Relating groundwater heads to stream discharge by using machine learning techniques

A case study in subcatchment Chaamse Beken

V. Demetriades

TU Delft

Waterschap Brabantse Delta

Relating groundwater heads to stream discharge by using machine learning techniques

A case study in subcatchment Chaamse Beken

by

V. Demetriades

in partial fulfillment of the requirements for the degree of

Master of Science
in Civil Engineering, Watermanagement

at the Delft University of Technology,
to be defended publicly on Friday January 24th, 2020 at 11:00 AM.

Chairman:	Prof. dr. ir. M. Bakker,	TU Delft
Thesis committee:	Dr. M. Hrachowitz,	TU Delft
	Dr. R. Taormina	TU Delft
	Ir. T. Deurloo,	Waterschap Brabantse Delta
	Dr. K. Vink,	Waterschap Brabantste Delta

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Programming codes in Python are available at https://github.com/valdemetriades/msc-thesis.

TU Delft Delft University of Technology

"A baby learns to crawl, walk and then run. We are in the crawling stage when it comes to applying machine learning."
*Dave **Waters***

Acknowledgements

This thesis represents the final product in becoming a Master of Science in Watermanagement at the Delft University of Technology - faculty of Civil Engineering and Geosciences. I would like to acknowledge all the people who contributed to my graduation. First of all, I would like to thank the entire committee for guiding me throughout my thesis. In addition, I would like to give a special word of thanks to Waterschap Brabantse Delta - Kees Vink and Thomas Deurloo - for the daily supervision. Introducing me to the field of machine learning has been an amazing step in my future career. I am already looking forward to apply machine learning even more in the field of hydrology.

V. Demetriades
Delft, January 2020

Abstract

Waterschap Brabantse Delta (WBD) has the intention to implement measures that enhance the **baseflow**. Baseflow consists of the groundwater flow and a small part of the interflow. During dry periods, streams are dependent on the baseflow. Enhancing the baseflow has a proper effect on the **ecologically** relevant quality of waters, and is therefore wanted for WBD according to the regulations of the Water Framework Directive.

It is needed to quantitatively examine these measures that have been implemented in subcatchments of WBD. Therefore, sufficient data of good quality is needed. Especially, the **stream discharge** itself is important to collect. However, stream discharges are scarce datasets: it can not be measured directly, but needs to be derived from other parameters. In-situ devices (for example acoustic Doppler velocimeters and acoustic Doppler current profilers) measuring water depth and velocity, can be used to derive the stream discharge by applying the area-velocity method. However, these devices are expensive, time-consuming, and in-situ measurements are needed.

Therefore, other methods/models for obtaining stream discharge have been derived. These can be divided into two categories: **physically based models** and empirical methods or **data-driven models**. An example of a physically based model are conceptual rainfall-runoff models making use of the physical understanding behind the hydrological system. These rainfall-runoff models require subcatchment characteristics, which are hard to determine and hence often model calibration is applied. Therefore, also empirical methods have been used to derive the stream discharge. A well-known example is the rating-curve, making use of Q, h-relations. This method has as a downside that Q, h-relations are dependent on hydraulic parameters of the sides of the streams. These parameters are hard to determine, variable over time and time consuming to measure. Therefore, for this research the question arises if there is another data-driven model that can find a link between stream discharge and another variable within the subcatchment, which is not difficult to determine or obtain.

Data-driven models in the form of **machine learning** have been applied before for stream discharge prediction. In these models the stream discharge is predicted based on historical stream discharge data. These models can only be used on the short-term and are therefore not wanted for this research.

Groundwater heads are mostly not scarce datasets and it is expected that these groundwater heads play a large role within the production of the baseflow. Moreover, large historical datasets exist of groundwater heads. In this research, it is examined if the stream discharge can be simulated based on groundwater head time series by using machine learning algorithms. Four different machine learning algorithms are used: **decision tree regression (DTR), random forest regression (RFR), gradient boosting regression (GBR) and support vector regression (SVR)**. The models are applied to the subcatchment Chaamse Beken, for which it is wanted to simulate stream discharge between 2003-2019 (flow measuring weir has been removed in 2003), and possibly in the future. The training set of these models is set from 1985-1999, whereas the test set is from 1999-2003. Moreover, different input variables and combinations of these variables are chosen for the models: shallow wells (screen-1 wells), deeper wells (screen-2 wells), precipitation and potential evaporation. The model performance is evaluated with the metrics Nash-Sutcliffe Efficiency (NSE), mean absolute error (MAE), fourth root mean quadrupled error (R4MS4E) and mean squared logarithmic error (MSLE). The first two are considered for overall model performance, whereas the latter two are for high flow and low flow model performance.

The results of the machine learning models show that using one screen-1 well (with the highest correlation with other screen-1 wells in Chaamse Beken) did not succeed in simulating the stream discharge, for all four algorithms. Using all selected screen-1 wells results in reasonable values for simulating the stream discharge (NSE of 0.69-0.70), except for algorithm DTR (NSE of 0.42). Deeper wells (screen-2 wells) do not improve the relation between groundwater heads and stream discharge. Furthermore, adding variables precipitation and potential evaporation to the groundwater heads of screen-1 wells improve the overall model performance for SVR. Also, for the machine learning algorithm SVR the low flow model performance seems to be promising. A NSE of 0.75 and a MSLE of 0.017 $\ln^2(m^3/sec)$ can be reached for the SVR model by using groundwater heads of screen-1 wells, potential evaporation and precipitation as inputs. However, adding precipitation did not result in a better simulation of the peak flows (R4MS4E of 0.60 m^3/sec).

In order to examine if this machine learning model can be used in the future for stream discharge simulation, the SVR model is compared with an existing conceptual hydrological model: **GR4J**. The GR4J model is used as a baseline model in this research, as it can simulate reasonable stream discharge values by only using precipitation and potential evaporation as inputs. For the GR4J model the training set is set as calibration period and the test set as validation period. The model is calibrated with the objective function MAE. The GR4J model has a NSE value of 0.80 and a MSLE value of 0.0084 $\ln^2(m^3/sec)$ and can be rated as good. It has a larger NSE value than the SVR model and performs better than the SVR machine learning model.

It is important to stress that for the GR4J model the *memory (or state) of the system* is included. In other words, based on the previous day the water quantity in the the water reservoirs of the GR4J model is updated for the next day. This

inclusion of the memory of the system is not the case for the machine learning algorithms. Also, by manually adding the memory of the system, the machine learning algorithms did not or barely improve. Furthermore, an important difference is the fact that *groundwater heads* play a significant role in the simulation of the stream discharge by using machine learning algorithms. These groundwater heads are not directly used in the GR4J model. Lastly, it is stressed that for building the GR4J model *physical understanding of the hydrological system* is needed, whereas for machine learning this is not the case.

For further research it is recommended to first divide the discharge time series into a baseflow time series and peakflow time series (for example by using hydrograph separation). Separate machine learning algorithms can be applied for the peakflow, and for the baseflow. This may lead to better results for the baseflow, and for the peakflow itself, whereafter they can be combined to full discharge time series. Another recommendation for future research is to use a logarithmic transformation of the stream discharge, to make the relation between stream discharge and groundwater heads more linear and less skewed. In addition, also groundwater heads of wells outside the Chaamse Beken can be used as inputs for the machine learning models. In this research, only wells within the Chaamse Beken were used. Lastly, in this research, significant time was spent to validate the groundwater head time series (by using Pastas Time Series Analysis). However, the discharge time series is only checked on correlations with other discharge time series in the surroundings. In future research, other techniques for validating stream discharges need to be developed and applied.

Overall, it can be concluded that GR4J is favoured above the machine learning model SVR. However, machine learning techniques (especially SVR) show promising results for the future in simulating stream discharge by using groundwater heads and without taking the state of the hydrological system into account.

Contents

1 Introduction

Waterschap Brabantse Delta (WBD) has the obligation to implement the Water Framework Directive (WFD) as have all the other water authorities in Europe. According to the WFD, in 2027 all designated waters in Europe must have a good chemically and ecologically relevant quality (Ligtvoet et al., 2008). Therefore, Waterschap Brabantse Delta has assigned 25 WFD subcatchments to better define the status of each single subcatchment (Waajen et al., 2018). Some of these subcatchments have been suffering from severe droughts in recent periods (WBD, 2018). As a result, the discharge of the main streams within these subcatchments is decreasing and parts of the streams are ending up dry (Vink, 2019). Taking climate change into account, the dry periods may even increase with adverse consequences for the streams in the WFD subcatchments such as the ecological quality of these streams. In general, discharges in streams can be classified into three components: *overland flow* produced by water that does not infiltrate into the soil and travels quickly to the stream, *interflow* consisting of water that infiltrates into the soil and travels laterally downslope through upper soil layers, and *groundwater flow* that infiltrates and travels through the aquifer (Bosch et al., 2017). Interflow moves more slowly than overland flow but typically more rapidly than groundwater. Interflow can be further distinguished in a quickflow portion and a portion moving slowly through the subsoil. During dry periods, discharges in streams consist of the slow portion of interflow and groundwater flow as overland flow due to precipitation is absent. In hydrological terms, the slow portion of the interflow together with the groundwater flow is often named *baseflow* (Bosch et al., 2017). The baseflow is dependent on the groundwater reservoir which is being replenished by infiltration of precipitation (Zhang and Schilling, 2006). Enhancing the base flow of streams in the WFD subcatchments is expected to lead to a better ecological quality of these streams. Therefore, Waterschap Brabantse Delta has the intention to implement measures that enhance the base flow. Choosing the right measures requires a proper understanding of the water-groundwater relations and sufficient data of good quality.

1.1 Problem Statement

Sufficient data of good quality within a subcatchment is still an issue for small streams in WBD and among other places. Stream discharges in particular are scarce datasets. This is due to the fact that stream discharge itself can not be measured directly, but needs to be derived from other parameters, such as the cross-sectional area and velocity (Luxemburg and Coenders, 2017). Since both the velocity and water depth vary over a cross section, it is difficult to derive the discharge from one cross section over a stream directly. Therefore, a common method to derive discharge of the total stream is the area-velocity method by using current meters, which is time-consuming. Other discharge measurement devices making use of this method are acoustic Doppler velocimeters and acoustic Doppler current profilers. Deriving the discharge with these devices is less time consuming, but expensive. Furthermore, these devices can only be used at specific locations (Muste et al., 2007). Moreover, there is no single measurement device that can continuously measure discharge directly. It can be concluded that measuring stream discharge accurately, efficiently and cheaply is still an unsolved problem. Without being able to measure the stream discharge in a stream of a single subcatchment, it is impossible to examine the hydrological impacts of the - ecology improving - measures being taken in that same subcatchment. These hydrological impacts can only be examined by physically looking at the stream itself. It will be more sophisticated to quantitatively examine these measures. In other words, the stream discharge itself needs to be known.

1.2 Literature background and knowledge gaps

A well known method in literature to more efficiently obtain the stream discharge data is by applying conceptual hydrological models, or more simply known as rainfall-runoff models (Sitterson et al., 2017). Conceptual models are **physically based models** and they interpret runoff processes by connecting simplified components in the overall hydrological process, while making use of the water balance equation. In these models the discharge is derived from the inputs precipitation, potential evaporation and subcatchment characteristics such as slopes and land cover (Sitterson et al., 2017). These models have a downside that these subcatchment characteristics are hard to determine.

In addition to the physically based models, literature shows that there are some **empirical methods** or **data-driven models** (DDM's) to obtain stream discharge data (Solomatine and Ostfeld, 2008). An example is using machine learning algorithms as DDM's for predicting the stream discharge a number of days ahead (Adnan et al., 2019). However, within this research the stream discharge of the previous days is used as an input parameter. These models can be used for obtaining stream discharge data on the short-term.

Another example of an empirical method is obtaining stream discharge (Q) from water stages (H) (Luxemburg and Coenders, 2017). Stages can be measured directly in streams and can be measured continuously, whereas discharge can not be measured directly and continuously. Relating stream stages to stream discharge can therefore be a solution to obtain continuous stream discharge data. An example of a Q, H-relation is the *rating curve*. Each side of a stream has a specific Q, H-relation dependent on different hydraulic parameters among other bottom slope, bottom width and bed roughness. Since these parameters are hard and time consuming to determine for each specific stream and vary along a stream, the rating curve method is still not an ideal method to gather stream discharge data. Again as a downside, it is hard

to determine these subcatchment characteristics. It would be more beneficial to find a link between stream discharge (Q) and another variable within the hydrological subcatchment which is not difficult to determine or to obtain.

1.3 Strategy within this research

Since it is known that stream discharge also largely relates on the base flow and hence the corresponding groundwater reservoir, the main focus within this research is on relating groundwater heads (h) from different wells to stream discharge (Q) in the same subcatchment. Groundwater head datasets are being measured in monitoring wells (see Appendix A) and are in contrast to discharge data and subcatchments characteristics not scarce at all. However, the quality of these datasets can be poor and needs to be validated before a possible relation with stream discharge can be found. After having validated the groundwater head datasets, the relation between these groundwater heads and stream discharge is being approached from a **data-driven** point of view (without using the physical processes of hydrology (Solomatine and Ostfeld, 2008)), in contrast to the already known conceptual hydrological models found in literature. Different **machine learning algorithms** are applied as data-driven models.

Since there is a certain scepticism about DDM among many hydrologists and water resources specialists (because of the fact that the physical understanding of the hydrological system does not play a role within DDM's) (Solomatine and Ostfeld, 2008), the outcomes of the machine learning algorithms are compared with a traditional conceptual hydrological model. Note that the DDM's and the calibrated conceptual model will be specific for each single subcatchment and can not be used for other subcatchments, whereas the research method proposed here is applicable for each single subcatchment in general.

The focus area of this research (or strategy) is visualised in a cross-section of a single subcatchment in Figure 1.

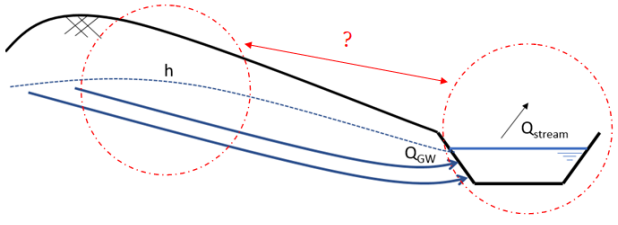


Figure 1. The strategy within this research by obtaining the stream discharge Q from the groundwater heads h

1.4 Research Questions

In this research three main research questions are examined. The first research question focuses on the relation between groundwater heads (h) and stream discharge (Q) within a single subcatchment, by using **DDM's**. Multiple machine learning algorithms are used for these DDM's. Furthermore, different input variables sets are used for these models, such as shallow wells (screen-1 wells), deeper wells (screen-2 wells), precipitation and potential evaporation. This is done to examine which input variables give the best DDM output. In research question two, the DDM with the best output (depending on the machine learning algorithm and the input variables) is chosen.

This chosen DDM is compared to an already existing rainfall-runoff model in the third research question, which is as mentioned before a **physical model**. This model is used as a baseline to compare with the outputs of the best DDM. The research questions are elaborated below:

1. Can we analyze if there is/are (a) relation(s) between groundwater heads (X) and stream discharge (Y) in the same subcatchment by using machine learning algorithms, such that we can simulate the stream discharge time series from these groundwater heads in the future, or to fill in gaps in the historical stream discharge time series? In formula form with $n+1$ the number of wells:

$$Y(\vec{X}) = F(X_0, X_1, ...X_n)? \quad (1)$$

   (a) Is it possible to find this relation when only using one screen-1 well? Or do we need all screen-1 wells within the subcatchment in order to find the relation with the stream discharge?

   (b) Can the relation be found with only screen-1 wells, or are screen-1 and screen-2 wells needed to find an accurate relation with the stream discharge?

   (c) Is it necessary to add other hydrological variables such as precipitation (P) & potential evaporation (Ep), in order to find a relation with the stream discharge?

   (d) Is it necessary to introduce memory (M) and delays (Δt) to the screen-1 wells, precipitation and potential evaporation variables, in order to find a relation with the stream discharge?

2. Which DDM is the most suitable for finding a relation with the stream discharge?

3. Does the best performing DDM perform as well or better than a conceptual hydrological model?

1.5 Outline

In the next chapter "Materials", first of all the chosen subcatchment is described in section 2.1, including the selection criteria for a subcatchment for this research. In the second section, the data collection and validation of the stream discharge series is elaborated, followed by the same procedure

for the groundwater head time series in section 2.3. In addition, a data analysis is performed for the different groundwater head time series. In this subsection, the data is being described for the DDM's.

The first part of chapter three "Methods", focuses on machine learning: what is machine learning, the principle of machine learning within this research (including different assessments to tackle the research questions), which machine learning algorithms are chosen (based on literature) and a thoroughly explanation of how these algorithms work. The second part focuses on the conceptual hydrological model: decision of the chosen model based on literature and the working principle of this model. The last part of "Methods" describes the evaluation metrics to compare the model performances.

Chapter four "Results, Discussion & Limitations" covers the comparison of the different machine learning models and evaluates the performance of these models based on their output. Furthermore, the output of the conceptual hydrological model is added to the results as a baseline and is compared with the machine learning algorithms outputs. In addition to these results, the limitations of this research are elaborated.

In chapter five "Recommendations and future research" some ideas and recommendations for further research are more elaborated. Finally, in chapter six "Conclusions" the goal of this research is emphasized and the research questions are answered.

2 Materials

The materials needed for this research are: a study area, groundwater head time series and a discharge time series in that same study area. In this chapter the study area is chosen and elaborated on, followed by the data collection, validation and analysis of the stream discharge time series. Lastly, the groundwater wells usable for this research are illustrated, whose groundwater head time series are validated and analysed.

2.1 Study area

The purpose of this research is to obtain the discharge time series from groundwater head time series in the same subcatchment, such that ecology improving measures can be quantitatively analysed for subcatchments where a part of the discharge time series is unknown. In general, this relation can be used for every subcatchment to fill in gaps within the discharge time series or for interpolation. Moreover, the relation can be used as a validation tool when the discharge time series is already known. In order to find the data-driven relation, the time series of the groundwater heads and stream discharge itself must be large enough for machine learning purposes (Raschka and Mirjalli, 2017). This is defined as the first criteria for selecting a proper study area. The next criteria is the fact that there should be free flow in the subcatchment, without the presence of weirs inside the streams. This is only necessary for the conceptual hydrological model, but is not expected to be a problem for the DDM's. Moreover, the ideal scenario is to have a stream discharge gauge at the outlet point of a subcatchment, as this is also the point where the stream discharge is simulated for a rainfall-runoff model. To summarize the criteria:

- large enough historical dataset of the stream discharge and groundwater head time series
- free flow or absence of weirs in the streams of the subcatchment
- preferably a stream discharge measurement gauge at the outlet point of the subcatchment

WBD has the supervision of in total 25 WFD subcatchments which together form 6 catchments: Aa of Weerijs, Bovenmark, Mark-Vliet, Dongestroom, Hollandsch Diep/Amer and Brabantse Wal. The Bovenmark has in total 6 subcatchments, of which the streams in Strijbeekse Beek and Chaamse Beken have been identified as streams where base flow has to increase for ecological purposes and hence multiple measures have been taken. Unfortunately, a lot of data in the discharge time series is missing and makes it difficult to quantitatively examine if the base flow is increasing. For this research, Chaamse Beken is chosen as study area, since the discharge measurement gauge, Chaamse Beken Ulvenhout (CBU), is almost at its outlet point, the presence of free flow and the fact that the historical dataset of CBU and groundwater head time series are large enough (18 years) for the DDM. The location of the subcatchment Chaamse Beken within WBD is depicted in Figure 2, and in Figure 3 the subcatchment itself is given with its stream discharge measurement gauge CBU and the streams.

The subcatchment has a surface area of almost 50 km^2 (Broekhoven et al., 2019), from which the land use is mostly forest and grassland, as can be seen in Figure 4.

Furthermore with respect to the surface level, there is a descending slope from the southeast towards the stream discharge point CBU in the northwestern part, as can be seen in Figure 5a. The highest surface level is around 25 meters NAP, while the lowest level is around 2 meters NAP. The soil within this subcatchment is made of phreatic aquifers. In addition, there is one shallow clay layer. This clay layer is been deposited by the Stamproy Formation (Broekhoven et al., 2019): a formation including deposits that are formed by northward flowing rivers that drained the central and northern part of Belgium (Westerhoff et al., 2009). Note that this clay layer is not present within the whole subcatchment (almost none in the north western part), but at most parts it is present. Especially in the southern part of the Chaamse Beken, the clay layer can reach thickness levels of 7meters. The top-, bottom level (m NAP) and the thickness (m) of this clay layer are depicted in respectively Figures 5b to 5d. The location of this clay layer can also be seen in these Figures.

