF)
    % set min and max of the graphs:
    ming = min(OBSmont)-1;
    if ISPREC == 1
        ming = 0 ;
    end
    maxg = max(OBSmont)+1 ;
    set(gcf,'position',[ 530, 190, 1111, 794])
    subplot(3,4,i)
    plot(
sort(OBSmont),(1:length(OBSmont))/(length(OBSmont)+1),'Color',0.7*[1
,1,1], 'linewidth',7)
    hold on
    plot(sort(FORmont),(1:length(FORmont))/(length(FORmont)+1), 'b-
','linewidth',4)
    % plot the corrected CDF, eliminating nan values
    plot(
sort(rmmissing(FORCDF)),(1:length(rmmissing(FORCDF)))/(length(rmmiss
ing(FORCDF))+1), 'r--', 'linewidth',2)
    xlabel('data'), ylabel('Cumulative Density Function')
    xlim([ming maxg]) ;
    monthTitle
=['Jan';'Feb';'Mar';'Apr';'May';'Jun';'Jul';'Aug';'Sep';'Oct';'Nov';
'Dec'];
    title(monthTitle(i,:),'fontweight','bold','fontsize',10), grid
on
    if i==12
     legend ('Reference data','Original biased data','Corrected
data','Location', 'Southeast');
     strtitle =  name + " " + name2 + "\n" + "P. degree " +
num2str(degree) + ", subdivisions "+ num2str(nrpct) ;
     strtitle2 ="\n" + "Period " + num2str(min(year(D_OBS))) +" - "
+ num2str(max(year(D_OBS))) ;
     strtitle3 = strtitle + strtitle2 ;
     strtitle = compose(strtitle3) ;
     sgtitle(strtitle) ;

    end
    end

end
end
```

quantilemapping_validation.m

```matlab
% this function applies the quantilemapping correction coefficients
found
```

```matlab
% to the second dataset to test them
function [] = Quantilemapping_validation(OBS, FORECASTS, ISPREC,
COEFF, param, reservoir,graph)
%----------------------------------------------------------------
-------
% THIS IS THE VALIDATION FUNCTION: IT APPLIES THE COEFFICIENTS FOUND
IN
% CALIBRATION TO THE VALIDATION DATASET
% create string of names for the graph
if reservoir ==1
    name = "Alarcón" ;
elseif reservoir ==2
    name = "Contreras" ;
else
    name = "Bellús" ;
end
if ISPREC ==1
    name2 = "Precipitation" ;
else
    name2 = "Temperature" ;
end
% prepare observation data
D_OBS = OBS(:,1);
D_OBS = datetime(D_OBS,'ConvertFrom','datenum');
OBSdata=(OBS(:,2));
% load parameters
limit = param.limit ;
nrpct = param.nrpct ;
degree = param.degree ;
% filters
lw = param.mind ;
hi = param.maxd ;
% preallocate variables for speed
FORCDF = nan(length(FORECASTS),1) ;
for i=1:12      % for each month of the year
    OBSmont = OBSdata(month(D_OBS)==i);
    % account for measurement error: observations lower than limit
    % approximated to 0. This applies only with precipitation data
    if ISPREC == 1
    OBSmont(OBSmont<limit) = 0 ;
    end
    %remove nan and select column correspondent to month i
    FORmont = rmmissing(FORECASTS(:,i)) ;
    % set forecasts lower than the limit to 0(if data is
precipitation)
    if ISPREC == 1
    FORmont(FORmont<limit) = 0 ;
    end
    % calculate the cumulative probability
    PFOR = (1:length(FORmont))'./(length(FORmont)+1);
    %sort forecasts
    sortFOR = sort(FORmont) ;