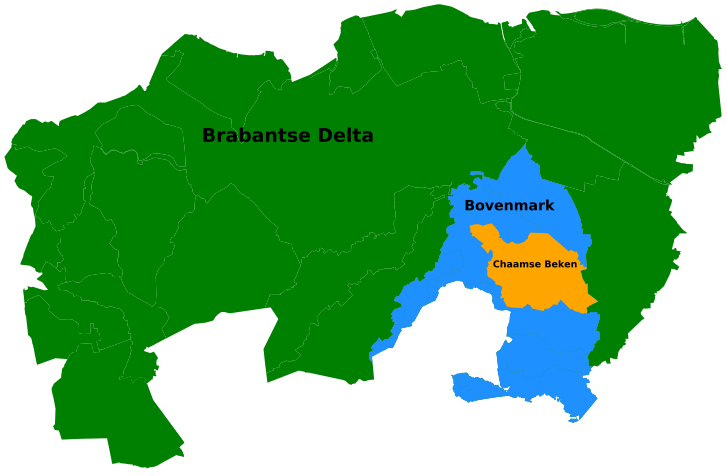


Figure 2. WBD with the location of catchment Bovenmark and subcatchment Chaamse Beken

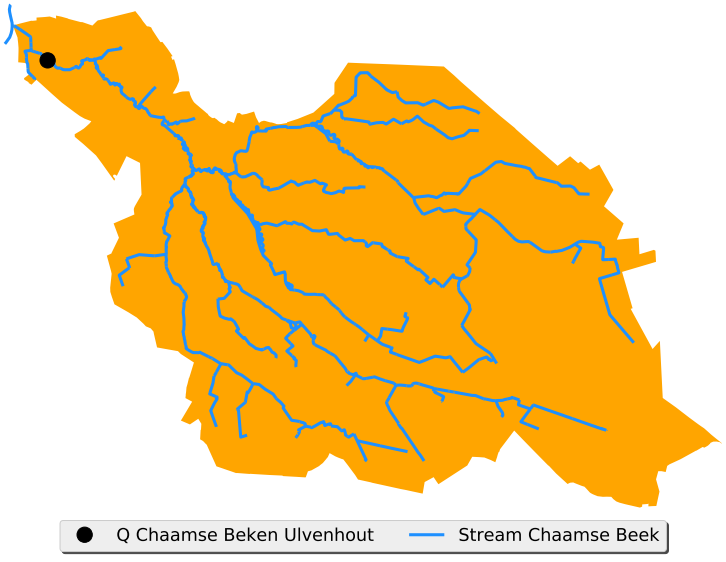


Figure 3. The study area of this research: subcatchment Chaamse Beken with its discharge stream point and its streams

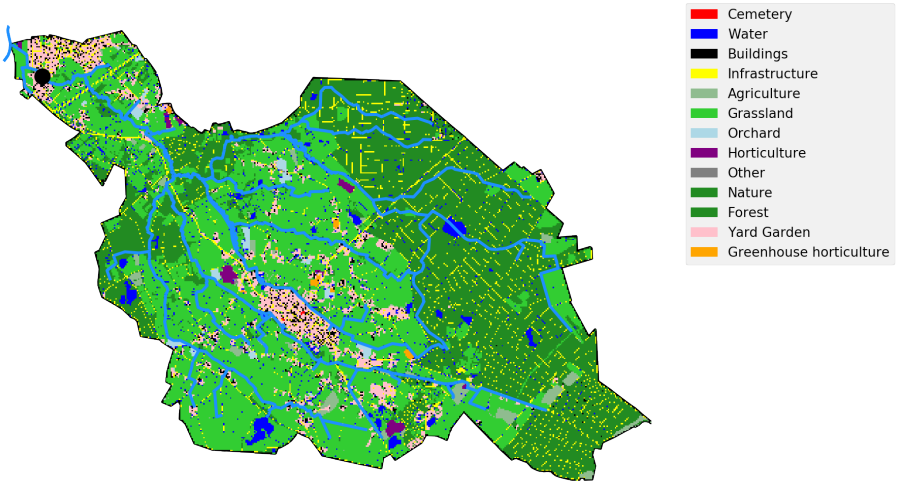
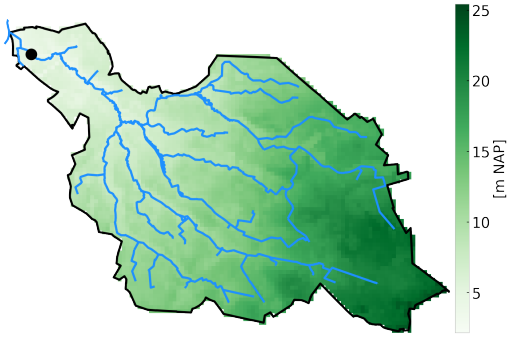
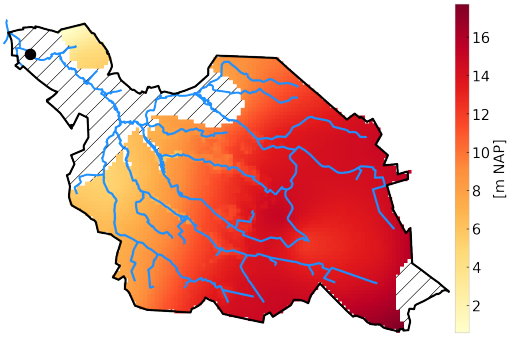


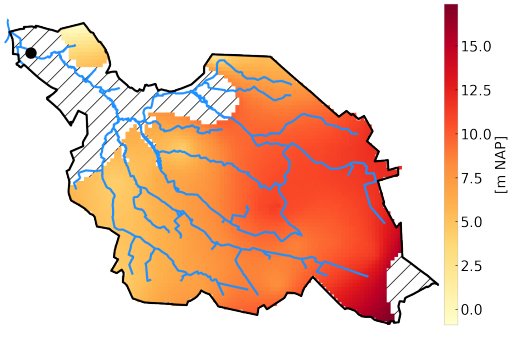
Figure 4. Land use within the Chaamse Beken, showing mostly grassland, nature and forest



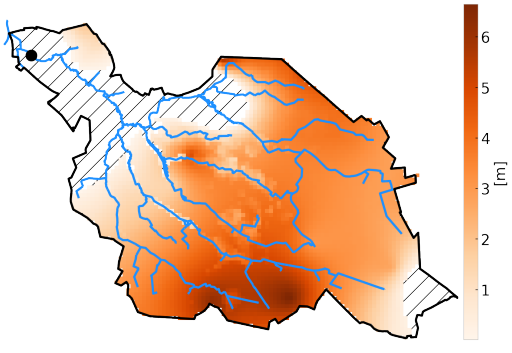
(a) Surface level Chaamse Beken [m NAP]



(b) Top level of Stamproy clay layer [m NAP]



(c) Bottom level of Stamproy clay layer [m NAP]



(d) Thickness of Stamproy clay layer [m]

Figure 5. Surface Level & geo-information of the Stamproy clay layer within the subcatchment Chaamse Beken

2.2 Stream Discharge Time Series CBU

2.2.1 Data collection & analysis

As stated before, the stream discharge gauge CBU can be seen as the outlet point of the subcatchment and it has a long historical daily dataset from 1984 to 2003. The stream discharge is derived from a certain Q,h-relation from a flow measuring weir placed in 1984 (Vink, 2019). This flow measuring weir is able to continuously measure the upstream water level and taken into account the physical properties of the standardized weir, the water level can be converted to a stream discharge. Flow measuring weirs are mostly seen in small rivers or streams for the purpose of measuring stream discharge. In the last years a lot of flow measuring weirs have been removed (Hartong and Termes, 2009), which is also the case for the weir CBU in 2003. The most important reason for this is the fact that fish ladders can not be combined with the weirs, which is desirable for ecological benefits. Moreover, some weirs are removed as to reinforce natural flow conditions within the subcatchment (Hartong and Termes, 2009).

The stream discharge series of CBU is depicted in Figure 6 and is calculated from the water level in the unit of m^3/sec with a daily average frequency. The time series is available from 03-05-1984 to 31-10-2003, however, in this research for simplicity reasons the time series is used from 1985 to 2003 to have only full years (sea black lines in Figure 6). The average stream discharge is 0.35 m^3/sec, while the minimum and maximum value are respectively 0.0 and 4.69 m^3/sec. The time series shows that during some periods the stream runs almost dry. Moreover, some high peaks are present in the time series as a result of heavy precipitation. To have more insight in the time series of the years itself, a single year comparison is shown in Figure 7.

This year comparison shows wet and dry years within the time period of 1985-2003. The year 1996 is a very dry year with an average stream discharge of only 0.11 m^3/sec. This year is also in the record books of extremely dry years within the Netherlands due to a large precipitation deficit (Beersma et al., 2004). Furthermore, the years 1988 and 2001 are wet years with respect to the other years in the time period.

2.2.2 Data validation

In order to find a relation between groundwater heads and the stream discharge CBU, it is important to first validate the data. Validation of stream discharge data is still very complex and possible methods are outlier detection and a correlation analysis with other surrounding stream discharge series (according to Witteveen & Bos, personal communication, 2019). For this research, the last method is applied.

The locations of the chosen stream discharge points are shown in Figure 8 and the start- & end measuring period, the mean discharge and the corresponding subcatchment & catchment are depicted in Table 1. Note that the stream discharge point Merkske Castelré is connected to the stream upstream of Galderse Beek Galder via a stream in northern direction. In the correlation analysis, each time series is compared with the time series of CBU taken into account only the overlapping time period. As can be seen in Table 2, all the surrounding discharge time series have a positive correlation coefficient of 0.86 or higher with CBU. According to this correlation analysis, the discharge time series of CBU is assumed to be validated for this research. In additional research, outlier detection can be used to have more certainty about the validated discharge time series.

2.3 Groundwater head time series within Chaamse Beken

The next step is to select the groundwater wells within the Chaamse Beken, and to validate the groundwater head time series of these wells. Furthermore, an analysis is performed on these time series to gain more insight into the groundwater system of the Chaamse Beken.

2.3.1 Data selection groundwater wells

The source Dinoloket (the main source in the Netherlands for obtaining subsurface data) is used to collect groundwater head time series of monitoring wells. The data selection process consists of multiple steps, from which each step will be elaborated below. The selection process of screen-1 wells and screen-2 wells is also visualized respectively in Figure 9 and 10.

Step 1: Shapefile in Dinoloket
The first step is to draw a large shapefile around Chaamse Beken in Dinoloket and the data of these wells within the Chaamse Beken is collected. The Dinoloket shapefile contains 962 different groundwater head time series, from which some are part of a single well screen series. The screens of these wells are placed at approximately the same location, but in different aquifers. For example, screen-1 is placed in the phreatic aquifer, screen-2 is placed in a deeper aquifer, etcetera, but they have the same screen name. Most of the wells have around 2 to 6 screens, from which the 6th screen can reach depths of almost 100 meters NAP. The purpose of placing screens in different aquifers is to examine the phenomena seepage and filtration (Ritzema et al., 2012). The phenomena of seepage and infiltration are visualised in Appendix A. In this research, it is firstly assumed that only screen-1 & screen-2 wells are responsible for groundwater flow towards the stream Chaamse Beken, whereas deeper layer screens will bypass this stream. Furthermore as a last criteria in this step, the groundwater head time series should have a minimum number of 100 measurements. Groundwater head time series with a sample size of less than 100 are removed. To conclude, there are 210 screen-1 wells

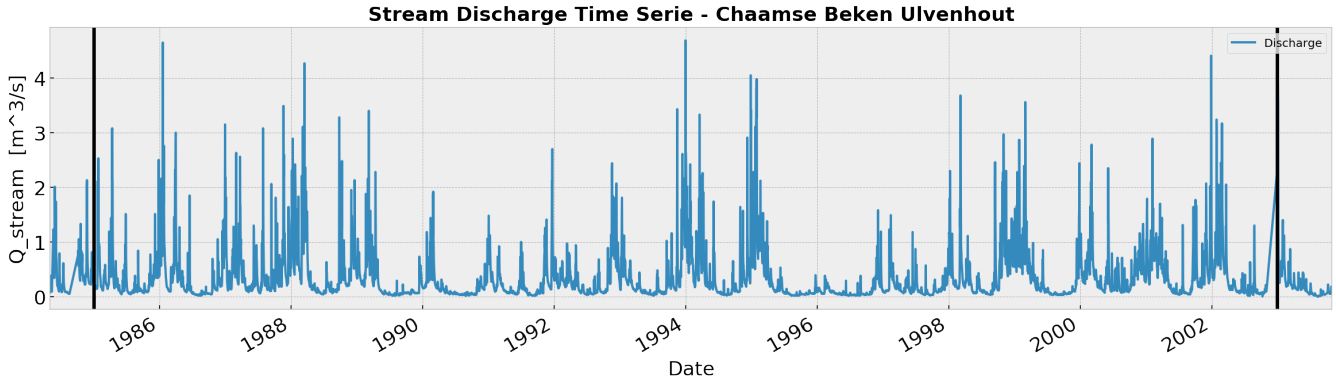


Figure 6. Stream Discharge time serie Chaamse Beken Ulvenhout (CBU). The black lines are representing the period from 1985 to 2003 to be used for this research.

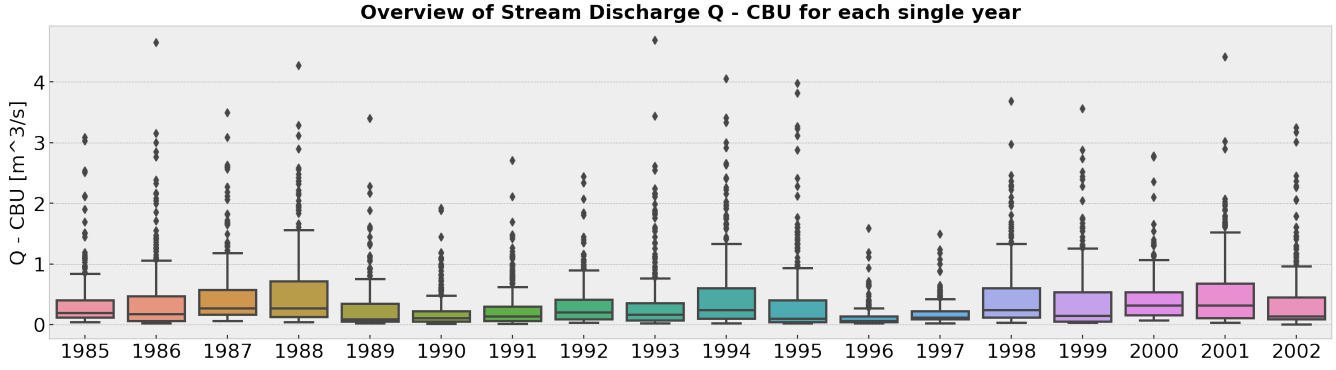


Figure 7. Year-Comparison Stream Discharge Q - CBU

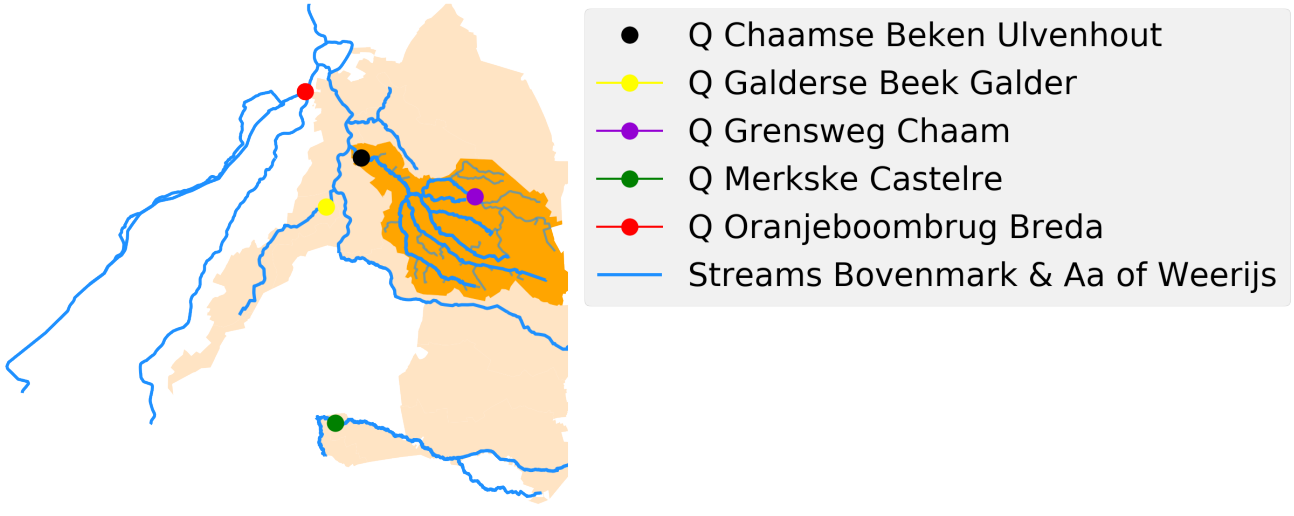


Figure 8. Locations of the other stream discharge points for the correlation-analysis

Stream Discharge point	Start	End	Mean Discharge [m^3/sec]	KRW subcatchment	KRW catchment
Q Chaamse Beken Ulvenhout	1985	2002	0.35	Chaamse Beken	Bovenmark
Q Galderse Beek Galder	1988	2003	0.13	Galderse Beek	Bovenmark
Q Grensweg Chaam	1992	1994	0.035	Chaamse Beken	Bovenmark
Q Merkske Castelré	1987	2019	0.58	Merkske	Bovenmark
Q Oranjeboombrug Breda	1982	2019	2.82	AA en Weerijs	Aa of Weerijs

Table 1. Stream Discharge points for the correlation analysis

Stream Discharge point	Correlation with CBU
Q Chaamse Beken Ulvenhout	1
Q Galderse Beek Galder	0.92
Q Grensweg Chaam	0.87
Q Merkske Castelré	0.94
Q Oranjeboombrug Breda	0.94

Table 2. Correlation of CBU and the stream discharge points in the surroundings

within the shapefile of Dinoloket with a minimum number of 100 measurements (Figure 9a), and 59 screen-2 wells (Figure 10a).

Step 2: Wells within Chaamse Beken
The next step is to eliminate the screen-1 & screen-2 wells from the shapefile that are not within the Chaamse Beken. From the 210 screen-1 wells, 73 are remaining within the Chaamse Beken (Figure 9b). From the 59 screen-2 wells, only 19 are remaining within the Chaamse Beken (Figure 10b). For this research, it was decided to take only the screen wells within the Chaamse Beken into account. Note that the boundaries of the subcatchment Chaamse Beken do not also have to be the boundaries of the groundwater system itself. Therefore, in future researches the collection of screen wells can be extended to outside the subcatchment itself.

Step 3: Define necessary time period for the model
The selection of wells in step two is further refined so that only usable screen series for this research are remaining. Since the goal of this research is to link groundwater head time series with the stream discharge series of CBU, the chosen time period of the groundwater head time series is dependent on the available time period of the stream discharge CBU (1985-2002). Therefore, monitoring wells with at least a data availability for the years 1985-2002 are selected to apply for the data-driven model. From this criteria, 13 screen-1 wells (Figure 9c) & 8 screen-2 wells (Figure 10c) are left within the Chaamse Beken.

Step 4: Wells from step 3 & measuring up to 2019
Wells that have been measuring groundwater heads after 2002 and are still measuring today are preferable, such that based on these groundwater head series the data-driven model can be used to get the corresponding discharge time series until today. Therefore, this criteria is defined as selecting wells that are still in place and are measuring up to 2019 and hence in 2019 (assumed as still measuring today). This last step results in 8 screen-1 wells (Figure 9d)& 6 screen-2 wells (Figure 10d), which are the following:

- Screen-1 wells: B50B0074_1, B50B0075_1, B50B0101_1, B50B0216_1, B50B0374_1, B50B0380_1, B50B0498_1, B50E0140_1
- Screen-2 wells: B50B0074_2, B50B0075_2, B50B0101_2, B50B0216_2, B500380_2, B50E0140_2

Note that there are two screen-1 wells which are not within the screen-2 wells selection: well B50B0374 at almost the same place at B50B0380, and well B50B0498 at almost the same place at B50B0075. Since these two wells are having similar time series as their neighbour wells, they do not add new information to this research and are therefore not considered. With this given, there are 6 screen-1 and 6 screen-2 wells from the same monitoring well and can therefore also be more easily compared for answering research question 1b. In the end, the following screen-wells are used in this research:
Screen-1 wells:

- $X1_0$: B50B0074_1
- $X1_1$: B50B0075_1
- $X1_2$: B50B0101_1
- $X1_3$: B50B0216_1
- $X1_4$: B50B0380_1
- $X1_5$: B50E0140_1

Figure 9. Selection Process of the screen-1 wells: Step 1 (green), step 2 (light green), step 3 (yellow), step 4 (red)

.

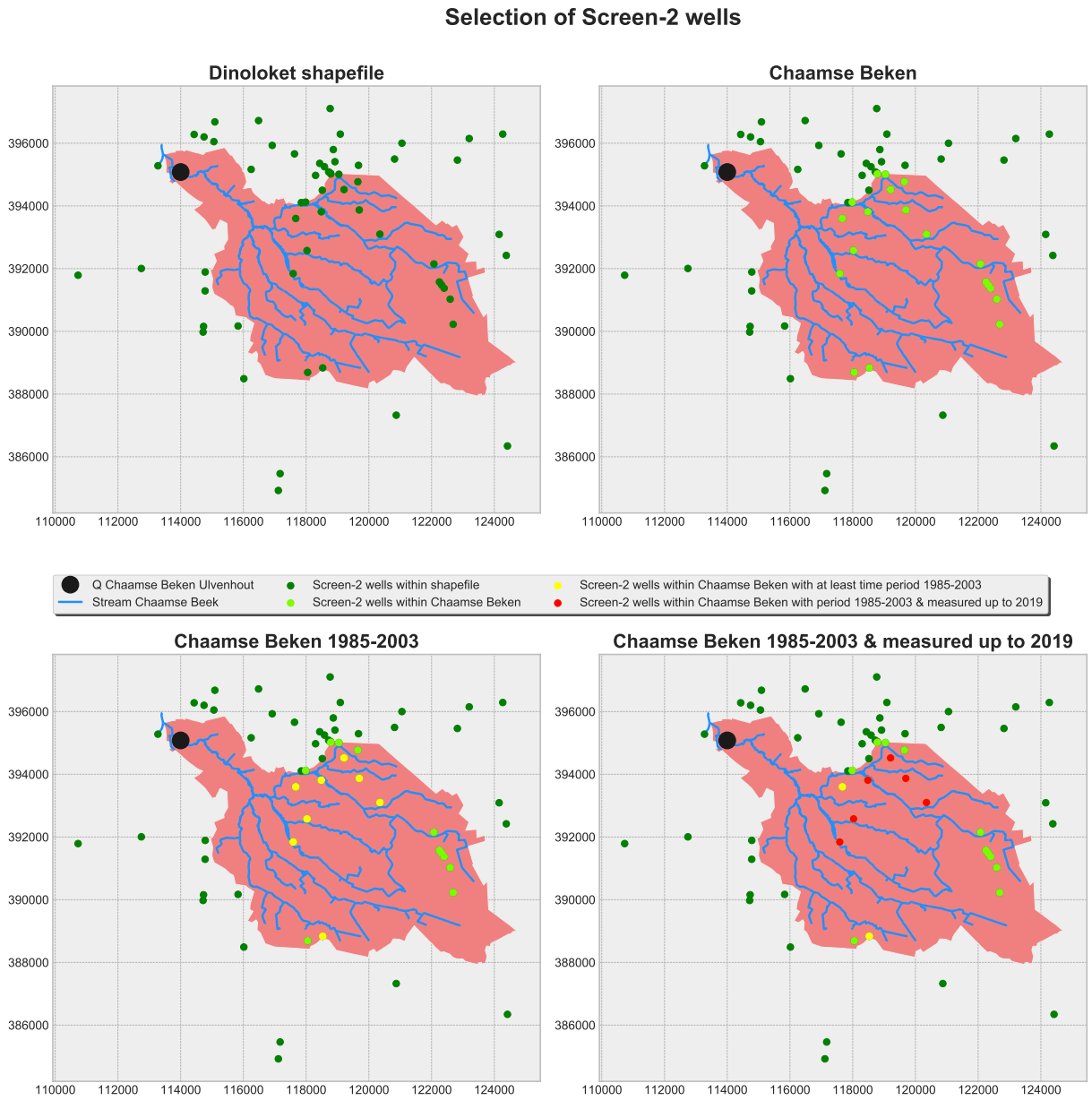


Figure 10. Selection Process of the screen-2 wells: Step 1 (green), step 2 (light green), step 3 (yellow), step 4 (red)

Screen-2 wells:

- $X2_0$: B50B0074_2
- $X2_1$: B50B0075_2
- $X2_2$: B50B0101_2
- $X2_3$: B50B0216_2
- $X2_4$: B500380_2
- $X2_5$: B50E0140_2

The locations of the screen-1 ($X1_0$-$X1_5$) and screen-2 wells ($X2_0$-$X2_5$) are depicted in Figure 11 & 12.

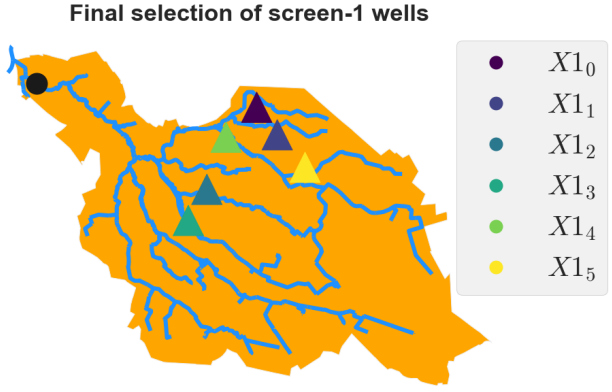


Figure 11. screen-1 wells for further research

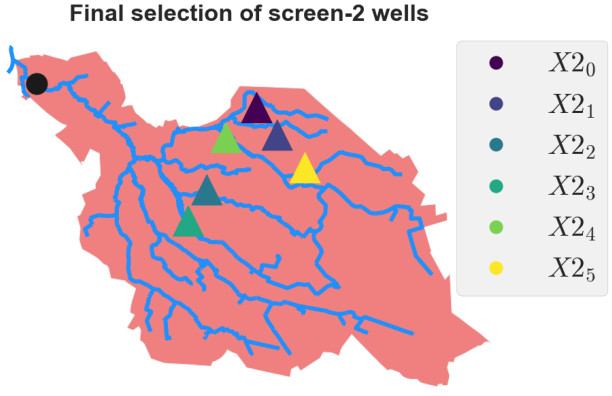


Figure 12. screen-2 wells for further research

2.3.2 Data validation groundwater wells with Pastas

The groundwater head time series of the 6 screen-1 wells ($X1_0$-$X1_5$) are depicted in Figure 13 and the series of the 6 screen-2 wells ($X2_0$-$X2_5$) are depicted in Figure 14. As can be seen from these time series, they do not have a continuous daily measurement frequency as the stream discharge timeseries has. The measurement frequency is high for most timeseries before the year 1980 and after the year 1995. This means a lot of data is missing within the necessary time period of 1985 to 2003. Moreover, the data present in the timeseries needs to be validated as well since groundwater heads can be prone to measurement errors. The 2 problems of missing data & possible errors within the data are solved by using the time series analysis (TSA) package Pastas (Collenteur et al., 2019b): an open-sourceframework for the analysis of hydrological time series.

Theory of Pastas

Pastas is making use of transfer function noise (TFN) modelling, which attempts to translate one or more input series to an output series using a statistical model (Collenteur et al., 2019b). During Pastas TSA the groundwater heads (output series) are being explained and simulated based on different explanatory variables (input series), such as: precipitation (P), potential evaporation (Ep) or actual evaporation (Ea). Moreover, extraction of pumping wells ($Qpumping$) can also cause groundwater head fluctuations and can therefore also be used as an explanatory variable. And lastly, interference with surface water can also play an important role in groundwater head fluctuations, resulting in water stages ($Hstage$) as an explanatory variable. So, TFN modelling tries to explain the observed groundwater heads by one or more other observed time series (P, Ep/Ea, $Qpumping$ and $Hstage$). A TFN model has the following structure (Collenteur et al., 2019b):

$$h(t) = \sum_{m=1}^{M} h_m(t) + d + r(t) \tag{2}$$

where $h(t)$ are the observed groundwater heads, $h_m(t)$ is the contribution of the explanatory variable m to the groundwater head, d is the base elevation of the model (a constant), $r(t)$ are the residuals, and m the number of explanatory variables. Note that this TFN model has a linear relation between the in- and output. The contribution of explanatory variable m to the groundwater head is calculated through convolution (Collenteur et al., 2019b)):

$$h_m(t) = \int_{-\infty}^{t} S_m(\tau)\theta_m(t-\tau)d\tau \tag{3}$$

where S_m is the time serie of the explanatory variable m, and θ_m is the impulse response function for stress m. The impulse response is the head response due to an instantaneous stress event of unit magnitude at time t=0 (Collenteur et al., 2019b). For example, the head response due to a precipitation event of 1mm at t=0. Note that $h_m(t)$ can also be negative, for instance with the explanatory variable evaporation: evaporation will result in a lower groundwater level, while precipitation will result in a higher groundwater level. Possible impulse response functions (τ_m) are (Collenteur et al., 2019b):

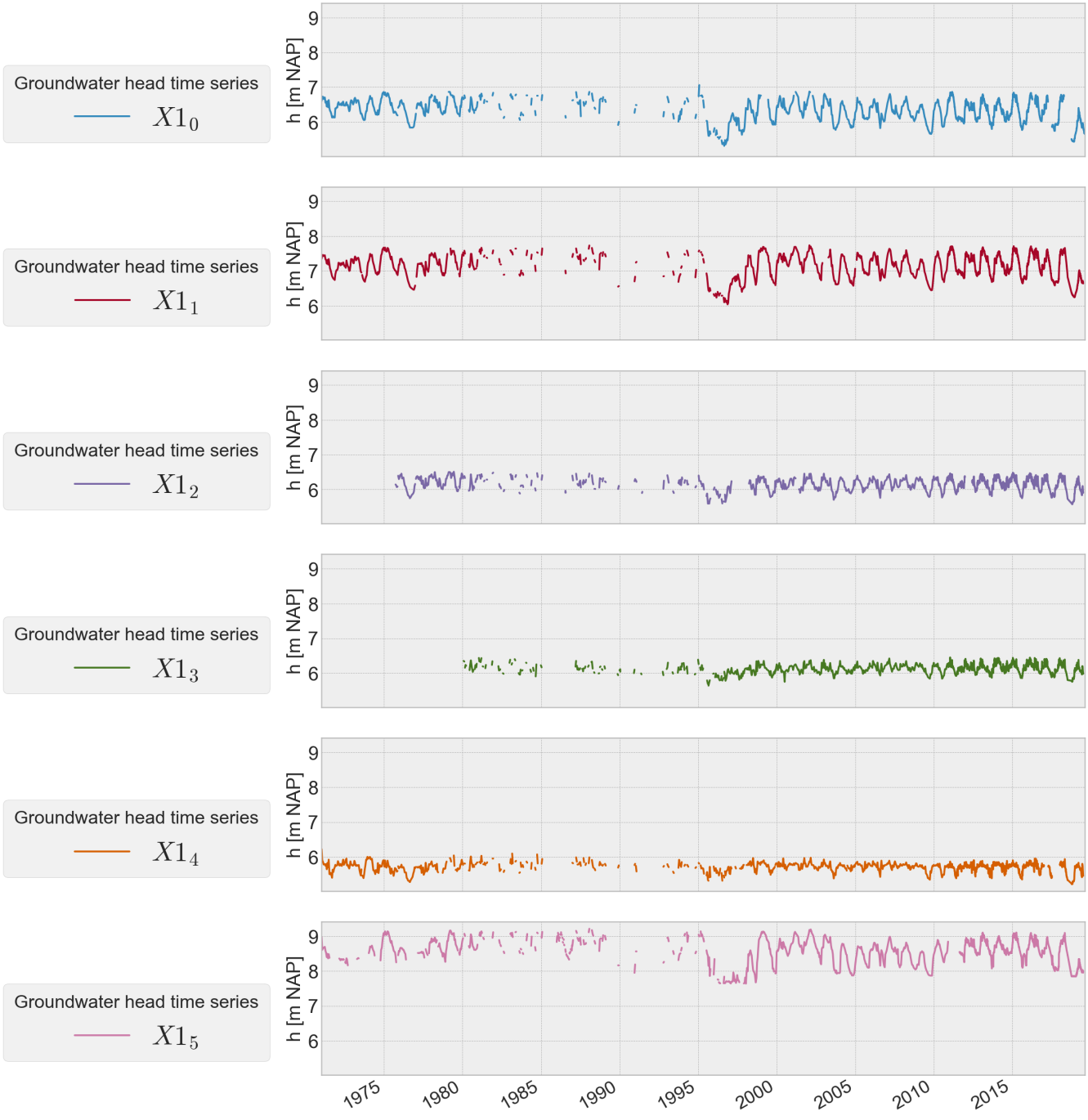


Figure 13. Observed groundwater head time series of the 6 screen-1 wells

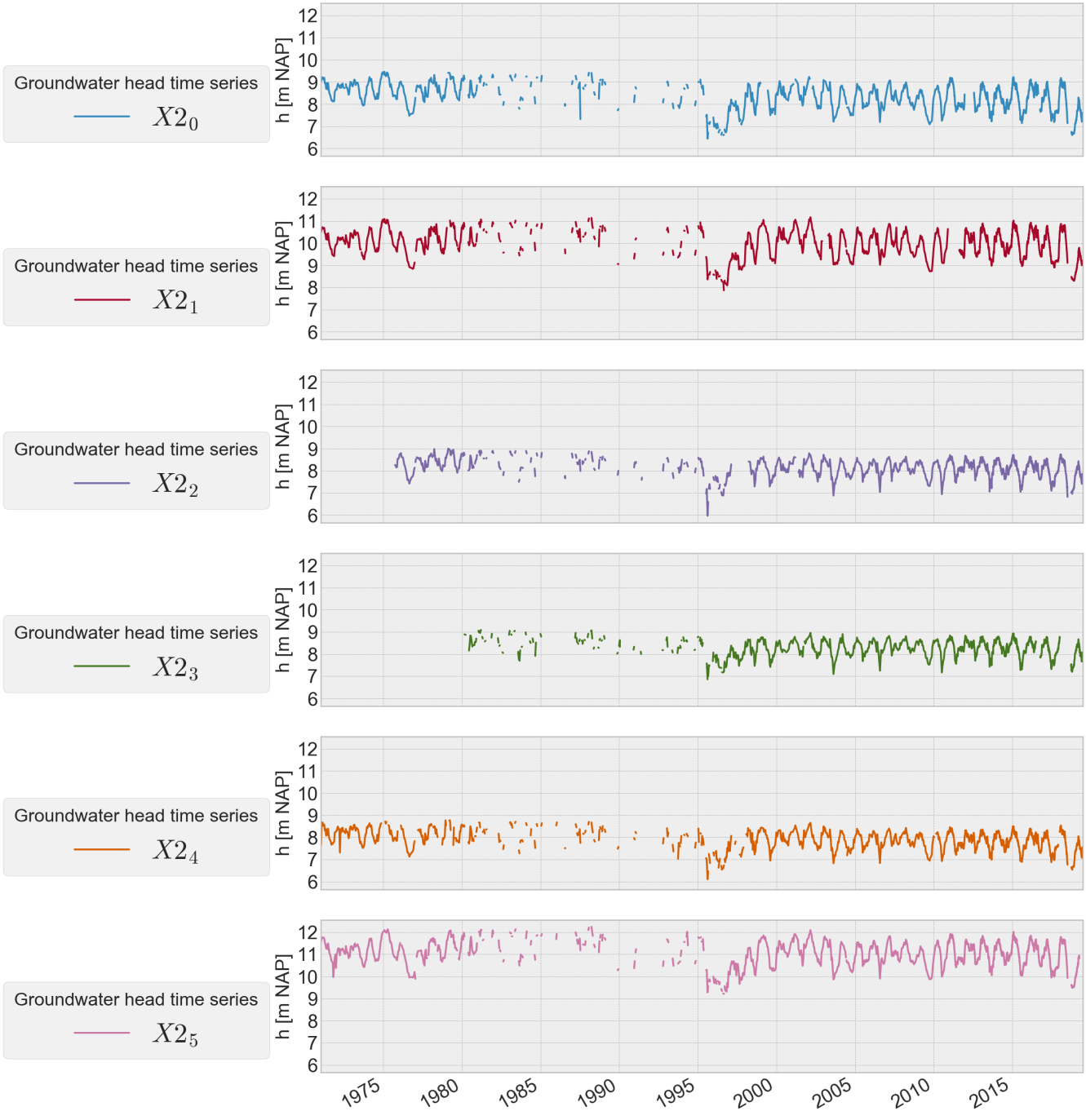


Figure 14. Observed groundwater head time series of the 6 screen-2 wells

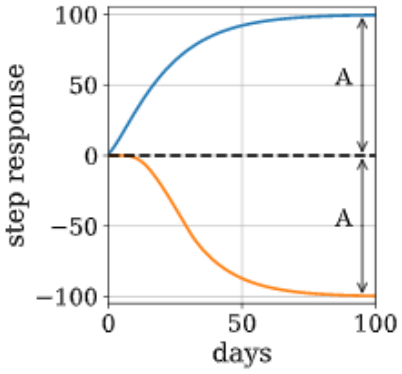


Figure 15. Example of the Gamma (blue) and Hantush (orange) step response function

- Gamma: used for groundwater response to P & Ep (recharge) with 3 parameters A, a and n
- Hantush: commonly used for groundwater response to Qpumping with 3 parameters A, ρ and Cs
- Exponential: commonly used for Hstage with 2 parameters A and a

In contrast to the impulse response function $\tau(t)$, often the step response (Θ) is used: the head response due to an uniform stress, for example the head response due to a constant pumping rate. The step response due to a constant and unit stress starting at t=0 can then be easily obtained from the impulse response through integration:

$$\Theta(t) = \int_0^t \theta(t)dt \tag{4}$$

Note that this step response will eventually reach a steady state value (the maximum change in head due to the constant and unit stress) which is defined in the parameter A of the impulse response function. Figure 15 shows an example of a Gamma step response and a Hantush step response, where the parameter A is shown. The parameter A is positive for the Gamma function, while it is negative for the Hantush function. Furthermore, it is visible that these 2 step response functions have different shape parameters. The calculation time of the time when the steady state value (time of A) has occured can become too long, and therefore, often a cutoff value of 0.99 is defined in the response functions (Collenteur et al., 2019b): the response is cutoff after 99% of the maximun change in head occured.

In addition to the parameters within the step response functions, there are also some parameters that need to be explained within the residuals (r(t)):

$$r(t_i) = v(t_i) + r(t_{i-1})e^{-\delta t_i/\alpha} \tag{5}$$

where α is the decay parameter, δt_i is the timestep between observations at t_i and t_{i-1}, and $v(t_i)$ is the noise as a result of a random process.

As a last step of the Pastas TSA, the parameters of the TFN model are estimated by optimizing the objective function which is set as the least squares solver. With these chosen parameters, the simulated heads can be calculated by looping through all explanatory variables and summing up their head contributions $h_m(t)$, and adding a certain base level d (Collenteur et al., 2019b). The difference between these simulated heads and the observed heads is then defined in the residuals $r(t)$, which subtracts the simulated heads from the observed heads. This process is performed during the calibration period, which is in terms of machine learning known as the train set procedure (Raschka and Mirjalli, 2017). The same set of parameters of the calibration period is then also used to simulate the heads for the period after the calibration period (the validation period) to examine the performance of the model. It is common to have a larger calibration period than a validation period (Raschka and Mirjalli, 2017). In this research the calibration period is set from the beginning of the observed groundwater heads (around 1970) till the end of the year 2010, while the validation period is from 2011 till 2019.

A last remark need to be made on possible outliers present in the observation series. These outliers will results in a less better TFN model fit, and this may have result on the model performance in the validation set. Therefore, clear outliers are manually removed before the Pastas TSA.

Explanatory variables Chaamse Beken

Within subcatchment Chaamse Beken there are 4 possible important explanatory variables affecting the groundwater heads, from which the two most important ones are: precipitation and actual evaporation. Actual evaporation is difficult to measure. Therefore, it is often being estimated as a factor of the potential evaporation. An additional parameter is namely added to the TFN model. The precipitation data is obtained from the KNMI station Chaam and the potential evaporation data from the weather station Gilze-Rijen just outside the Chaamse Beken. These observation series have a daily measurement frequency in the unit of mm/day and are added with a Gamma response function.

The third explanatory variable that could possible cause groundwater head fluctuations, is the pumping well from the water treatment plant Prinsenbosch (located just outside Chaamse Beken). The extraction rates of this well are obtained from Brabant Water and the time series are having a

monthly measuring frequency in the unit of $m^3/month$, being added in the form of a Hantush response function. The program Pastas is able to cope with these different measuring frequencies.

There is another explanatory variable which could have effect on the groundwater heads within the Chaamse Beken: the sewage treatment plant Chaam discharging effluent water in the streams. This last explanatory variable has a discontinuous measurement frequency in the unit of m^3/sec, and is being added with an Exponential response function. The locations of the explanatory variables explained above are depicted in Figure 16, while the timeseries of these explanatory variables are depicted in Appendix B1.

Figure 16. Explanatory variables for the TSA Pastas

Results simulations Pastas TSA

In Appendix B2 (screen-1 wells) and Appendix B3 (screen-2 wells), for each selected well a summary of the results of the Pastas TSA is collected consisting of:

- Figure of observed groundwater heads & simulated heads for the calibration and validation period
- Figure of observed groundwater heads & simulated heads for the period 1985-2003 (to be used for further research)
- Figure representing the contributions of the explanatory variables in the groundwater heads
- Table of the optimal parameters set and their boundaries
- Table of fit statistics

An example of well B50B0074_1 is given in the section below:

Figure 17 shows the observed groundwater heads and the simulated groundwater heads with the TFN model. Note that the calibration period (red dots) is used to get the model with the best fit with the optimal parameters. And in the validation period this model is used to simulate again the groundwater heads, based on the observed groundwater heads in the calibration period (black dots).

Figure 18 is zooming in on the observed & simulated groundwater heads in the research period 1985-2003. Note that with the simulated groundwater heads, a time series with daily values is created, and therefore the simulated heads are used in the further research with machine learning, instead of the observed series.

Figure 19 shows that the fluctuations in groundwater head are mostly due to the rain and evaporation. Furthermore, the pumping rates in Prinsenbosch are resulting in a constant drop of groundwater head of approximately 0.4 meters until half 1993. From then, the pumping rates result in a larger drop of a maximum of 1.1 meters. This is due to the fact that since half 1993 the pumping rates are increasing. And the last explanatory variable "Effluent Chaam" has only since its start in 1996 a positive influence on the groundwater head, with a maximum value of 0.5 meters. Note that the hydrological system is a very complex system and different processes occur also in the ground, and therefore, it is always the case to have a certain "unexplained" percentage on the groundwater head fluctuations.

B50B0074_1	Paramater	Optimal P	Pmin	Pmax
	A	1.19	0	20.88
P & Ep	**n**	0.97	0.1	10
(Gamma)	**a**	224.64	0.01	5000
	f	-1.08	-2	0
Qpumping	**A**	-0.000080	-0.020	0
Prinsenbosch	**\rho**	0.71	0.0001	10
(Hantush)	**cS**	933.84	0.001	10000
Effluent Chaam	**A**	0.00020	0	0.10
(Exponential)	**a**	59.97	0.01	5000

Table 3. A summary of the optimal parameters with their predefined boundaries for well B50B0074_1

Note that for each parameter, the minimum and maximum value are predefined as $pmin$ & $pmax$, as can be seen in Table 3. For example, the parameter f to estimate the actual evaporation as a fraction of the potential evaporation should be between 0 and -2. This is due to the fact that the potential evaporation is estimated as a reference evaporation of a certain crop. However, in reality the actual evaporation can be larger than this reference potential evaporation. For well B50B00741 the optimal parameters are all within the range of the $pmin$ and $pmax$ values.

Summarized Results

Table 4 gives an overview of the fit statistics, in which multiple criteria are given: EVP, R^2, $RMSE$, AIC and the BIC. The question arises which statistic need to be used

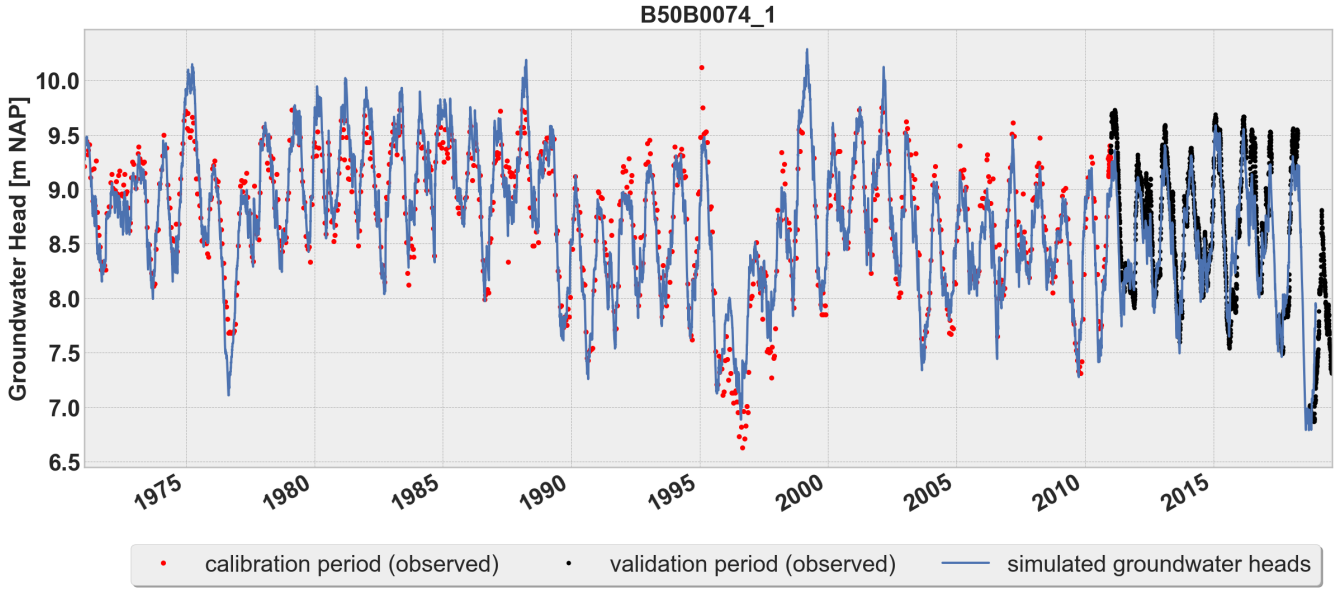


Figure 17. Observed & simulated groundwater heads with a division in calibration and validation period, for well B50B0074_1