    % find an equation for each 10th percentile
    for j = 1:nrpct
```

```matlab
    sortFORj = sortFOR(PFOR<=j*(1/nrpct) & PFOR>j*(1/nrpct)-
(1/nrpct)) ;
    % Evaluation of the polynomial curve to FOR data
    FORCDF(FORmont<=max(sortFORj)& FORmont>=min(sortFORj))=
polyval(COEFF(i,(j-1)*(degree+1)+1:(j-
1)*(degree+1)+1+degree),FORmont(FORmont<=max(sortFORj)&
FORmont>=min(sortFORj)))+FORmont(FORmont<=max(sortFORj)&
FORmont>=min(sortFORj));
    % filter for negative values (only if precipitation)
    if ISPREC == 1
        FORCDF(FORCDF<0) = 0 ;
    end
    % filter for low values (only if temperature)
    if ISPREC ~= 1
      FORCDF(FORCDF<=lw(i)) = lw(i) ;
    end
  % filter for high values
  FORCDF(FORCDF>=hi(i)) = hi(i) ;
    end
    if graph == 1
    % Comparison of cdf curves estimated for: benchmark data
(OBSmont),
    % data to modify (FORmont), corrected data (FORCDF)
     % set min and max of the graphs:
    ming = min(OBSmont)-1;
    if ISPREC == 1
        ming = 0 ;
    end
    maxg = max(OBSmont)+1 ;
    set(gcf,'position',[ 530, 190, 1111, 794])
    subplot(3,4,i)
    plot(
sort(OBSmont),(1:length(OBSmont))/(length(OBSmont)+1),'Color',0.7*[1
,1,1], 'linewidth',7)
    hold on
    plot(sort(FORmont),(1:length(FORmont))/(length(FORmont)+1), 'b-
','linewidth',4)
    % plot the corrected CDF, eliminating nan values
    plot(
sort(rmmissing(FORCDF)),(1:length(rmmissing(FORCDF)))/(length(rmmiss
ing(FORCDF))+1), 'r--', 'linewidth',2)
    xlabel('data'), ylabel('Cumulative Density Function')
    xlim([ming maxg]) ;
    monthTitle
=['Jan';'Feb';'Mar';'Apr';'May';'Jun';'Jul';'Aug';'Sep';'Oct';'Nov';
'Dec'];
    title(monthTitle(i,:),'fontweight','bold','fontsize',10), grid
on
    if i==12
    legend ('Reference data','Original biased data','Corrected
data','Location', 'Southeast');
     strtitle =  name + " " + name2 + "\n" + "P. degree " +
num2str(degree) + ", subdivisions "+ num2str(nrpct) ;
     strtitle2 ="\n" + "Period " + num2str(min(year(D_OBS))) +" - "
+ num2str(max(year(D_OBS))) ;
     strtitle3 = strtitle + strtitle2 ;
```

```
    strtitle = compose(strtitle3) ;
    sgtitle(strtitle) ;
  end
  end


end
end
```

thornthwaite.m

```matlab
% this function calculates the daily potential evapotranspiration
according to
% the Thornthwaite method
function [EVAP] = thornthwaite(daydur,temp)
% daydur input is a nx2 matrix, first column is the date (datenum
format), second day
% duration (float number). Temp has the same format. Input files
must
% contain 365 days, doesn't matter which is the first day.
% format temp data
D_temp = temp(:,1); %date of the obs
Tempdata = temp(:,2) ; %values of the temp obs
D_temp1 = datetime(D_temp,'ConvertFrom','datenum');
% sort from 1st of Jan to 31 of dec:
% get the day of year
sortday = day(D_temp1,'dayofyear') ;
% create a matrix with the dayofyear and data
matrix = [sortday Tempdata] ;
Tempdata = sortrows(matrix,1) ;
Tempdata = Tempdata(:,2) ;

% format daylight data
D_daydur = daydur(:,1); %date of the obs
Daydurdata = daydur(:,2) ; %values of the ddur obs
D_daydur1 = datetime(D_daydur,'ConvertFrom','datenum');
% get number of days per month
N = nan(12,1) ;
for i = 1:12
    N(i) = length(D_daydur1(month(D_daydur1)==i)) ;
end
% light factor calculation: 16/12*daylight length
LF = 16/12*Daydurdata ;

% month duration factor: N(i)/30
MF = N./30 ;
% repeat it for each day of each month
MFr = repelem(MF,N) ;

% calculate the yearly heat index for each year
% limits of the loop:
% preallocation of variables for speed
mtemp = nan(12,1) ;
mhindex = nan(12,1) ;
for k = 1:12
```