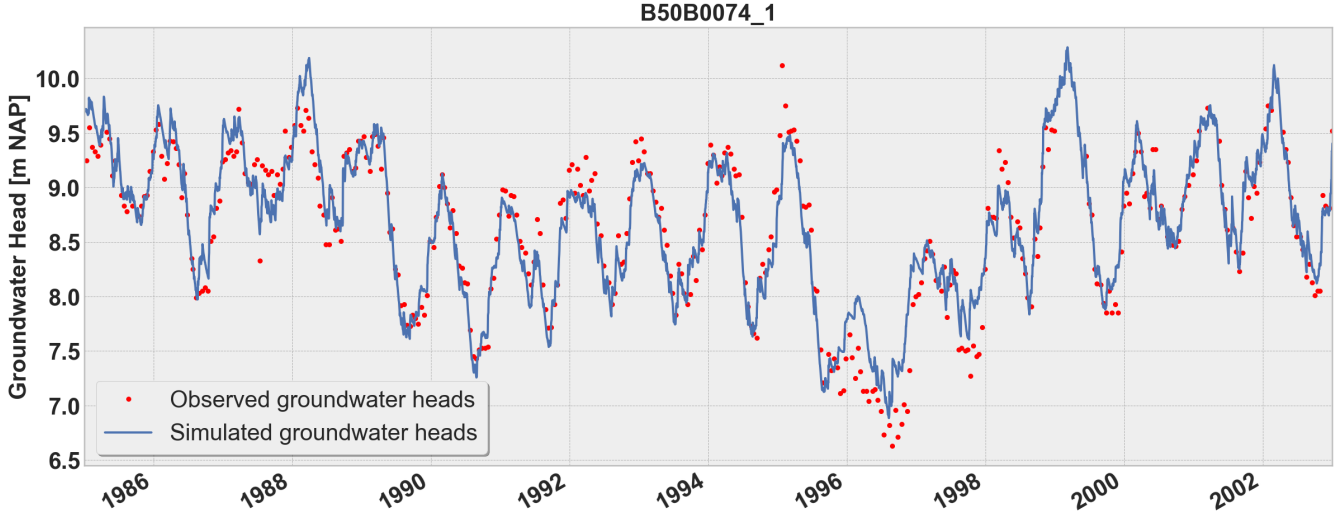


Figure 18. Observed & simulated groundwater heads for further research period 1985-2003 for well B50B0074_1

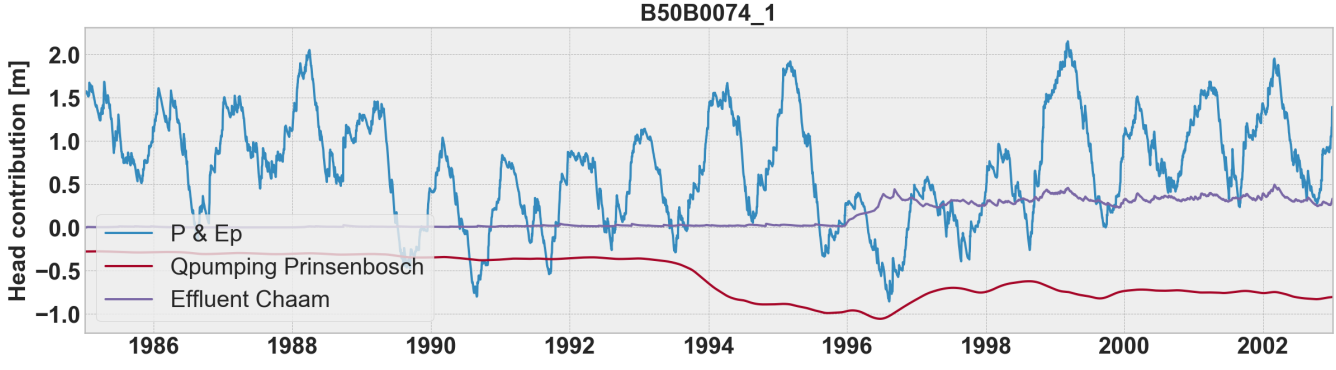


Figure 19. Contributions of the explanatory variables to the groundwater head during the research period for well B50B0074_1

Statistic Well $X1_0$	Value
Akaike information Criterion (AIC)	16.96
Bayesian Information Criterion (BIC)	69.79
Explained Variance Percentage (EVP)	88.78
Pearson R^2	0.89
Root Mean Squared Errors (RMSE)	0.21

Table 4. A summary of the fit statistics of the calibration period for well B50B0074_1

for this research. It is common in time series of groundwater heads to use the EVP as a criterion (Collenteur et al., 2019a). The **explained variance percentage** (EVP) measures the percentage to which the TFN model accounts for the variation of the observed groundwater heads. A higher EVP means a better cover of the variance within the model, and hence a better model performance. The EVP is obtained with the following formula:

$$EVP = \frac{variance(h_{observed}) - variance(h_{residuals})}{variance(h_{observed})} \quad (6)$$

in which $variance(h_{observed})$ is the variance of the observed groundwater head time series and $variance(h_{residuals})$ is the variance of the residual (observed-simulated) groundwater head time series. The EVP is used to check the performance of the TFN model first during the calibration period, and second during the validation period. The results of the EVP's for the screen-1 wells are depicted in Table 5, and for the screen-2 wells in Table 6.

Note that in the world of geohydrology it is common to say to only use observed groundwater head series that meet the requirements of an EVP of 70 or larger. This requirement is also used within this research. As can be seen in Table 5 there is one screen-1 well that does not meet the requirements of an EVP of 70 or larger: well $X1_4$. A possible reason for this low EVP can be on the one hand due to the measurements of observed head series itself, on the other hand due to the missing of an explanatory variable within Pastas TSA, or because of the fact that some groundwater head time series can not be simulated with Pastas TSA due to non-linear behaviour. The second reason of missing an explanatory variable is less likely since time series of other wells around this specific well can be simulated with Pastas TSA.

When having a look at the geographical location of well $X1_4$, it becomes clear that the location of this well is outside the presence of the already discussed Stamproy clay layer, as can be seen in Figure 20, in contrast to the other screen-1 wells. This can be a possible reason for the disability of simulating groundwater head time serie with the TFN model for well $X1_4$. Note that the simulated time series of this well will be taken into account for the further research, as the machine learning models itself are able to figure out if this variable is important enough for modelling the discharge time series.

Screen-1 well	EVP calibration period (start - 2010) [%]	EVP validation period (2010 - 2019) [%]
$X1_0$	88.78	87.10
$X1_1$	84.85	87.10
$X1_2$	88.85	82.46
$X1_3$	83.65	84.48
$X1_4$	66.96	59.14
$X1_5$	85.13	78.52

Table 5. EVP results of Pastas TSA screen-1 wells for the calibration period and the validation period

Screen-2 well	EVP calibration period (start - 2010) [%]	EVP validation period (2010 - 2019) [%]
$X2_0$	91.81	89.40
$X2_1$	88.84	91.06
$X2_2$	91.02	89.98
$X2_3$	90.57	89.67
$X2_4$	89.72	92.02
$X2_5$	91.69	90.14

Table 6. EVP results of Pastas TSA screen-2 wells for the calibration period and the validation period

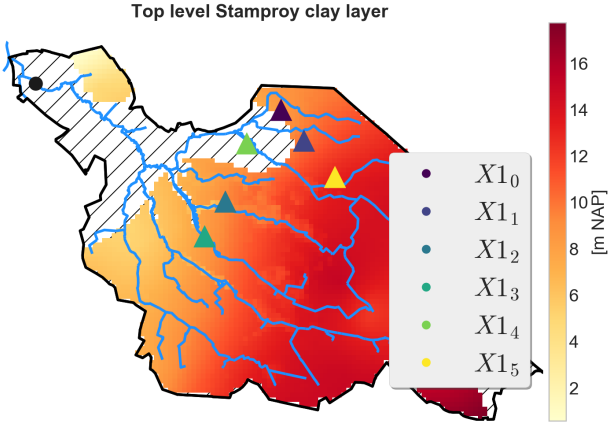


Figure 20. The top level of the Stamproy clay layer, including the locations of the screen-1 wells, showing well $X1_4$ does not lay within the clay layer

Furthermore, having a look at the EVP results, it is remarkable to see that the EVP's of the screen-2 wells are in general higher than the ones of the screen-1 wells. One possible reason for this given, is the fact that a deeper aquifer (screen-2) has less interaction with the surroundings above surface level and hence the observed head series are easier to explain with the explanatory variables than for the series of the screen-1 wells. And lastly it can be concluded that the model has also a good performance for the validation period, taking also the very dry year 2018 into account. Therefore, for the remaining part of this study the simulated series of all above-mentioned screen-1 & screen-2 wells will be taken into account for the research period 1985-2003.

2.3.3 Data analysis of groundwater wells

The simulated groundwater heads of the 6 screen-1 & -2 wells are further analysed in this section. For example, a certain correlation exists between the simulated heads of the different wells and the distance between the wells. Moreover, each well has a different response time on the precipitation, evaporation and Qpumping Prinsenbosch. Therefore, the step-response of each single well is analysed. The goal of this analysisis to get more insight into the hydrological system of the subcatchment Chaamse Beken.

Correlations of simulated heads

As expected, the correlation of the simulated head time series of wells decreases as the distance between the wells increases. This hypothesis will be tested for screen-1 & screen-2 wells. First of all, a heatmap showing the Pearson's correlation is depicted in Figure 21 for screen-1 wells, and in Figure 8 for screen-2 wells.

The Pearson's correlation coefficient is a measure for the linear correlation between two time series. The simulated series of the screen-1 wells (0.54-0.97) have a larger range than the series of the screen-2 wells (0.83-0.99). In other words, the series of the screen-2 wells are more strongly correlated than the screen-1 wells. This can again be due to the fact that screen-2 wells are located in deeper aquifers and hence depend less on events above surface level.

To test the hypotheses "the larger the distance between wells, the smaller the Pearson's correlation of the wells", for well $X1_0$ a graph representing the correlation of simulated heads versus the distance of well $X1-0$ with the other 5 screen-1 wells is shown in Figure 23. The same procedure is applied for well $X2_0$ with the other 5 screen-2 wells, as can be seen in Figure 24.

In Figure 23 it is visible that one well (well $X1_4$) does not fit within the hypotheses of the larger the distance between the wells, the smaller the correlation of the simulated heads of those wells. The same applies for well $X2_5$ in Figure 24. These 2 wells are hence an exception of the hypotheses, and overall the hypotheses can not be rejected. Furthermore, note that for the screen-2 wells the negative trendline of this

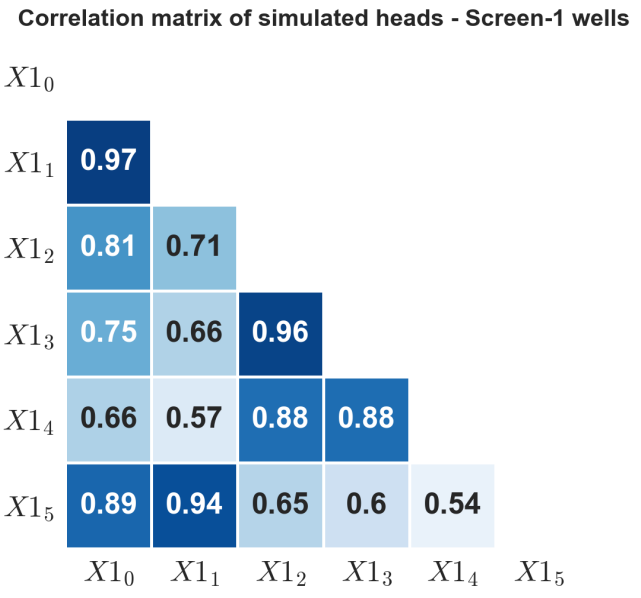


Figure 21. Pearson's correlation matrix of simulated groundwater head series screen-1 wells (0.5-1)

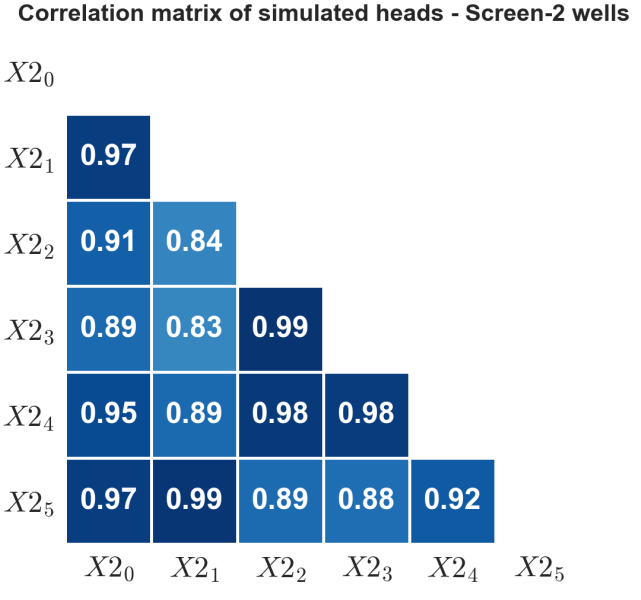


Figure 22. Pearson's correlation matrix of simulated groundwater head series screen-2 wells (0.5-1)

hypotheses is stronger than for the screen-1 wells.

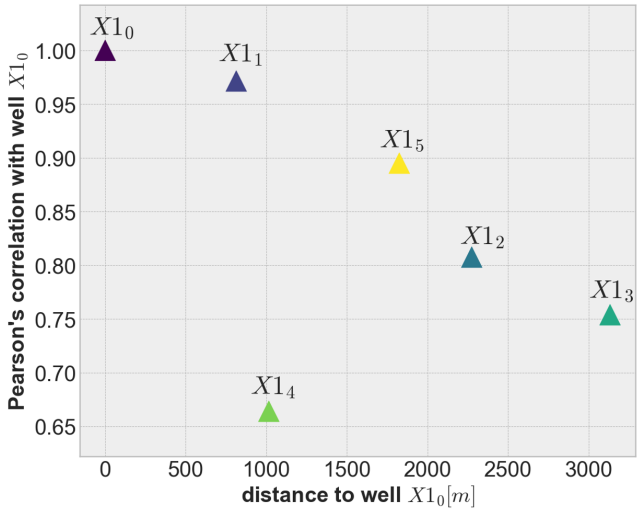


Figure 23. Relation between correlation of simulated heads and distance between well $X1_0$ and the other screen-1 wells ($X1_1$ to $X1_5$)

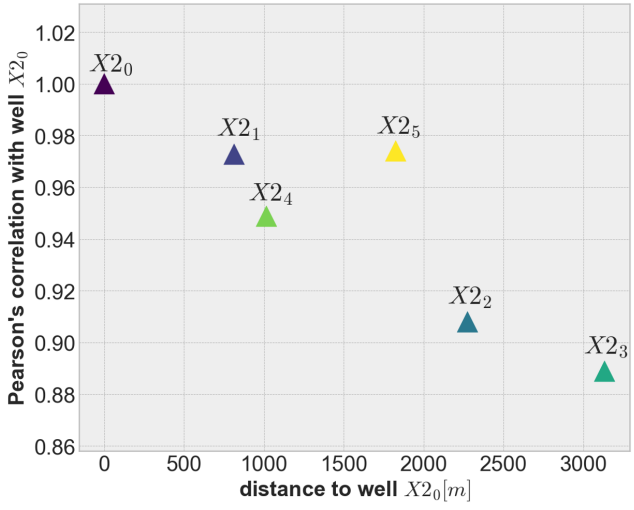


Figure 24. Relation between correlation of simulated heads and distance between well $X2_0$ and the other screen-2 wells ($X2_1$ to $X2_5$)

Step responses

The next analysis is performed on the different step responses (already explained in section 2.3.2) of the screen-1 & screen-2 wells. As said before, for the recharge a Gamma function is used, for the Qpumping Prinsenbosch a Hantush function, and for the Effluent Chaam an Exponential function. These different step response functions of the screen-1 wells are depicted in respectively Figures 25, 26 and 27.

Note that in these figures the line ends at the 99% step response: when 99% of the maximum influence on the groundwater head has occured. In reality, this line will reach a

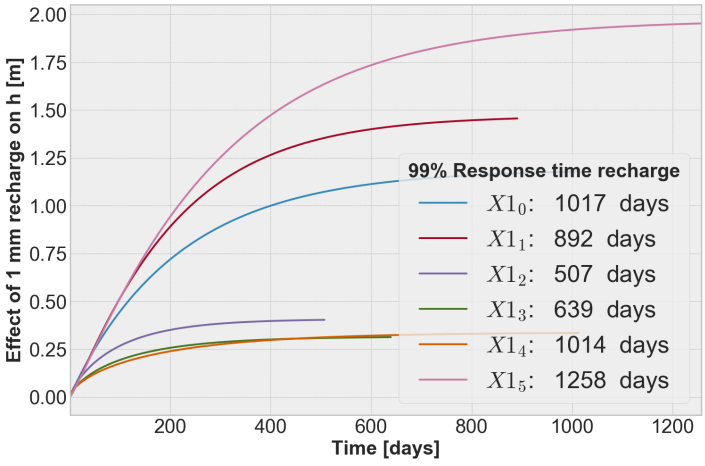


Figure 25. The Gamma step response functions on the recharge (P-E) for the screen-1 wells

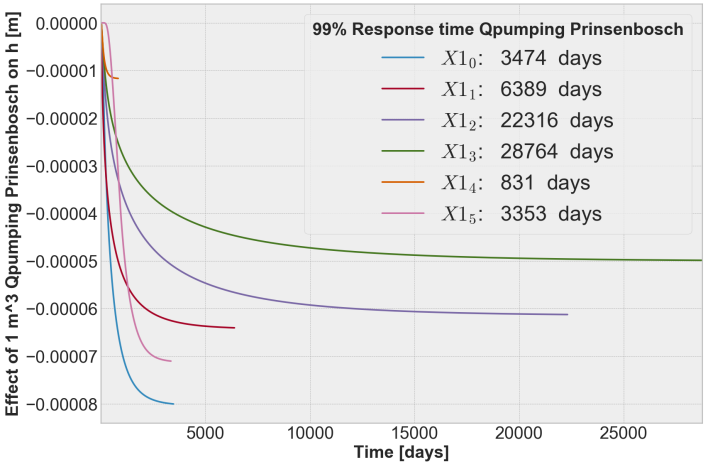


Figure 26. The Hantush step response functions on the Qpumping Prinsenbosch for the screen-1 wells

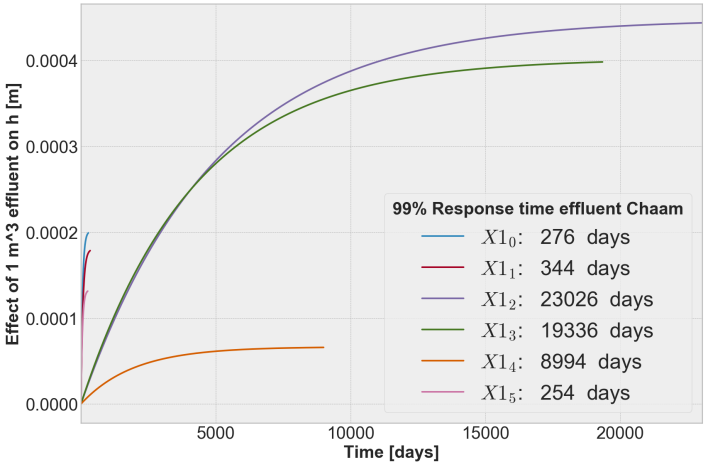


Figure 27. The Exponential step response functions on the Effluent Chaam for the screen-1 wells

steady-state which will continue for a very long time. To reduce the calculation time, the line is ended at the 99% step response time. As can be seen in Figure 25, the largest influence on the groundwater head due to the recharge is at well $X1_5$. However, it will take also the longest time to reach the 99% response time (1258 days). The lowest influence on the groundwater head due to the recharge is seen at wells $X1_3$ & $X1_4$. However, well $X1_4$ responds slower than well $X1_3$ on recharge. This indicates that the hydrological system is extremely complex and can not be seen as one system where all wells responds similarly.

Figure 26 shows the Hantush step response functions of the Qpumping Prinsenbosch, which have in general lower influence on the groundwater head fluctuations than the recharge itself. Well $X1_0$ has the largest negative influence on the groundwater head, which is as expected since this well has the smallest underlying distance with the Qpumping Prinsenbosch extraction point and hence the largest influence. Another remarkable point is the fact that well $X1_4$ has a very small 99% response time with regard to the other wells. A possible reason for this given, is the absence of the clay layer at this well, and it might therefore respond faster than the other wells. In general, it can be concluded that the extraction rates at Prinsenbosch do not affect the groundwater heads in the screen-1 wells, since the pumping well is present in a much deeper aquifer.

Figure 27 shows the Exponential step response functions of the Effluent Chaam. These step response functions can be clustered into 3 categories:

- well $X1_0$, well $X1_1$ & well $X1_5$ with very short 99% response times and an average influence on the groundwater head. These wells are also the furthest away from the effluent point Chaam.
- well $X1_3$ & well $X1_2$ with a respectively large positive influence on the groundwater heads and a very large 99% response time. These wells are also the closest to the effluent point Chaam.
- well $X1_4$ with a very low influence on the groundwater head and an average 99% response time.

The 99% response times of the different step response functions for the screen-2 wells are depicted in Table 7. Overall, it can be concluded that the groundwater system is very complex and every screen is responding differently on explanatory variables. Still a lot is unknown in this system.

	99% Response time [days]		
Screen-2 well	**Recharge Gamma**	**Qpumping Prinsenbosch Hantush**	**Effluent Chaam Exponential**
$X2_0$	820	5380	7990
$X2_1$	832	3340	293
$X2_2$	638	26826	18796
$X2_3$	499	28697	23026
$X2_4$	771	5787	8362
$X2_5$	855	7275	17739

Table 7. The 99% step responses in days for the screen-2 wells on the explanatory variables recharge, Qpumping Prinsenbosch and Effluent Chaam

3 Methods

This chapter covers in section 3.1 the first method for finding a relation between groundwater heads and stream discharge: **machine learning**. A short description is given of machine learning itself in subsection 3.1.1, followed by how it is applied in this research (subsection 3.1.2). Section 3.1.3 describes which machine learning algorithms are chosen based on literature review, including an explanation of these algorithms.

In the second section (3.2), the second method for this research is described: using a **conceptual hydrological model** to obtain the stream discharge. First of all, a conceptual model is chosen based on literature review, whereafter the model itself is thoroughly described.

In the last section of this chapter, the metrics used to evaluate the model performances are described.

3.1 Method 1: Machine learning

3.1.1 What is machine learning?

These days there is an abundance of data, which might be structured or unstructured data. Machine learning is related to Artificial Intelligence (AI) by deriving knowledge from this data in order to make data-driven decisions (Raschka and Mirjalli, 2017). It involves self-learning algorithms that derive knowledge from data in order to make predictions, identify patterns and make decisions with minimal human intervention (Raschka and Mirjalli, 2017).

Within machine learning, a distinction is made between *supervised* and *unsupervised* learning (Raschka and Mirjalli, 2017). Supervised learning labels the data before it is put into the machine in input variable(s) X and a target Y. This target Y is in fact the output, and is hence actually part of the dataset. With the machine learning algorithm we can predict the target Y for other input variables X. In contrast to supervised learning, for unsupervised learning the data is not labeled into input variables X and a target Y before it is put into the machine. The output stands actually for finding a hidden pattern/structure in the input data. In this research, supervised learning is used instead of unsupervised learning since the data is labeled into input variables and an output.

Zooming in on the principle of supervised learning, we put our data (divided into input and output) into a machine, define an algorithm within the machine, let the machine learn from the input data and output without explicitly programming ("train the model"), and in the end giving the algorithm

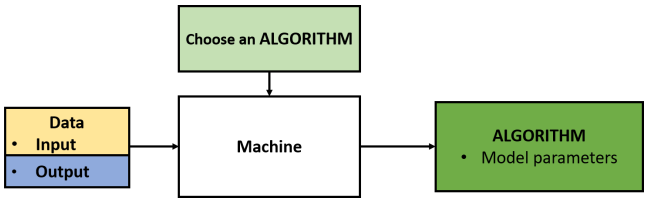


Figure 28. The self-learning process of supervised learning

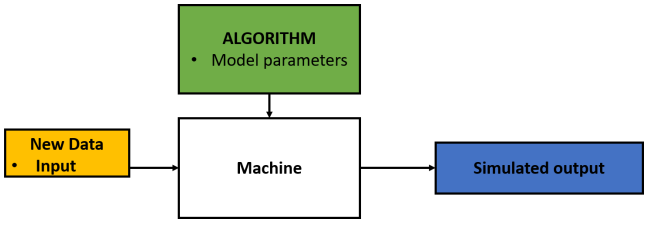


Figure 29. The usage of the self-learning process of supervised learning

with model parameters linking the input data with the output (see Figure 28). With this self-learning algorithm it is possible in the future to put a similar set of unseen input data into the machine with the self-learning algorithm, to get the desired simulated output (Figure 29).

3.1.2 Principle of machine learning within this research

Within this research the stream discharge CBU is derived from other input variables such as for example groundwater heads, precipitation and evaporation, among others. As said before, supervised learning is used, instead of unsupervised learning since the data is labeled in input variables and a target. First, the machine learning model is trained from the in- and output data from 1985-2003. Second, this trained model can be used to simulate the unknown stream discharge data from the input variables which can be used for the years where data of the input variables is known (at least the years 2003-2019 for now). The model training and usage is illustrated in Figure 30. Different important characteristics of how this learning principle is precisely used, will be elaborated below.

Different model setups and their dataset collection

In order to answer the research questions 1a to 1d for the study area Chaamse Beken, different model setups are formulated. Each setup has a different set of input variables, whereas the target Y stays the same for each setup: $Qobs$. The output of each setup $Qsim$ needs to be comparable with $Qobs$. The dataset of each setup contains of data from 1985 to 2003 and have all a daily frequency. The different model setups needed for research questions 1a to 1d are thoroughly elaborated below and are summarized in Table 8.

research question 1a

For this research question it is examined if $Qobs$ can be simulated with only one screen-1 well or if all selected screen-1 wells within Chaamse Beken are necessary for obtaining $Qobs$. To answer this research question, two model setups are used. The first model setup focuses on only one screen-1 well, namely well $X1_0$. The reason for this well is the fact that this well has the largest correlation with all the other screen-1 wells and is hence the most representative for the screen-1 wells group (Figure 21) . The second model setup takes the whole screen-1 wells group into account.

Model setup 1:

$$Y(t) = F(X1_0(t)) \tag{7}$$

- Input variable: $X1_0$ $[mNAP]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_1$ $[m^3/sec]$

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \tag{8}$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_2$ $[m^3/sec]$

research question 1b

This research question focuses on how deep the wells should be in order to find an accurate enough relation between groundwater heads and the stream discharge. In subsection 2.3.1 it is already explained that for this research only screen-1 and maybe screen-2 wells play a contributing role in the baseflow. Therefore, this research question focuses on if adding screen-2 wells as input variables improve the output of the second model setup. The model setups used for this research are hence the already explained model setup 2 and a new defined model setup 3 including the screen-2 wells.

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \tag{9}$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$

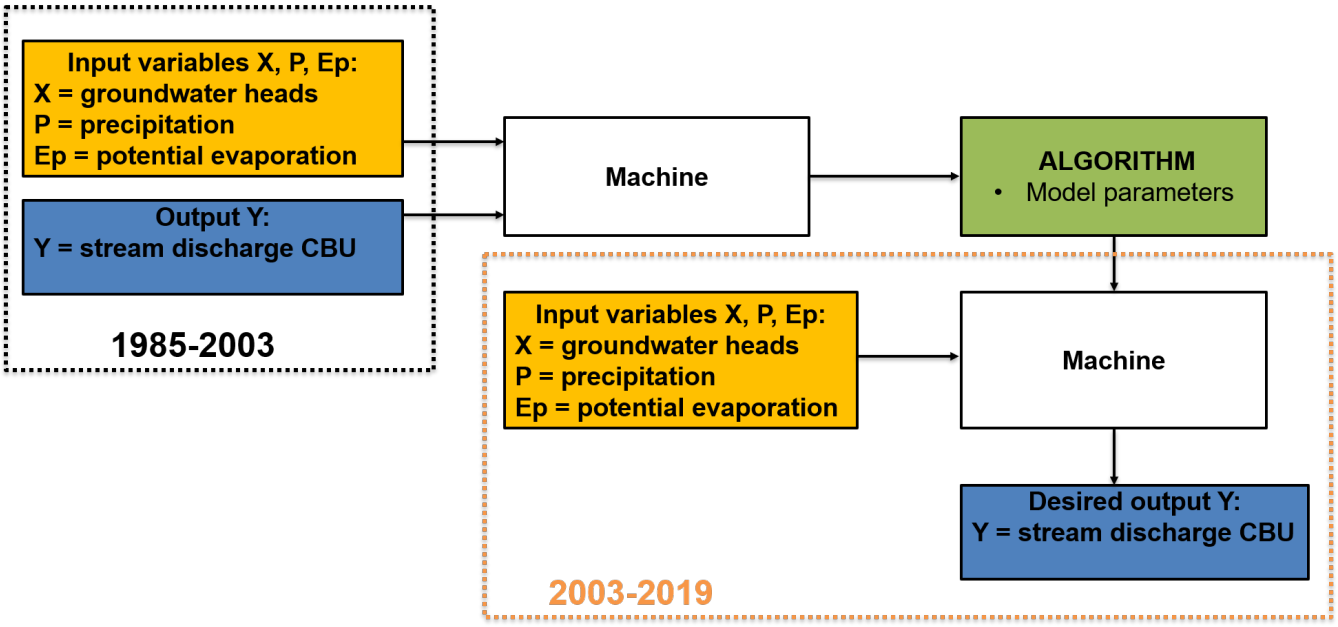


Figure 30. The supervised machine learning principle within this research

- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_2$ $[m^3/sec]$

Model setup 3:

$$Y(t) = F(X1_0(t), ...X1_5(t) \& X2_0(t), ...X2_5(t)) \quad (10)$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$
- Input variable: $X2_0$ $[mNAP]$
- Input variable: $X2_1$ $[mNAP]$
- Input variable: $X2_2$ $[mNAP]$
- Input variable: $X2_3$ $[mNAP]$
- Input variable: $X2_4$ $[mNAP]$
- Input variable: $X2_5$ $[mNAP]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_3$ $[m^3/sec]$

research question 1c

This research question focuses on if adding hydrological variables P and Ep improves the output of the model when only using screen-1 wells. For answering this research question, the already defined second model setup is compared with a new defined model setup (setup 4), in which the screen-1 wells are taken into account and the hydrological variables P and Ep.

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \quad (11)$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_2$ $[m^3/sec]$

Model setup 4:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t)) \quad (12)$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$
- Input variable: P $[mm/day]$
- Input variable: Ep $[mm/day]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_4$ $[m^3/sec]$

Note that in these above mentioned model setups (1-4), the timestep t is in days. This means that the relation with the stream discharge is tried to be found for data samples at the

same days: "simulating $Qobs$ at day t based on the input variables at day t".

research question 1d

It is known within the hydrological system that the input variables are subject to delays Δt. For example, a part of the rainfall will percolate to the groundwater and it can take days, months or even years before it enters the stream and hence contributes to the stream discharge (Gonzales et al., 2009). Furthermore, the hydrological system has a certain memory: groundwater heads one day, one month or even one year prior to the groundwater head at t can also have effect on the relation with the stream discharge (Gonzales et al., 2009). This research question focuses on if adding these delays and memory to the input variable set of model setup 4 (screen-1 wells, P and Ep) improve the output of the model, resulting in model setup 5.

To account for these delays Δt and memory M of the system, a new set of input variables is added to model setup 5. For the delays, we take into account a delay of 1, 2, 3, 4, 5 and 6 days. This can be added in the dataframe by shifting only the input variables 1,2,3,4,5 and 6 days forward. This is only performed for the input variables P & Ep, as groundwater heads will not change that much in a period of 6 days. In addition, the memory will be added in the form of rolling means: for the groundwater heads the mean of the past $t-3$, $t-7$, $t-14$, $t-30$ and $t-120$ days is added at $t+1$. The shifting and rolling principle is explicitly explained in Figure 31.

Date	X with rolling 3 days	Input Variable X [m NAP]	X with shift 1 day	Target Y [m^3/sec]
01-01-1985	-	8	-	1
02-01-1985	-	9	8	1
03-01-1985	-	10	9	2
04-01-1985	10+9+8/3	10	10	3
05-01-1985	9+10+10/3	9	10	2

Figure 31. Shifts (green) and rolling means (red) to add delay and memory in the system, only performed for the input variables

Note that shifted input variables P and Ep will get an addition "_S+("number of days shifted") in their names. The same principle applies for the rolling means of the screen-1 wells, as they will get an addition of "_R+("number of rolling days") in their names. This results in the following model setups for research question 1e:

Model setup 4:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t)) \quad (13)$$

- Input variable: $X1_0$ $[mNAP]$
- Input variable: $X1_1$ $[mNAP]$
- Input variable: $X1_2$ $[mNAP]$
- Input variable: $X1_3$ $[mNAP]$
- Input variable: $X1_4$ $[mNAP]$
- Input variable: $X1_5$ $[mNAP]$
- Input variable: P $[mm/day]$
- Input variable: Ep $[mm/day]$
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_4$ $[m^3/sec]$

Model setup 5:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t), Mand\Delta t) \quad (14)$$

- Input variable: $X1_0$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: $X1_1$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: $X1_2$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: $X1_3$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: $X1_4$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: $X1_5$ $[mNAP]$ + R3, R7, R14, R30, R120
- Input variable: P $[mm/day]$ + S1, S2, S3, S4, S5, S6
- Input variable: Ep $[mm/day]$ + S1, S2, S3, S4, S5, S6
- Target Y: $Qobs$ $[m^3/sec]$
- Output: $Qsim_5$ $[m^3/sec]$

Now that we have different model setups, also different machine learning algorithms are needed to actually use these different model setups. However, first, a way of testing each machine learning model is explained.

Input variables	Setup 1	Setup 2	Setup 3	Setup 4	Setup 5
$X1_0$	x	x	x	x	x
$X1_1$		x	x	x	x
$X1_2$		x	x	x	x
$X1_3$		x	x	x	x
$X1_4$		x	x	x	x
$X1_5$		x	x	x	x
$X2_0$			x		
$X2_1$			x		
$X2_2$			x		
$X2_3$			x		
$X2_4$			x		
$X2_5$			x		
P				x	x
P-S1					x
P-S2					x
P-S3					x
P-S4					x
P-S5					x
P-S6					x
Ep				x	x
Ep-S1					x
Ep-S2					x
Ep-S3					x
Ep-S4					x
Ep-S5					x
Ep-S6					x
$X1_0$-R3					x
$X1_0$-R7					x
$X1_0$-R14					x
$X1_0$-R30					x
$X1_0$-R120					x
$X1_1$-R3					x
$X1_1$-R7					x
$X1_1$-R14					x
$X1_1$-R30					x
$X1_1$-R120					x
$X1_2$-R3					x
$X1_2$-R7					x
$X1_2$-R14					x
$X1_2$-R30					x
$X1_2$-R120					x
$X1_3$-R3					x
$X1_3$-R7					x
$X1_3$-R14					x
$X1_3$-R30					x
$X1_3$-R120					x
$X1_4$-R3					x
$X1_4$-R7					x
$X1_4$-R14					x
$X1_4$-R30					x
$X1_4$-R120					x
$X1_5$-R3					x
$X1_5$-R7					x
$X1_5$-R14					x
$X1_5$-R30					x
$X1_5$-R120					x
Qsim	$Qsim_1$	$Qsim_2$	$Qsim_3$	$Qsim_4$	$Qsim_5$

Table 8. The different model setups used in this research. Each model setup has a different set of input variables, whereas the target Qobs stays the same for each model setup. Each model setup results in a different $Qsim$ ($Qsim_1 to Qsim_5$

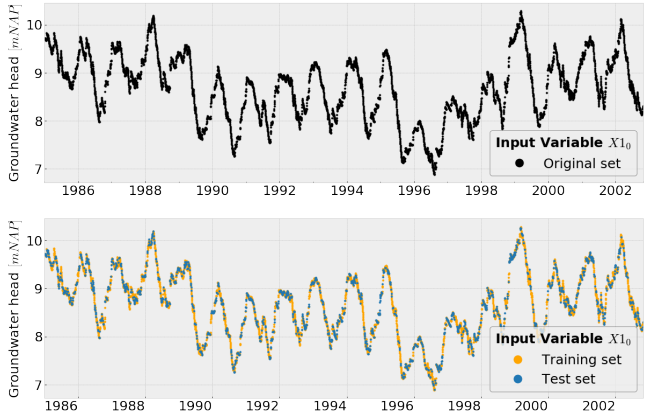


Figure 32. The **random** division of the input variable $X1_0$ into a training and test set, used in this research

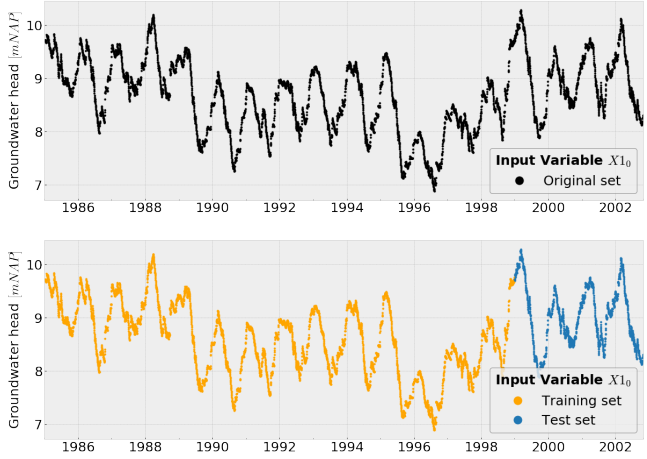


Figure 33. The **non-random** division of the input variable $X1_0$ into a training and test set, used in this research

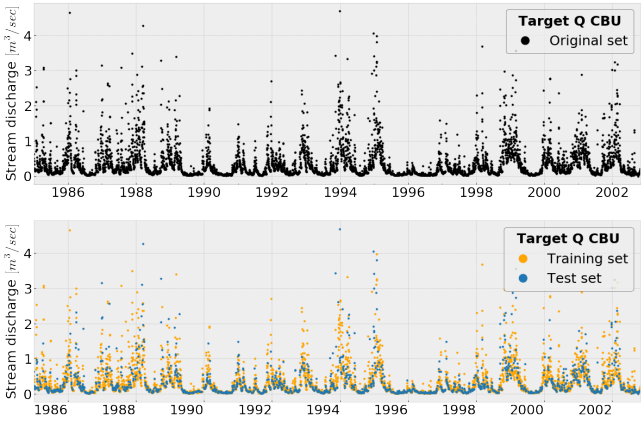


Figure 34. The **random** division of the original target $Q_C BU$ into a training and test set, with the same division as in the random division of the input variable

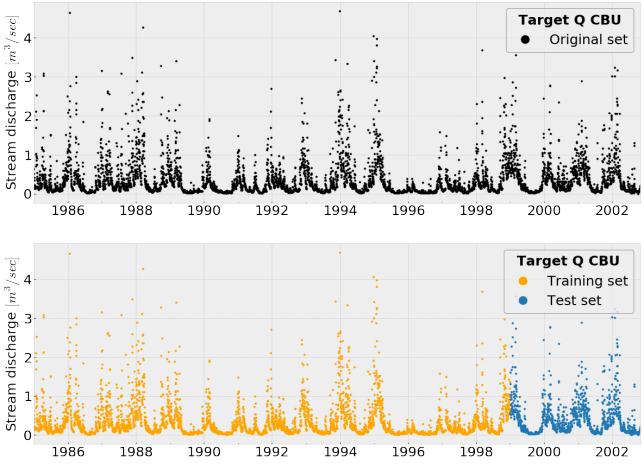


Figure 35. The **non-random** division of the original target set $Q_C BU$ into a training and test set, with the same division as in the non-random division of the input variable

Division in training and test set

For each machine learning algorithm it is important to test the performance of the machine learning model. Therefore, the dataset of 1985-2003 is split into a training and test set: the training set to train the model and find the optimal model parameters of the algorithm ("train the model"), and a test set to use this model with the optimized model parameters to derive the output of the model ($Qsim$) and hence compare it with the original output ($Qobs$) ("test or evaluate the model"). The question arises how large the training set and hence the test set should be. The training set should be large enough in order to find the possible relation between the input(s) and output of each model. However, there should still be a part left to use as test set. This test set should be representative for the total set, to make sure there are enough dynamics in the system.

From previous hydrological researches (Zhao et al., 2019; Sachindra et al., 2018; Buckingham et al., 2015) it can be said that approximately 65-75% of the data should be set as the training set, and the remaining 25-35% as the test set, in order to capture a relation between the input(s) and output. Note that in the dataset 1985-2003, dry years and wet years are present. To include these wet and dry years as much as possible in the training set and in the test set, the training set is chosen as 75% of the total set, whereas the test set is only 25%.

A common approach in machine learning is to randomly divide the total set in a training set and a test set (Raschka and Mirjalli, 2017). An example of a randomized split of an input variable and the target is depicted in Figure 32 & Figure 34 .

However, in this research this common random splitting in training and test set is not applied. This is due to the fact that the goal of this research is to fill up full years of stream discharge between 2003-2019, and simulate full years after

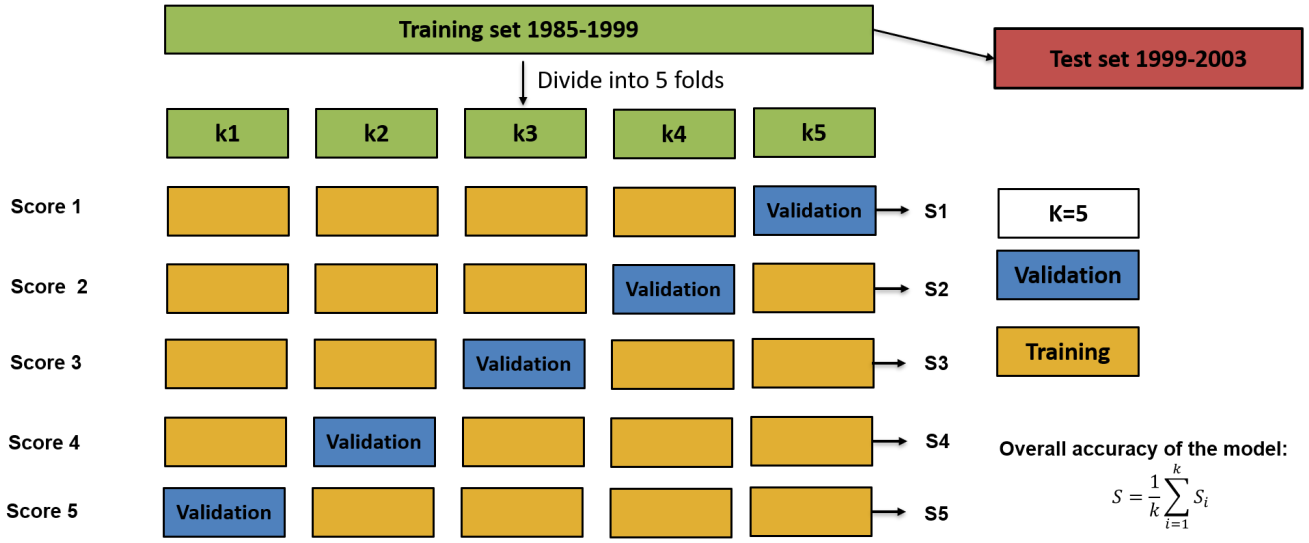


Figure 36. The principle of 5-folds grid search cross validation

2019. So, full years as a test set are also wanted. This means that the first 14 years (1985-1999) are the training set, and the remaining 4 years (1999-2003) are part of the test set (Figures 33 & 35). These 4 years within the test set consists of dry and wet years and are hence a good representation for the total dataset.

With this random or non-random division of a training and test set, a data point is either part of the training set, or part of the test set. Note that the non-random division in a training and test set has been applied before in hydrological researches, for example in (Giri et al., 2019).

Model parameter and hyperparameters

The division of the dataset into a training and test set is explained above. It might get more complicated when taking hyperparameter tuning into account. Each machine learning model has a few *model parameters* and a set of *hyperparameters* (Raschka and Mirjalli, 2017). A machine learning model is the definition of a mathematical formula with a number of *model parameters* that need to be learned from the data (Probst et al., 2019). During training of the model, the model parameters are determined such that the model fits best to the existing data.

On the other hand, there are *hyperparameters* which can not be directly learned from the regular training process. These hyperparameters express "higher-level" properties of the model such as its complexity or how fast it should learn (Raschka and Mirjalli, 2017; Probst et al., 2019). These parameters must be set manually before training the model (Probst et al., 2019). In other words, hyperparameters are settings of an algorithm that can be tuned to optimize the model performance.

The question arises which hyperparameters are optimal for the model performance. The hyperparameters can be chosen randomly, but a common procedure within machine learning is to use **5-Folds Grid Search Cross Validation**,which is explained below.

5-Folds Grid Search Cross Validation

As a first step, the hyperparameters needed to be tuned are selected for each machine learning algorithm and for each hyperparameter a dictionary of values is defined. For example, for the algorithm random forest, the depth of the trees is set as [2,4,6] and the number of regression trees within the random forest is set as [10,25,50]. This means in total 3*3 = 9 different sets of hyperparameters. For the second step, the training set is taken into account (the test set is left apart) and is divided into 5 folds. Each fold must be used for validation once, and k-1 times for training (see Figure 36). Until this point, the steps are just preparations for the hyperparameter tuning with 5-Folds Grid Search Cross Validation. The hyperparameter tuning starts with taking the first hyperparameter set (tree depth = 2, number of trees = 10), training the model with this hyperparameter set for the k-1 folds and evaluating the model performance with the first hyperparameter set on the first validation fold (score 1). The accuracy of the model is noted as S1. The next step is to go through score 2 where another fold is chosen as the validation set. The accuracy of the model for score 2 is noted as S2. This process continues until all 5 folds have been validation set once. So for the first hyperparameter set, S1 to S5 are noted down. In the end, the evaluation of the total model (S) for the first hyperparameter set is calculated as the average of each single

performance (S_k):

$$S = \sum_{i=1}^{k} S_i \tag{15}$$

For the first hyperparameters set 5 loops are performed. For the second hyperparameter set another 5 loops are performed. This process is continued untill all hyperparameters sets have followed the procedure explained above. For the 9 different hyperparameters sets of the example, in total 9*5=45 loops are performed. The optimum hyperparameter set is chosen as the one that gives the best overall accuracy of the model. Note that the larger the hyperparameters set and the number of folds chosen, the more loops are performed and the longer the calculation time. Therefore, the number of folds in researches when applying k-folds grid search cross validation is often maximized to 5 (Lu et al., 2019), as is also the case for this research.