```matlab
    idx = month(D_temp1) == k ; %index to select data of month k &
year j
    mtemp(k) = mean(Tempdata(idx)); % mean temp of the month k and
year j
    % if monthly mean temp lower than 0, put it to 0
      if mtemp(k)< 0
         mtemp(k) = 0 ;
      end
    mhindex(k) = (mtemp(k)/5)^(1.514) ; %compute the monthly heat
index
end
yh = sum(mhindex(:)) ;
% calculation of the alpha coefficient
alpha = (6.75*10^(-7))*(yh)^3 - (7.71*10^(-5))*(yh)^2+(1.792*10^(-
2))*yh + 0.49239 ;
% CALCULATION OF THE POTENTIAL EVAPORATION
% temperature < 0 = 0: when temp is lower than 0, no evaporation
Tempdata(Tempdata<0) = 0 ;
EVAP = LF.*MFr.*(10.*Tempdata./yh).^alpha ;

EVAP = rmmissing(EVAP) ;
end
```

VBA source codes

sheet_correction.xlsm

```vba
Sub Elimina_cartella()
'
' Delete sheet Macro
' Delete sheet and change date format
'
Dim i As Integer
Dim name As String
For i = 1981 To 2015
name =
"C:\Users\avefe\Desktop\tesi\DAILY_FORECASTS\ECMWF_forecast_values_d
aily_" & i
    Workbooks.Open Filename:=name
' Delete empty sheets
    Sheets("Data_per_cell_prec").Delete
    Application.DisplayAlerts = False
    Sheets("Data_per_cell_temp").Delete
    Application.DisplayAlerts = False
' Format date in the temp sheet
    Sheets("Data_per_subbasin_temp").Activate
    Sheets("Data_per_subbasin_temp").Columns("C:C").Select
    Selection.Insert Shift:=xlToRight,
CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlToRight,
CopyOrigin:=xlFormatFromLeftOrAbove
    Sheets("Data_per_subbasin_temp").Range("C8").Select
```

```vba
    Sheets("Data_per_subbasin_temp").Range("C2587").Select
    ActiveCell.FormulaR1C1 = "x"
    Sheets("Data_per_subbasin_temp").Range("C8").Select
    Sheets("Data_per_subbasin_temp").Range(Selection, _
Selection.End(xlDown)).Select
    Selection.FormulaR1C1 = "=INT(RC[-1])"
    Selection.Copy
    Sheets("Data_per_subbasin_temp").Range("D8").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, _
SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Sheets("Data_per_subbasin_temp").Range("B4:B7").Select
    Selection.Copy
    Sheets("Data_per_subbasin_temp").Range("D4").Select
    ActiveSheet.Paste
    Sheets("Data_per_subbasin_temp").Columns("B:C").Select
    Application.CutCopyMode = False
    Selection.Delete Shift:=xlToLeft
    Sheets("Data_per_subbasin_temp").Range("B8").Select
    Sheets("Data_per_subbasin_temp").Range(Selection, _
Selection.End(xlDown)).Select
    Selection.NumberFormat = "m/d/yyyy"


' format date in the prec sheet
    Sheets("Data_per_subbasin_prec").Activate
    Sheets("Data_per_subbasin_prec").Columns("C:C").Select
    Selection.Insert Shift:=xlToRight, _
CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlToRight, _
CopyOrigin:=xlFormatFromLeftOrAbove
    Sheets("Data_per_subbasin_prec").Range("C8").Select
    Sheets("Data_per_subbasin_prec").Range("C2587").Select
    ActiveCell.FormulaR1C1 = "x"
    Sheets("Data_per_subbasin_prec").Range("C8").Select
    Sheets("Data_per_subbasin_prec").Range(Selection, _
Selection.End(xlDown)).Select
    Selection.FormulaR1C1 = "=INT(RC[-1])"
    Selection.Copy
    Sheets("Data_per_subbasin_prec").Range("D8").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, _
SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Sheets("Data_per_subbasin_prec").Range("B4:B7").Select
    Selection.Copy
    Sheets("Data_per_subbasin_prec").Range("D4").Select
    ActiveSheet.Paste
    Sheets("Data_per_subbasin_prec").Columns("B:C").Select
    Application.CutCopyMode = False
    Selection.Delete Shift:=xlToLeft
    Sheets("Data_per_subbasin_prec").Range("B8").Select
    Sheets("Data_per_subbasin_prec").Range(Selection, _
Selection.End(xlDown)).Select
    Selection.NumberFormat = "m/d/yyyy"
```

```
' save and close file
    ActiveWorkbook.Save
    ActiveWorkbook.Close
    Next i
End Sub
```