After having found the best hyperparameter set with 5-Folds Grid Search Cross Validation, the total training set is used again to retrain the model with the found hyperparameter set.

Overfitting and underfitting

Overfitting and underfitting are two phenomena that often occur within machine learning. *Overfitting* occurs when the machine learning algorithm captures the noise of the data. It often occurs when the algorithm fits the data too well. Specifically, overfitting occurs if the model shows low bias but high variance.

Underfitting on the other hand, occurs when the machine learning algorithm cannot capture the underlying trend of the data. It occurs when the algorithm does not fit the data well enough. Specifically, underfitting occurs if the algorithm shows low variance but high bias. An example of underfitting, an optimal fit and overfitting are depicted in Figure 37. The precise principle of the variance and bias affecting overfitting or underfitting is outside the scope of this research. However, for each machine learning algorithm and its model setup, overfitting and underfitting needs to be checked.

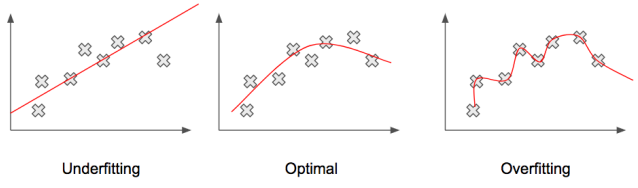


Figure 37. An example of underfitting, an optimal fit and overfitting

3.1.3 Chosen Machine Learning Algorithms

Data-driven models present *linear* or *non-linear* relationships between input variables and its target. The least squares (LS), multiple linear regression (MLR), autoregressive (AR), moving average (MA), autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) are examples of algorithms assuming a linear relationship exist between the input and output of these models (Adnan et al., 2019). These linear algorithms have been applied for streamflow forecasting since 1970 (Yaseen et al., 2015). For example, AR, ARMA and ARIMA have been presented by Salas et al. (1980) in the application of modelling hydrologic time series. Moreover, Valipour (2015) used an ARIMA model for long-term runoff forecasting in the United States. Since the streamflow process is complex and it is characterized by nonlinear relationship between streamflow and the characteristics of its watershed, these linear algorithms result in poor model performance (Adnan et al., 2019).

Therefore, in the last decades, researchers have concentrated on machine learning algorithms that can capture the non-linear relationship (Adnan et al., 2019), for example model tree (MT), support vector machine (SVM) and artifical neural networks (ANN's). These models are helpful in identifying the inherent nonlinearity in streamflow processes, according to Adnan et al. (2019).

Stravs and Brilly (2007) succesfully applied a MT to predict streamflows of several tributaries of Sava River and concluded that especially MT is useful for streamflow prediction during rainless periods. This means that MT's are useful in predicting the streamflow during baseflow conditions. A MT model can exist of one regression tree as in **decision tree regression (DTR)**, or of multiple trees in **random forest regression (RFR)** and **gradient boosting regression (GBR)**. RFR and GBR are an extension of DTR and are able to improve the model performance with regard to DTR. As these models proved to be useful in predicting the baseflow, these three mentioned machine learning algorithms are considered in this research. In addition to these MT algorithms, SVM is also considered in this research as a machine learning algorithm to have also another kind of algorithm than tree-based modelling. According to Msiza et al. (2007) SVM in the form of regression (**support vector regression (SVR)**) shows to be equal to ANN's. Moreover, ANN's are rather used when there is big data and that is not the case in this research. Therefore, ANN's are not considered in this research.

Note that in all above-mentioned researches the discharge time series itself has been used as an input variable to *predict* the discharge at 1,2,3 days, 1 month or 1 year ahead. In this research, the discharge time series itself is not part of the input variable set. It can be said that the discharge is not predicted, but *simulated* on day i based on other hydrological time series on day i.

To conclude, the 5 different model setups (as described in subsection 3.1.2) will be used by applying four different machine learning algorithms: **decision tree regression (DTR)**, **random forest regression (RFR)**, **gradient boosting regression (GBR)** and **support vector regression (SVR)**. DTR, RFR, GBR and SVR, as the name already suggests, are based on a form of regression. The different algorithms

are explained below. For simplicity reasons, each single algorithm is explained with only 2 input variables $X1_0(t)$ & $P(t)$ and the target $Qobs(t)$. Furthermore, the simulated discharge is defined as $Qsim$ and the observed target as $Qobs$. The goal of these machine learning algorithms is to find the hyper- and model parameters such that the difference between $Qsim$ and $Qobs$ is minimized.

Decision Tree Regression (DTR)

DTR consists of one single regression tree. The general idea of regression trees is to *(a)* split the input variables space into nested rectangular regions (Ri), and *(b)* within each region, the target $Qobs$ is simulated with a constant value ($Qsim_i = c_i$) (Bhatnagar, 2018). This constant value is the average of all the corresponding $Qobs$ in that region. The input variables space splitted in regions is visualised in Figure 38, according to an example of the corresponding regression tree (Figure 39).

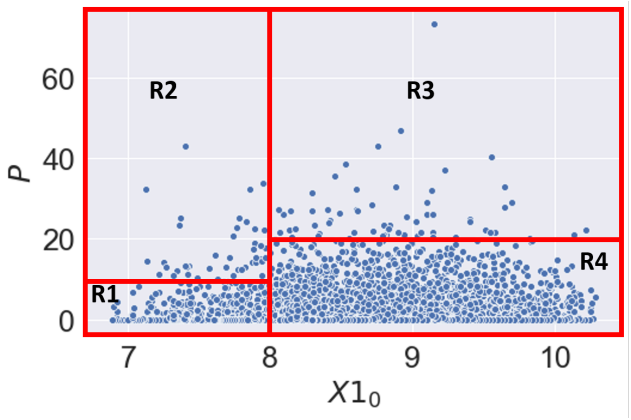


Figure 38. A visualisation of the input variables divided in multiple nested rectangular regions, used for DTR

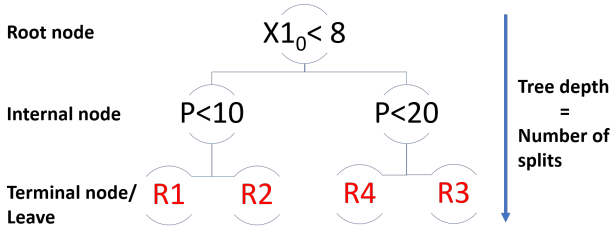


Figure 39. The corresponding regression tree of Figure 38

This method is called a tree, because the rectangular regions are created by using a branching structure in which each branch is a binary split obtained by applying a threshold to the value of one of the input variables (Mishra and Datta-Gupta, 2018). Each tree starts with the **root node** where the binary splitting starts and the **internal nodes** where the splitting process continues until a **terminal node** is reached. The terminal nodes are also called the **leaves** of the tree. Each regression tree has a certain **tree depth**, which is equal to the number of splits. These definitions are also visualised in Figure 39.

The question arises how the input variables space can be splitted into the best regions, and therefore regression trees make use of the so-called CART algorithm, consisting of 3 important components (Bhatnagar, 2018):

1. Selecting the best partition: defining a criterion to select the best partition/split among all input variables. This component is set in the hyperparameter "criterion".
2. Stopping rule: a rule to decide when a node is terminal, such that it becomes a leaf.
3. Setting the tree depth: to avoid overfitting.

Note that these components are defined in different hyperparameters of the regression tree. These components and their corresponding hyperparameters are shortly explained below, starting with the first one. The objective of the regression tree is to find the regions $R_1, .., R_j$ that minimize the squared error loss (defined in the hyperparameter "criterion" = mean squared error):

$$\sum_{j=1}^{j} \sum_{i \in R_j} (Qobs_i - \overline{Qobs}_{R_j})^2 \tag{16}$$

with $Qobs_i$ is the observed discharge of each sample i and $Qobs_{Rj}$ is the mean observed discharge for the samples within the jth box R (Bhatnagar, 2018). The solution is found by the top-down "greedy" approach. It begins at the top of the tree at which point all samples belong to a single region (top-down), followed by binary splitting: each split creates exactly two internal nodes R_1 & R_2 which leads to the greatest possible reduction in the residual sum of squares.

$$R_1(j,s) = \{x | x_j < s\}, \qquad R_2(j,s) = \{x | x_j \geq s\} \tag{17}$$

The values j and s can then be found, that minimize the following equation:

$$\sum_{i: x_i \in R_1(j,s)} (Qobs_i - \overline{Qobs}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (Qobs_i - \overline{Qobs}_{R_2})^2 \tag{18}$$

This algorithm is called "greedy", because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step (Mishra and Datta-Gupta, 2018; Bhatnagar, 2018). This process is continued until the leaves of the tree are reached. The second component

of the algorithm determines when the leaves are reached, and hence the end of the tree-building process (the stopping rule). This is determined by two hyperparameters: " the minimum number of observations that must exist in a node in order for a split to be attempted", and "the minimum number of observations in any leaf". Furthermore, the last step of the algorithm is to set the hyperparameter "tree depth", in order to avoid a very complex regression tree resulting in overfitting (Bhatnagar, 2018). In the example of Figure 39 the tree depth is set at 3, equal to the number of splits in the regression tree. This algorithm is trained based on the training data. In other words, the rectangular regions R_j are defined by only using the training data. When $Qsim_i$ from the input variables of the test data for a specific sample i needs to be derived, it can be found out in which region R_j the sample i fits within. For example, if sample i fits within region R_1, it will get a $Qsim$ of the mean $Qobs$ of the training data within that region.

Let's assume now only having one variable $X1_0$ and the regression tree is only the first split of Figure 39. To visualize and understand this binary split a bit more, this split is visualised in Figure 40. Apparently if a groundwater head lower than 8.0 m, the $Qsim$ will be 0.062 m^3/sec, while a groundwater head equal to or larger than 8.0 m will result in a $Qsim$ of 0.407 m^3/sec. Equation 17 is solved with a cutoff point s of 8.0 m for the variable $X1_0$.

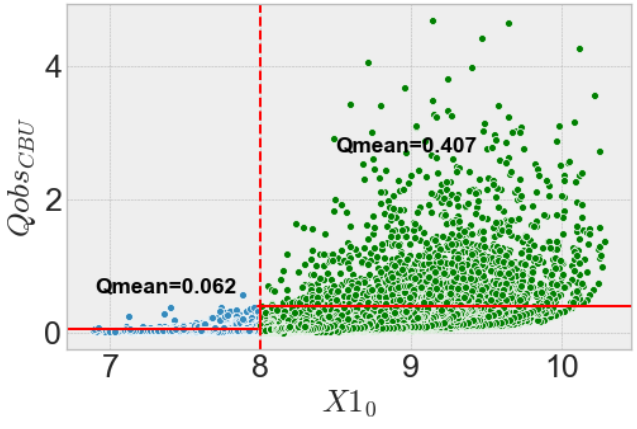


Figure 40. The first split in the regression tree of Figure 38: groundwater heads of well $X1_0$ smaller than 8.0 m will get a $Qsim$ of 0.062 m^3/sec and groundwater heads equal to/larger than 8.0 m will get a $Qsim$ of 0.407 m^3/sec, when using DTR.

Note that in the examples above the "mean squared error" is used as the partition criteria. However, other partition criteria can also be used within the first step of the CART algorithm. Another popular partition criteria in addition to the "mean squared error (MSE)" is the "mean absolute error (MAE)":

$$\sum_{j=1}^{j} \sum_{i \in R_j} | \, Qobs_i - \overline{Qobs}_{R_j} \, | \tag{19}$$

The algorithm DTR is applied for the 5 different model setups by using 5-folds grid search cross validation with the following grids for the hyperparameters (often used for DTR)(Mishra and Datta-Gupta, 2018):

Hyperparameters DTR	Grid
partition criteria	['mse', 'mae']
max tree depth	[2,4,6,8,10]
min samples leaf	[1,2,4]
min samples split	[2,5,10]

Table 9. The hyperparameters set used for the 5-folds grid search cross validation for DTR

This results in 2*5*3*3=90 different hyperparameter sets and taking 5-folds into account, in total 5*90=450 loops are performed for the DTR algorithm, for each model setup.

Random forest Regression (RFR) & Gradient Boosting Regression (GBR)

Random forest regression (RFR) and gradient boosting regression (GBR) are both built from an ensemble of multiple regression trees to increase the performance of a single regression tree. However, the ensembling of the regression trees is different in RFR and GBR (Mishra and Datta-Gupta, 2018).

Random Forest Regression (RFR)

RFR is ensembling the regression trees by using *b*ootstap *agg*regation or *"bagging"* (Mishra and Datta-Gupta, 2018): build a model n times and take for each model a different set of samples (approximately 60%) from the training set. For each model, the same set of input variables is used, while the samples are different in each model. So, in total n different subsets are extracted from the training data, resulting in n models. Note that for RFR a model is equal to one single regression tree. The samples are chosen randomly with replacement. Replacement means that a certain sample from the training set can be picked multiple times for choosing a subset. For example, sample i can be present in subset n four times. Each model with its own subset will result in a single $Qsim$, according to the principle of DTR. In the end, we take the mean of all the $Qsim$ from each single model to get our final desired output $Qsim$. The process of "bagging" is visualised in Figure 41 and Figure 42.

For RFR, using the entire training set in building a regression tree will eventually always result in the same regression tree. Therefore, the principle of "bagging" is used (Mishra and Datta-Gupta, 2018). In this way, each regression tree focuses on subtly different aspects of the input variable-target relationship. Aggregating these regression trees is expected to combine the information of the relationships into a powerful simulation tool of the discharge.

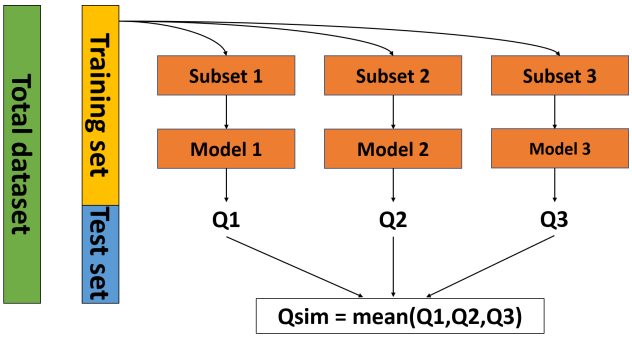


Figure 41. An example of the "bagging" process with 3 models ($n = 3$), used for RFR with each model visualising another regression tree

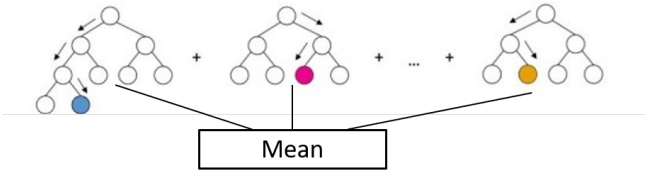


Figure 42. An example of RFR with 3 different regression trees

RFR has the same hyperparameter set as the regression tree, but in addition the number of regression trees is also an important hyperparameter for random forests. Normally, the more regression trees used in a random forest, the better the performance of the random forest and the longer the running time of the model. However, this can also result in overfitting (Mishra and Datta-Gupta, 2018) and needs to be checked on the test set. The hyperparameters used for the 5-folds grid search cross validation used for RFR for each model setup are depicted in Table 10.

Hyperparameters RFR	Grid
partition criteria	['mse', 'mae']
max tree depth	[2,4,6,8,10]
min samples leaf	[1,2,4]
min samples split	[2,5,10]
number of trees	[10,25,50,100,250]

Table 10. The hyperparameters set used for the 5-folds grid search cross validation for RFR

This results in 2*5*3*3*5=450 different hyperparameter sets and taking 5-folds into account, in total 5*450=2250 loops are performed for the RFR algorithm.

A small example of an RFR containing 2 regression trees with only input variable $X1_0$ is explained step by step with visualisations below.

Step 1: create 2 random subsets from the training set
In this example, the hyperparameter "number of trees" is set as two. This means that 2 different random sample sets (60%) are extracted from the total sample set (see Figure 43).

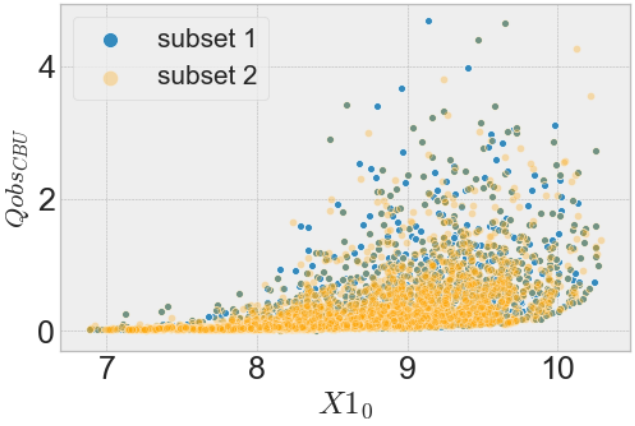


Figure 43. 2 random subsets with replacement, from the original training set with a fraction of 60%. The number of subsets is equal to the number of regression trees in the RFR.

Step 2: for each subset a regression tree is built according the theory of the DTR explained above
The regression trees of each subset are depicted in Figure 44.

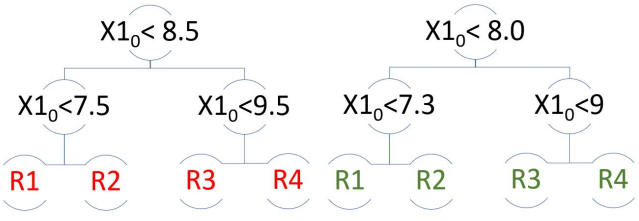


Figure 44. Each subset of Figure 43 has a specific regression tree, according to the theory of DTR

Step 3: calculate the mean Qobs in each region R for each regression tree
The regression tree of subset 1 resulting in the regions R is depicted in Figure 45, and for subset 2 in Figure 46.

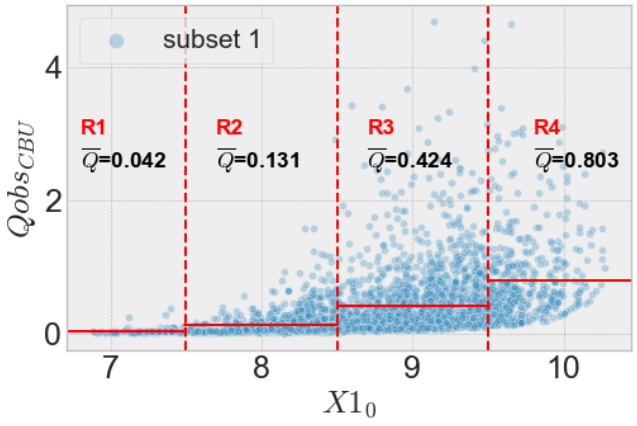


Figure 45. Visualisation of the regression tree of subset 1 with their regions R

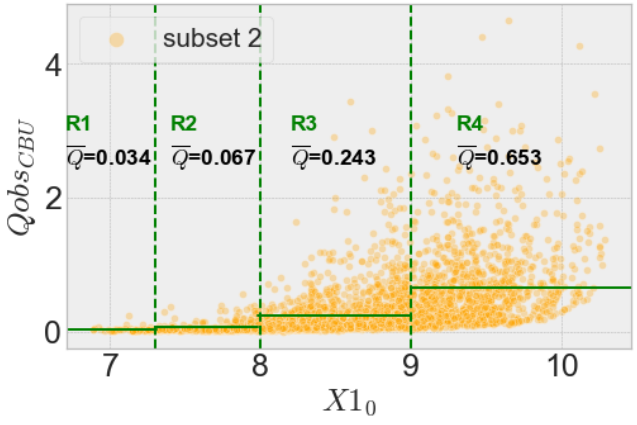


Figure 46. Visualisation of the regression tree of subset 2 with their regions R

Step 4: Derive $Qsim$ by averaging the $Qobs$ of each R
For example, a groundwater head of 8.6 $meters$ will result in 0.424 m^3/sec for the first regression tree in the RFR, and in 0.243 m^3/sec for the second regression tree. Overall, the $Qsim$ will be the average of the 2:

$$Qsim = \frac{0.424 + 0.243}{2} = 0.334 m^3/sec \quad (20)$$

Gradient Boosting Regression (GBR)
Where RFR is able to built regression trees in parallel, GBR can not. During GBR regression trees are built sequentially (Mishra and Datta-Gupta, 2018). Each new tree is constructed in such a way to compensate for the shortcomings of the previous tree (Li, 2018). GBR can be divided into 3 important steps, explained with one input variable $X1_0$ (Li, 2018):

- start with a base model $F1(X1_0)$ (one regression tree) where the model is fitted to the training data $Qobs$:

$$F1(X1_0) \cong Qobs \quad (21)$$

With this model $Qsim$ can be calculated resulting in residuals $h_1(X1_0)$, calculated as the difference between the simulated and observed discharge.

$$h_1(X1_0) = Qsim - Qobs \quad (22)$$

- fit a model to these residuals:

$$h_1(X1_0) = Qsim - Qobs = Qsim - F1(X1_0) \quad (23)$$

- create a new model (new tree) where the residuals of the previous step are taken into account:

$$F2(X1_0) = F1(X1_0) + h_1(X1_0) = F1(X1_0) + Qsim - F1(X1_0) \quad (24)$$

which results in:

$$F2(X1_0) = F1(X1_0) + h_1(X1_0) \cong Qsim \quad (25)$$

This last step shows that the second regression tree is a combination of the first regression tree and a model of the residuals of the first regression tree, and this needs to be fitted to $Qsim$. In the end, a combination of multiple regression trees is derived where the residuals need to be minimized (Li, 2018). This prinicple is also visualised in Figure 47.

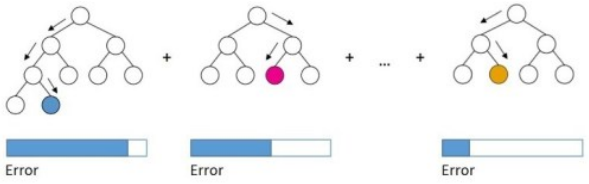


Figure 47. The principle of gradient boosting

The residuals as a function of $Qsim$ and $F(X1_0)$ can be minimized by defining a certain loss function L, such that they are equal to the negative gradient of the loss function (Li, 2018):

$$-\left(\frac{\delta L(Qsim, F1(X1_0)}{F1(X1_0)}\right) = -(F1(X1_0) - Qsim) = h_1(X1_0) \quad (26)$$

So, in each stage a regression tree is fit on the negative gradient of the given loss function. To decrease the calculation time of the GBR model, the hyperparameter "loss function" is fixed to the most common one: least squares regression, instead of using the hyperparameter within 5-folds grid search cross validation. This means that other loss functions are ignored within this research. With this given, the hyperparameters set is similar to the set for RFR (see Table 11.

Hyperparameters GBR	Grid
partition criteria	['mse', 'mae']
max tree depth	[2,4,6,8,10]
min samples leaf	[1,2,4]
min samples split	[2,5,10]
number of trees	[10,25,50,100,250]

Table 11. The hyperparameters set used for the 5-folds grid search cross validation for GBR

In common practice, GBR is a slower process than RFR since the trees are built dependently (Raschka and Mirjalli, 2017). GBR is also more sensitive to overfitting since it uses the residuals itself in the algorithm (Li, 2018).

Support Vector Regression

Support vector regression (SVR) is an extension of linear regression or polynomial regression (Paisitkriangkrai, 2012). When applying polynomial regression, a line is drawn through the predictor space such that the sum of the mean squared error of all the samples with this line is minimized (Figure 48).

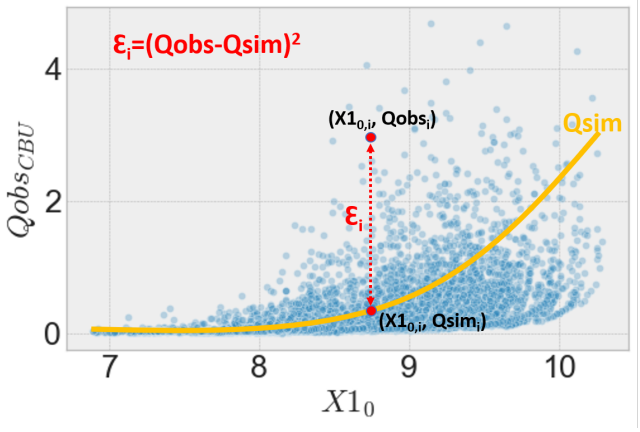


Figure 48. The principle of polynomial regression applied in the relation between input variable $X1_0$ and the stream discharge $Qobs$

For SVR a hyperplane is considered such that most of the samples fall within this hyperplane (Figure 49). The hyperplane is constructed by a line ($y = wx + b$) and adding an error ϵ and $-\epsilon$ to this line. The equations of the two boundary lines of the hyperplane become:

$$wx + b = +\epsilon \qquad \& \qquad wx + b = -\epsilon \tag{27}$$

The goal of SVR is a tradeoff between a minimization of the margin ϵ and as much of the samples inside the hyperplane as possible. In formula form:

$$-\epsilon \leq y - wx - b \geq +\epsilon \tag{28}$$

Or as Paisitkriangkrai (2012) stated: "We don't care about the error, as long as the errors are less than ϵ". As can be seen in

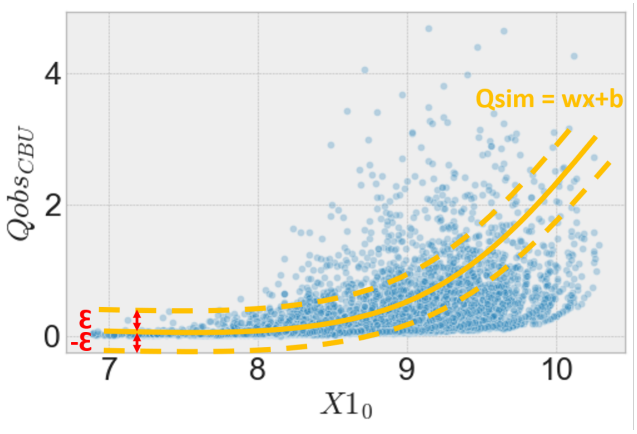


Figure 49. The principle of SVR applied in the relation between input variable $X1_0$ and the stream discharge $Qobs$

Figure 49, this problem is a non-linear case and therefore, the data is mapped into a higher dimensional space by using the Kernel trick. A common used kernel function is the Radial Basis Function (RBF) (Paisitkriangkrai, 2012), which is also used in this research:

$$K(X_1, X_2) = exponent(-\gamma||X_1 - X_2||^2) \tag{29}$$

Finding the hyperplane for a specific problem is solved by using an optimization function including a cost function. These functions require a lot of of mathematics and the in-depth description of this algorithm is outside the scope of this research.

The hyperparameters chosen to be tuned are the kernel coefficient γ of the Radial Basis function and a penalty error parameter C (as part of the optimization function). The grids for these hyperparameters used for the 5-folds grid search cross validation are depicted in Table 12.

Hyperparameters SVR	Grid
kernel coefficient of RBF (γ)	[0.001, 0.01, 0.1, 1]
Penalty error parameter (C)	[0.001, 0.01, 0.1, 1, 10]

Table 12. The hyperparameters set used for the 5-folds grid search cross validation for SVR

3.2 Method 2: Conceptual hydrological modelling

To answer the question if machine learning algorithms can be used in future researches for simulating stream discharge with groundwater heads, the outputs of these algorithms are compared with the output of an already existing model within literature: a conceptual hydrological model. A conceptual hydrological model is a precipitation-runoff model built on the observed or assumed empirical relations among different hydrological variables (Liu et al., 2017). The model conceptualises the catchment as consisting of several reservoirs with

mathematical equations describing the movement of water into and out of the reservoirs (Chiew et al., 2018). A huge difference with the machine learning models is the fact that mostly groundwater head measurements are not regarded as input variables.

3.2.1 Chosen conceptual hydrological model

The conceptual hydrological model is used in this research as a baseline for the other machine learning algorithms. In other words: "can machine learning algorithms outperform a conceptual hydrological model"? A large set of these conceptual models exist and are used within different researches, for example: Sacramento, HBV and GR4J. A very popular and simple model is the GR4J (*Genie Rural a 4 parametres Journalier*) model, which has only 4 parameters and 2 water reservoirs instead of for example 16 parameters and 5 reservoirs for the more complicated Sacramento model. Moreover, in various researches the GR4J model showed an output as good as the output of the Sacramento model (Kunnath-Poovakka and Eldho, 2019). The GR4J model also accounts for a lot of groundwater infiltration instead of direct runoff (overland flow), and hence fits the characteristics of the subcatchment Chaamse Beken as it is a catchment mostly dependent on groundwater flow. In addition, the HBV model showed in previous research (Van Loon et al., 2009) conceptual weaknesses in simulating stream discharge during dry periods , and is therefore not a good model for subcatchment Chaamse Beken. Taken all this into account, the simple GR4J model is chosen as the baseline for this study.

3.2.2 GR4J model

The GR4J model is as said before one of the simplest lumped conceptual hydrological models with a very good performance in multiple researches (Kunnath-Poovakka and Eldho, 2019). It has been used a lot for small and large catchments worldwide with a different climate: from catchments with a size of 10 km^2 in France (Le Moine et al., 2007) to 6200 km^2 in India (Kunnath-Poovakka and Eldho, 2019). The model simulates daily stream discharge data from the daily inputs rainfall and potential evaporation, all in mm/day. GR4J has two water reservoirs: *the production reservoir* and *the routing reservoir*. The production reservoir accounts for storage in the surface soil which can store rainfall and is influenced by percolation and evaporation. The second reservoir is the routing reservoir, accounting for the amount of water that can be stored in the deeper porous soil (Harlan et al., 2010). Moreover, the model accounts for groundwater exchange with surrounding catchments (Perrin et al., 2003). The model itself has only four parameters to optimize during calibration (Harlan et al., 2010):

- X1: the production reservoir maximal capacity [mm]
- X2: the catchment water exchange coefficient [mm/day]
- X3: the one-day maximal capacity of the routing reservoir [mm]
- X4: the HU1 unit hydrograph time base [days]

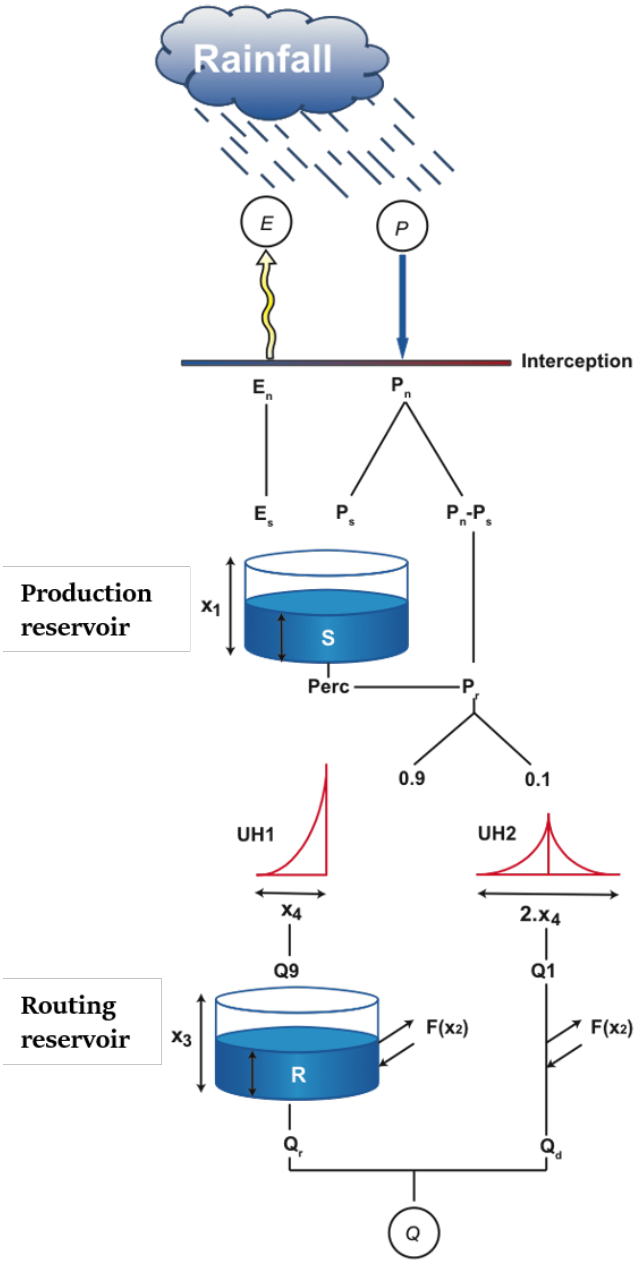


Figure 50. The conceptual hydrological **GR4J** model

The schematic version of the GR4J model is depicted in Figure 50, whereas the above-mentioned parameters are also visualised within. The working principle of this conceptual model is explained below in detail (Harlan et al., 2010), followed by the model calibration consisting of the chosen objective function and optimization algorithm.

Working principle of GR4J

The first step within the model determines either a net rainfall P_n or a net potential evaporation capacity of E_n. This computation is performed as if there were an interception reservoir of zero capacity, as the model does not account for an interception reservoir:

$$if P \geq E : P_n = P - E \qquad and \qquad E_n = 0 \tag{30a}$$

$$if P < E : P_n = 0 \qquad and \qquad E_n = E - P \tag{30b}$$

In the case $P > E$, a fraction of P (Ps) goes to the production reservoir, calculated by:

$$Ps = \frac{X1(1-(\frac{S}{X1})^2)tanh(\frac{Pn}{X1})}{1+(\frac{S}{X1})tanh(\frac{Pn}{X1})} \tag{31}$$

in which $X1$ is the first parameter to calibrate (the production reservoir maximal capacity) and S is the production reservoir level in mm. In the second case when $P < E$, a part of the evaporation is removed from the production reservoir, defined as Es:

$$Es = \frac{S(2-(\frac{S}{X1}))tanh(\frac{En}{X1})}{1+(1-\frac{S}{X1})tanh(\frac{En}{X1})} \tag{32}$$

The production reservoir level S can hence be calculated for each time step as:

$$S_{t+1} = S_t - Es_{t+1} + Ps_{t+1} \tag{33}$$

A part of the production reservoir is percolated ($Perc$) and is calculated according the following equation:

$$Perc = S\left(1-\left(1+\left(\frac{4}{9}\frac{S}{X1}\right)^4\right)^{-\frac{1}{4}}\right) \tag{34}$$

The production reservoir level is updated again via:

$$S_{t+1} = S_t - Perc_{t+1} \tag{35}$$

The production reservoir - part of the model stops here, and the routing part of the model starts from here. The water quantity that finally reaches the routing part of the model P_r consists of the part Pn not going to the production reservoir $Pn - Ps$, and the part $Perc$ from the production reservoir, in equation form:

$$P_r = (P_n - P_s) + Perc \tag{36}$$

P_r is then converted to a slow flow infiltrating into the routing reservoir/ground ($Q9$) and a fast flow that flows on the soil surface ($Q1$). 90% of P_r is routed by a unit hydrograph HU1 and a routing reservoir, while the remaining 10% is routed by the unit hydrograph HU2. The hydrographs HU1 and HU2 are dependent on the same parameter $X4$. From the unit hydrographs and the hydrographs ordinates ($UH1\&UH2$) by using S curves, the $Q9$ and $Q1$ are calculated according the following equations:

$$Q9(t) = 0.9 * \sum_{k=1}^{int(X4)+1} UH1(k) \cdot P_r(t-k+1) \tag{37a}$$

$$Q1(t) = 0.1 * \sum_{k=1}^{int(2 \cdot X4)+1} UH2(k) \cdot P_r(t-k+1) \tag{37b}$$

GR4J has the advantage to add a groundwater exchange term in the form of loss or gain ($F(X2)$) to the routing reservoir R and the fast flow $Q1$ dependent on parameters X2 and X3:

$$F = X2\left(\frac{R}{X3}\right)^{7/2} \tag{38}$$

with R the routing reservoir level, and X2 being positive for a gain, negative for a loss and zero for no groundwater exchange. The routing reservoir level (R) is then determined by $Q9$, X2 and X3 (in F) according the following equation:

$$R_t = max(0, R_t + Q9_{t+1} + F_t) \tag{39}$$

From the routing reservoir, a part is emptied in the output Qr, given by the following equation:

$$Qr = R\left(1-\left(1+\left(\frac{R}{X3}\right)^4\right)^{-\frac{1}{4}}\right) \tag{40}$$

After the determination of the Qr, the routing reservoir R is then updated by:

$$R_{t+1} = R_t - Qr_t \tag{41}$$

The part Q1 not going to the routing reservoir is added to a certain amount of exchange with other catchments, resulting in Q_d according the following equation:

$$Qd_t = max(0, Q1_t + F_t) \tag{42}$$

The total simulated stream discharge of the GR4J model $Qsim_{GR4J}$ is calculated by adding up Qr and Qd for each day.

Model calibration

For fair comparison, the GR4J model is calibrated on the same period as the training set used for machine learning algorithms (1985-1999). Moreover, the validation period is similar to the test set within machine learning (1999-2003). The model is fitted by using the following two *objective functions*: the Nash-Sutcliffe Efficiency (NSE) and the mean absolute error (MAE). In addition, the optimization algorithm "differential evolution" is used. The objective functions and the optimization algorithm are explained

below.

Objective function: NSE

The NSE function (Equation 43) has been extensively used in hydrological applications (Krause et al., 2005) and adds normalization of the variance of $Qobs$ (Krause et al., 2005). It normalizes the model performance into an interpretable dimensionless scale (Knoben et al., 2019): NSE=1 indicates perfect correspondence between $Qsim$ and $Qobs$, NSE=0 indicates that the model simulations ($Qsim$) have the same explanatory power as the mean of the observations ($\overline{Qobs}$) and NSE<0 indicates that the model is a worse predictor than $\overline{Qobs}$.

$$NSE = 1 - \frac{\sum_{i=1}^{n}(Qobs_i - Qsim_i)^2}{\sum_{i=1}^{n}(Qobs_i - \overline{Qobs})^2} \tag{43}$$

Despite the fact that the NSE is often used as a metric for the level of overall agreement between the observed and simulated values, it is sensitive to peak flows due to the use of squared deviations, making it less suited to low flow simulation (Krause et al., 2005). In other words, it better fits the peaks than the low flows. Therefore, the other basic objective function MAE is also used in this research.

Objective function: MAE

The MAE function (Equation 44) records in real units the level of overall agreement between $Qobs$ and $Qsim$ (Dawson et al., 2007). It is a non-negative metric that has no upper bound and for a perfect model the result would be zero. Moreover, it is not weighted towards high or low flows. This is an important difference with the NSE.

$$MAE = \frac{\sum_{i=1}^{n} \mid Qobs_i - Qsim_i \mid}{n} \tag{44}$$

Note that MAE and NSE do not provide any information about under- or overestimation (Dawson et al., 2007).

Optimization algorithm: differential evolution

To get the best set of parameters, an *optimization algorithm* is performed for each objective function. The global optimization algorithm "differential evolution" is applied since it has been used before for the calibration of GR4J models in previous researches (Shin and Kim, 2019). This optimization algorithm requires two inputs, namely the objective function to be minimized and bounds for the model parameters. Useful bounds for the parameters are given by Perrin et al. (2003), based on the 80% confidence intervals of the distribution of model parameters obtained over a large sample of catchments:

	Median value	Bounds
X1 [mm]	350	100-1200
X2 [mm/day]	0	-5 to 3
X3 [mm]	90	20-300
X4 [days]	1.7	1.1 - 2.9

Table 13. Bounds of the GR4J model parameters used for the "differential evolution" optimization algorithm

After optimization of the objective functions during the calibration period, the optimized parameters are used again for the validation period to assess the model performance.

3.3 Evaluation metrics

In this section, the metrics used for model performance are described. They are used for the machine learning algorithms for the training and test set and for the GR4J model in the calibration and validation period. The evaluation metrics are divided into 3 categories: overall model performance, model performance for peak flows and model performance for low flows.

3.3.1 Evaluation metrics - overall model performance

The evaluation metrics used in this research for the overall model performance are NSE and MAE. These evaluation metrics are already discussed in section 3.2.2 at "Model calibration" (Equation 43 and 44).

3.3.2 Evaluation metric - model performance peak flows

A common evaluation metric to emphasize the peak flows, is the fourth root mean quadrupled error (R4MS4E) (Dawson et al., 2007) given in Equation 45. This metric records in real units the level of overall agreement between the observed and simulated discharge. It is a non-negative metric that has no upper bound and for a perfect model the result would be zero. Due to the fourth power, more emphasis is put on the peak flows, since in most cases the larger errors occur during peak flow. Therefore, the R4MS4E is less insensitive to errors during low flows.

$$R4MS4E = \sqrt[4]{\frac{\sum_{i=1}^{n}(Qobs_i - Qsim_i)^4}{n}} \tag{45}$$

3.3.3 Evaluation metric - model performance low flows

Somewhat complementary to the R4MS4E for peak flows, is the mean squared logarithmic error (MSLE) (Equation 46) for low flows (Dawson et al., 2007). Due to the logarithmic transformations involved in its computation, MSLE is a preferred measure for evaluating the model performance for low flows. MSLE is non-negative and for a perfect model the result would be zero. Note that this metric can not be easily

converted to MSE due to the logarithmic transformation, and can therefore also not be easily made visible in the same unit as the stream discharge (m^3/sec).

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(ln(Qobs_i) - ln(Qsim_i))^2 \quad (46)$$

4 Results, Discussion & Limitations

In this chapter, first of all an extensive discussion is given of the results of the machine learning algorithms. Afterwards, the results of the GR4J model are given. Furthermore, in the last section the GR4J model is compared with the best machine learning algorithm(s), to determine if machine learning models equal the conceptual hydrological model such that it can be used for future purposes. In addition to the results and discussion, some limitations are given.

4.1 Results, discussion and limitations - Machine learning

This section covers the results of the different machine learning algorithms performed with their different model setups for research questions 1a to 1d, as previously described in section 3.1.2. In addition, an overall comparison is given for all model setups and all machine learning algorithms (within this research) to see which combination of model setup and algorithm results in the best model performance.

4.1.1 Results, discussion and limitations - research question 1a

This subsection covers the results of the approach defined for research question 1a, in which the chosen machine learning algorithms are performed with model setup 1 and 2, and are compared. For a short recap, the model setups are depicted below:

Model setup 1:

$$Y(t) = F(X1_0(t)) \quad (47)$$

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \quad (48)$$

The results itself of the algorithms by using model setup 1 are collected in Appendix C, consisting of a total dataset overview in C1, plots of the $Qsim$ time series in appendix C2, the optimal hyperparameters for each algorithm in appendix C3 and the regression trees of RFR in appendix C4. The same applies for the results of model setup 2 in Appendix D, except for the regression trees of RFR as they become to large to visualise. Therefore, in Appendix D4 the single regression tree of DTR is only visualised.

On the left of Figure 51, scatterplots of $Qobs$ versus $Qsim_1$ for the test set for model setup 1 are visualised. The right part of 51 shows scatterplots of $Qobs$ versus $Qsim_2$ for the test set for model setup 2. This is done for the four different machine learning algorithms DTR, RFR, GBR and SVR. The results are discussed in the next subsection, followed by a subsection giving a short overview of the limitations of this approach.

4.1.1.1 Results and discussion - research question 1a

The results of model setup 1 show that none of these machine learning algorithms is able to simulate $Qobs$. This means that simulating $Qobs$ with only well $X1_0$ by using machine learning algorithms DTR, RFR, GBR and SVR is not possible. However, the results of model setup 1 do show that $Qobs$ can be simulated better for low flows (<1 m^3/sec) than for peak flows. The evaluation metric R4MS4E for peak flows shows values larger than 0.80 m^3/sec, while a value of 0 m^3/sec means a perfect model performance. This is also visible in the $Qsim$ time series of the training and test set depicted in Figure 52. This figure also shows worse model performance for both the training and test set. It can hence be concluded that underfitting occurs for all machine learning algorithms by using model setup 1.

On the other hand, model setup 2 shows already much better results when simulating $Qobs$. Especially the algorithms RFR, GBR and SVR are able to simulate $Qobs$ with a mean absolute error of 0.15-0.17 $(m^3/sec)^2$. In terms of the NSE, this means an NSE of 0.60-0.70. According to various hydrological models (Cheng et al., 2017) a NSE smaller than 0.5 is rated as unsatisfactory, a NSE between 0.50 and 0.65 as satisfactory, a NSE between 0.50 and 0.65 as good and a NSE between 0.75 and 1.0 as very good. This means that using model setup 2 for the machine learning algorithms RFR, GBR and SVR results in a good model performance. In other words, the groundwater head time series of all screen-1 wells $X1_0$-$X1_5$ can be used as input variables for machine learning algorithms RFR, GBR and SVR to simulate stream discharge, when considering the overall model performance.

Furthermore, note that the algorithms DTR, RFR and GBR results in a less smooth $Qsim$ than when using the algorithm SVR. This is due to the fact that DTR, RFR and GBR consists of one or more regression trees where the outputs are discontinuous, whereas for SVR the output has a more smooth output by nature. This is visible when zooming in on a part of the test set in Figure 53. Note that using a regression tree with a larger tree depth, results in more leaves and hence a more smooth line (Mishra and Datta-Gupta, 2018).

An advantage of the algorithm RFR is its ability to give the relative importance of each input variable. The relative importance represents how much including a certain variable improves the simulation (Raschka and Mirjalli, 2017). Therefore, random forests are also often used to identify the most important variables, especially when the number of variables is larger than the number of samples (Behnamian et al., 2017). In this way, the variables set can be reduced for other machine learning algorithms.

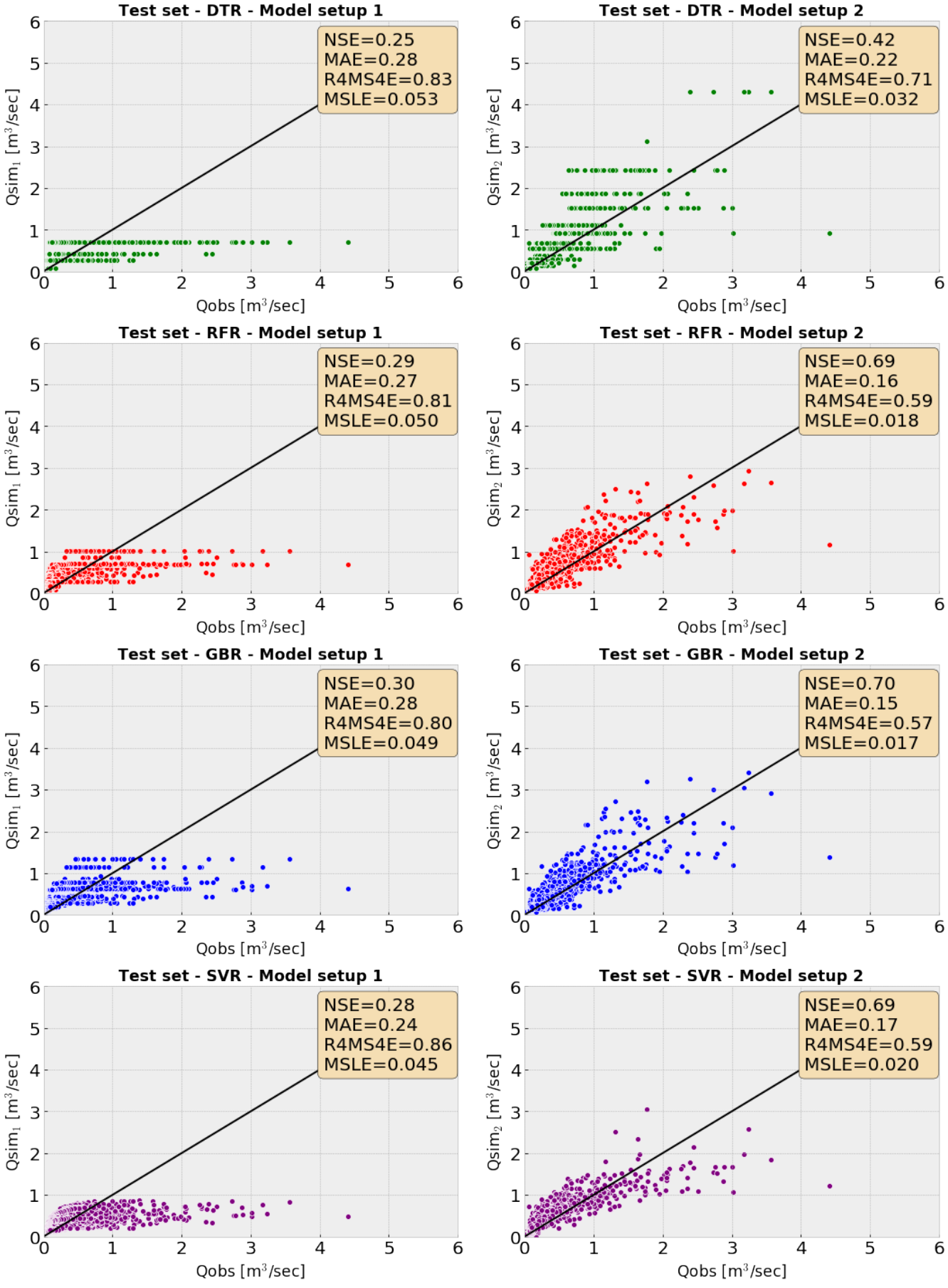


Figure 51. Test Results for research question 1a, including model setup 1 and model setup 2 for machine learning algorithms DTR, RFR, GBR and SVR

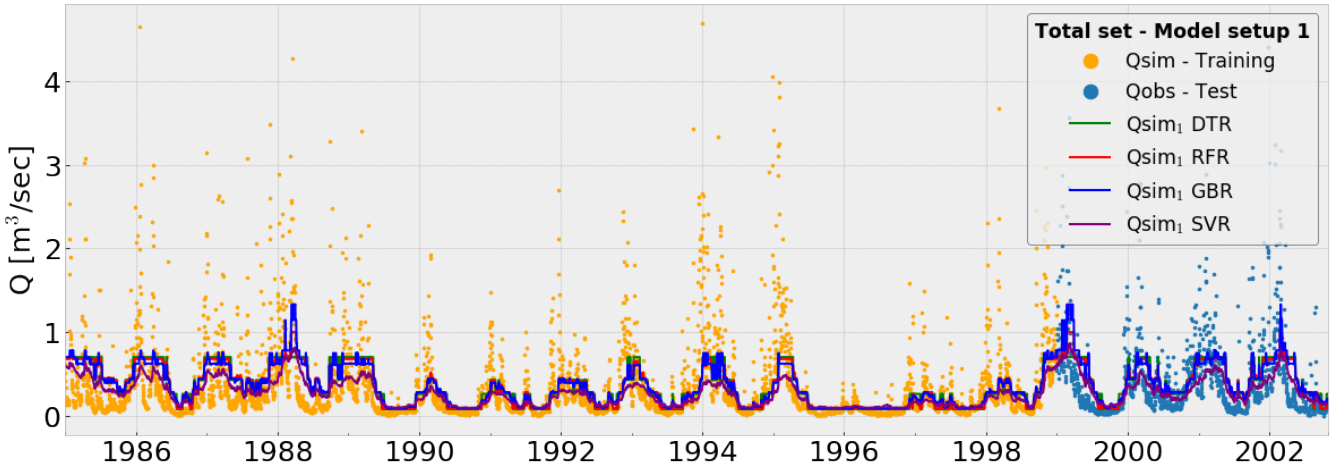


Figure 52. $Qsim_1$ time series of the 4 different machine learning algorithms by using model setup 1, including training and test set

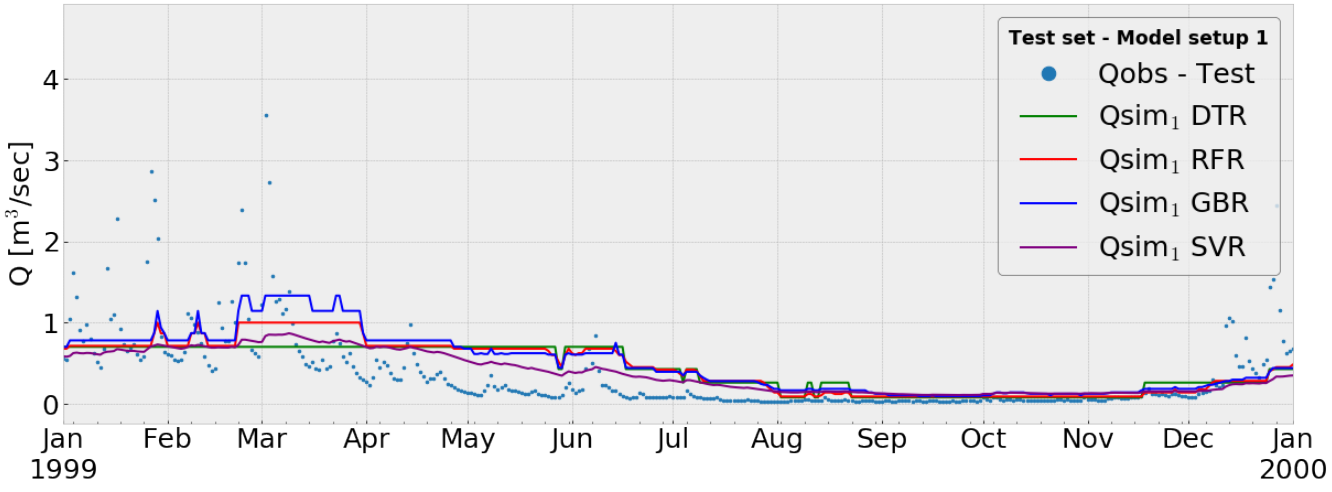


Figure 53. $Qsim_1$ time series of the 4 different machine learning algorithms by using model setup 1, zooming in on a part of the test set

For model setup 1 there is only 1 input variable and hence the relative importance for this input variable is 100%. For model setup 2 on the other hand there are 6 input variables ($X1_0$-$X1_5$) and the relative importance for well $X1_4$ is the largest, followed by low values for well $X1_1$, well $X1_0$ and well $X1_2$, as can be seen in Figure 54. The well with the largest relative importance has also the largest correlation with the target $Qobs$ as can be seen in the correlation heatmap of the dataset for model setup 2 in Figure D3 in Appendix D1.3. It is therefore logical that this well has the largest relative importance, as it is able to capture the variance of the target the most. Furthermore, note that this well is also the well in absence of the clay layer of Stamproy in its geological layers (see Figure 20). This could indicate that this specific well ($X1_4$) can be the one that is mostly responsible for the stream discharge at CBU. Note that input variables with the largest relative importance are mostly the input variables at the top of the regression trees within a random forest (Behnamian et al., 2017).

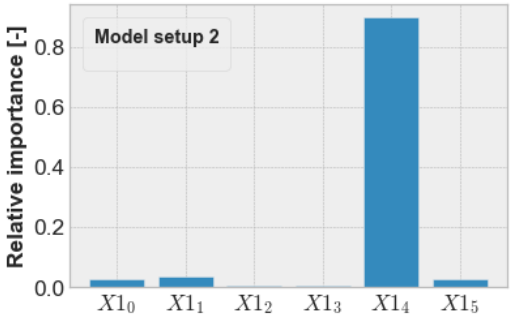


Figure 54. The relative importance of the input variables of model setup 2 when using RFR

Overall, it can be concluded that simulating $Qobs$ is not possible if only using well $X1_0$ as an input variable for all four machine learning algorithms. This is as expected with only one input variable in the machine learning algorithms. Using all screen-1 wells ($X1_0$-$X1_5$) as input variables instead of just well $X1_0$, is increasing the overall model performance a lot for especially algorithms RFR, GBR and SVR. Peak flows however, are still drastically underestimated for RFR, GBR and SVR. DTR has the worse model performance in model setup 2, as it over- and underestimates peak flows.

4.1.1.2 Limitations - research question 1a

The first limitation of this approach is the number of hyperparameters. The to be tuned hyperparameters in this research are limited to a pre-defined set of 4 hyperparameters for DTR, 5 hyperparameters for RFR, 5 hyperparameters for the GBR and only 2 hyperparameters for SVR. Note that these machine algorithms have more hyperparameters to tune, but they are not taken into account in this research to avoid very long computation times. For the hyperparameters not taken into account for tuning, the default value is chosen in its programming code.

Furthermore, in model setup 1 the well $X1_0$ is chosen as the most representative one of all the screen-1 wells, because of the fact that it has the largest correlations with the other screen-1 wells. In an additional research the screen-1 well $X1_4$ can be chosen for this model setup, since it has the largest correlation with the target $Qobs$, as can be seen in Figure D3 of Appendix D2. Moreover, this well has the largest relative importance, so it is interesting to examine if it is possible to simulate $Qobs$ based on only this well.

4.1.2 Results, discussion and limitations - research question 1b

This subsection covers the results of the approach defined for research question 1b, in which the chosen machine learning algorithms are performed with model setup 2 and 3, and are compared. For a short recap, the model setups are depicted below:

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \quad (49)$$

Model setup 3:

$$Y(t) = F(X1_0(t), ...X1_5(t) \& X2_0(t), ...X2_5(t)) \quad (50)$$

The results itself of the algorithms by using model setup 3 are collected in Appendix E, consisting of a total dataset overview in E1, plots of the $Qsim$ time series in appendix E2, the optimal hyperparameters for each algorithm in appendix E3 and the single regression tree of DTR in E4.

On the left of Figure 55, scatterplots of $Qobs$ versus $Qsim_2$ for the test set for model setup 2 are visualised. The right part of 55 shows scatterplots of $Qobs$ versus $Qsim_3$ for the test set for model setup 3. This is done for the four different machine learning algorithms DTR, RFR, GBR and SVR. The results are discussed in the next subsection, followed by a subsection giving a short overview of the limitations of this approach.

4.1.2.1 Results and discussion - research question 1b

The results of model setup 3 show that including screen-2 wells in the input variables set of model setup 2 do not yield better model performance for RFR, GBR and SVR. For RFR, GBR and SVR the NSE does barely increase or decrease from model setup 2 to 3. The NSE can still be seen as a satisfactory or good overall model performance. This means that screen-2 wells do not add additional variance of the $Qobs$ in the models. The $Qsim$ time series of SVR of model setup 2 and model setup 3 are plotted for a part of the test set in Figure 57. It can be seen that the peaks are more underestimated when also using screen-2 wells.

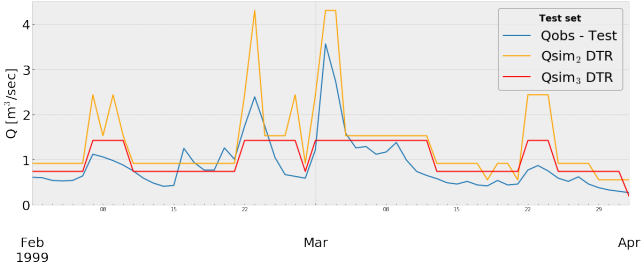


Figure 56. $Qsim_2$ and $Qsim_3$ time series of the machine learning algorithm DTR by using model setup 2 and 3, zooming in on a part of the test set

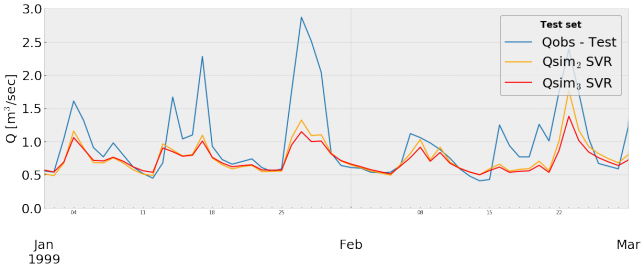


Figure 57. $Qsim_2$ and $Qsim_3$ time series of the machine learning algorithm SVR by using model setup 2 and 3, zooming in on a part of the test set

On the other hand, when using the machine learning algorithm DTR, the overall model performance is increasing a bit: going from a NSE of 0.42 to 0.59 m^3/sec. The increasing model performance of DTR is also visualised when plotting the $Qsim$ time series (Figure 56) of model setup 2 and 3. Adding screen-2 wells to the input variables set of algorithm DTR results in less overestimated peak flows. For DTR it can be stated that adding screen-2 wells will result in a bet-

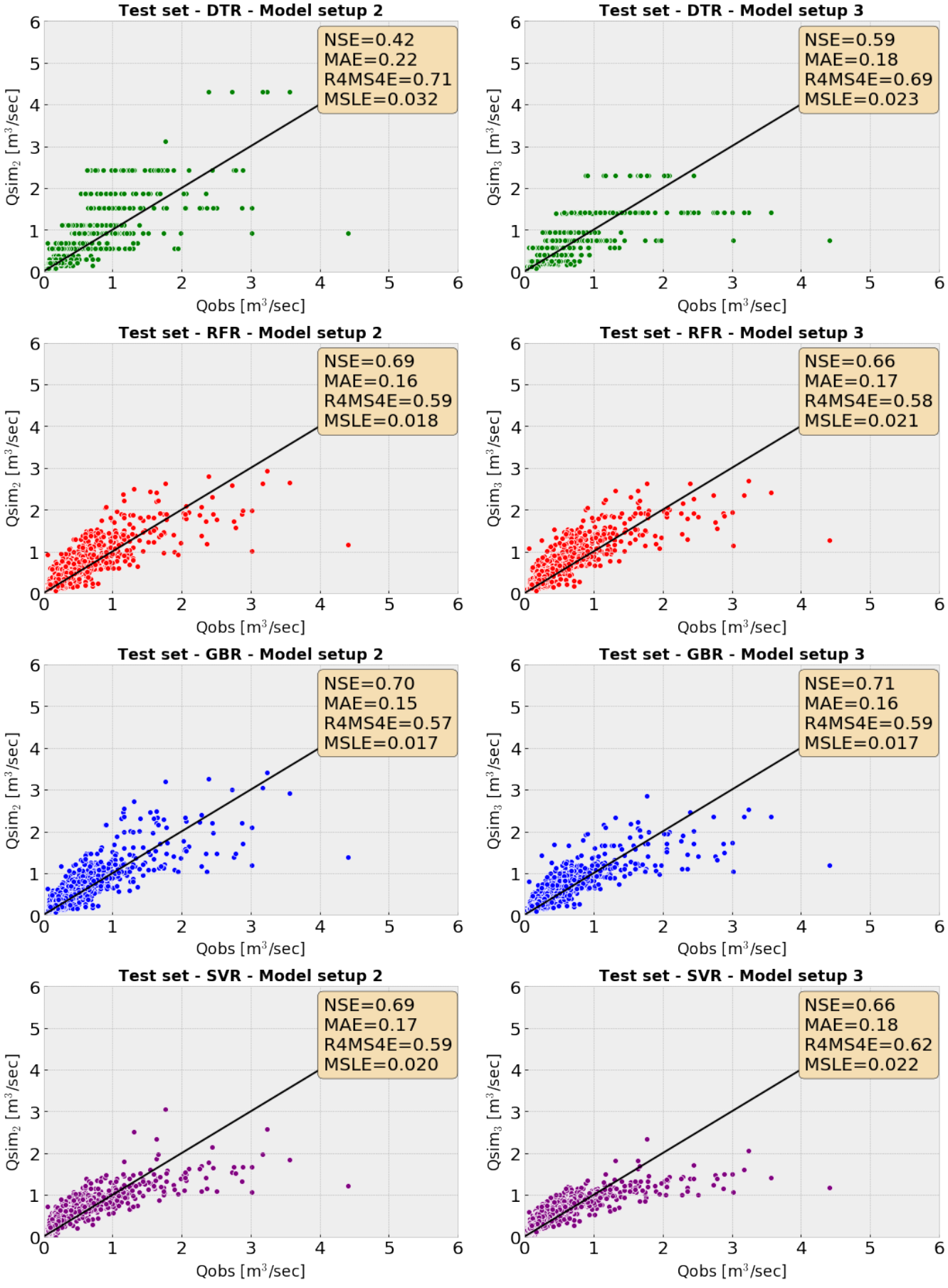


Figure 55. Test Results for research question 1b, including model setup 2 and model setup 3 for machine learning algorithms DTR, RFR, GBR and SVR

ter overall model performance, but the model performance is still not that comparable with RFR, GBR and SVR.
Taking the relative importances of the input variables of model setup 3 into account (see Figure 58, it becomes also visible that screen-2 wells do not play an important role in comparison with screen-1 wells. Only well $X2_1$ has a very small relative importance, which can be neglected. Again well $X1_4$ has by far the largest relative importance and can not be decreased by adding screen-2 wells.

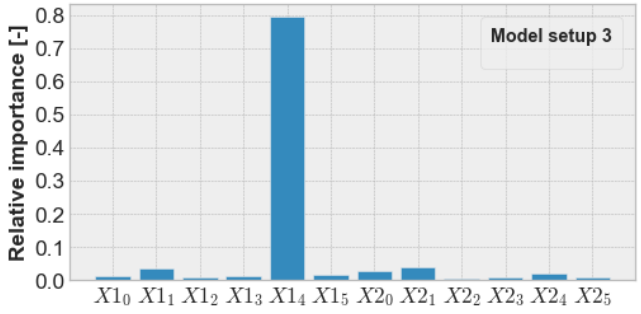


Figure 58. The relative importance of the input variables of model setup 3 when using RFR

As said in the introduction of this research the groundwater flow and the interflow are responsible for the baseflow (Bosch et al., 2017). It is expected that adding more and deeper wells (screen-2 wells), would result in a better simulation of the low flows. However, the evaluation metric MSLE for low flows shows that the MSLE is not improving for all four machine learning algorithms when adding screen-2 wells to the input variables set of screen-1 wells. This could indicate that screen-2 wells are not responsible for the baseflow production in the Chaamse Beken.

Overall, it can be concluded that adding screen-2 wells to the input variables set consisting of screen-1 wells do not or barely improve the overall model performance for RFR, GBR and SVR. Only for DTR, the model performance is increasing significantly, but is still not that good enough as the model performances of RFR, GBR and SVR for model setup 2. Moreover, adding screen-2 wells as input variables do not result in a better simulation of low flows. Lastly, peak flows can not be simulated with only screen-1 and screen-2 wells. This is as expected, since peak flows are mostly caused by overland flow due to heavy precipitation (Steenbergen and Willems, 2012). Precipitation is absent in the input variables set of model setup 2 and 3 and therefore, peak flows can not be simulated with model setup 2 and 3.

4.1.2.2 Limitations - research question 1b

The first limitation of this approach is again the number of hyperparameters, which is already discussed in subsection 4.1.1.2.. The same applies for this approach again. Moreover, in this approach only screen-1 and screen-2 wells are examined, but it would be interesting to examine what happens with the model performance when adding deeper wells (for instance screen-3 wells).

4.1.3 Results, discussion and limitations - research question 1c

This subsection covers the results of the approach defined for research question 1c, in which the chosen machine learning algorithms are performed with model setup 2 and 4, and are compared. For a short recap, the model setups are depicted below:

Model setup 2:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t)) \tag{51}$$

Model setup 4:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t)) \tag{52}$$

The results itself of the algorithms by using model setup 4 are collected in Appendix F, consisting of a total dataset overview in F1, plots of the $Qsim$ time series in appendix F2, the optimal hyperparameters for each algorithm in appendix F3 and the single regression tree of DTR in appendix F4.

On the left of Figure 59, scatterplots of $Qobs$ versus $Qsim_2$ for the test set for model setup 2 are visualised. The right part of 59 shows scatterplots of $Qobs$ versus $Qsim_4$ for the test set for model setup 4. This is done for the four different machine learning algorithms DTR, RFR, GBR and SVR. The results are discussed in the next subsection, followed by a subsection giving a short overview of the limitations of this approach.

4.1.3.1 Results and discussion - research question 1c

The results of this approach show that including P and Ep in the input variable set of model setup 2 do not improve the overall model performance of algorithms RFR and GBR. For DTR and SVR on the other hand, the overall model performance increases a bit. Especially, the overall model performance of SVR increased to a level that this model can be seen as usable for simulating $Qobs$. It has namely a NSE of 0.75, wich means that the model performance can be rated as very good according to Cheng et al. (2017). Moreover, the MAE is 0.15 m^3/sec. The combination of the NSE and MAE values is the best one so far for a certain model setup and a specific algorithm. It can be seen that the peak flows for SVR of model setup 4 are now also overestimated for some samples. For model setup 2 by using SVR, most of the peaks were underestimated. This is also visible in Figure 60.

Furthermore, it is remarkable to see that for machine learning algorithms RFR, GBR and SVR and using model setup 4, the peak flows are not better simulated than when using model setup 2. The evaluation metric R4MS4E for peak flows increases to higher values, while lower values are

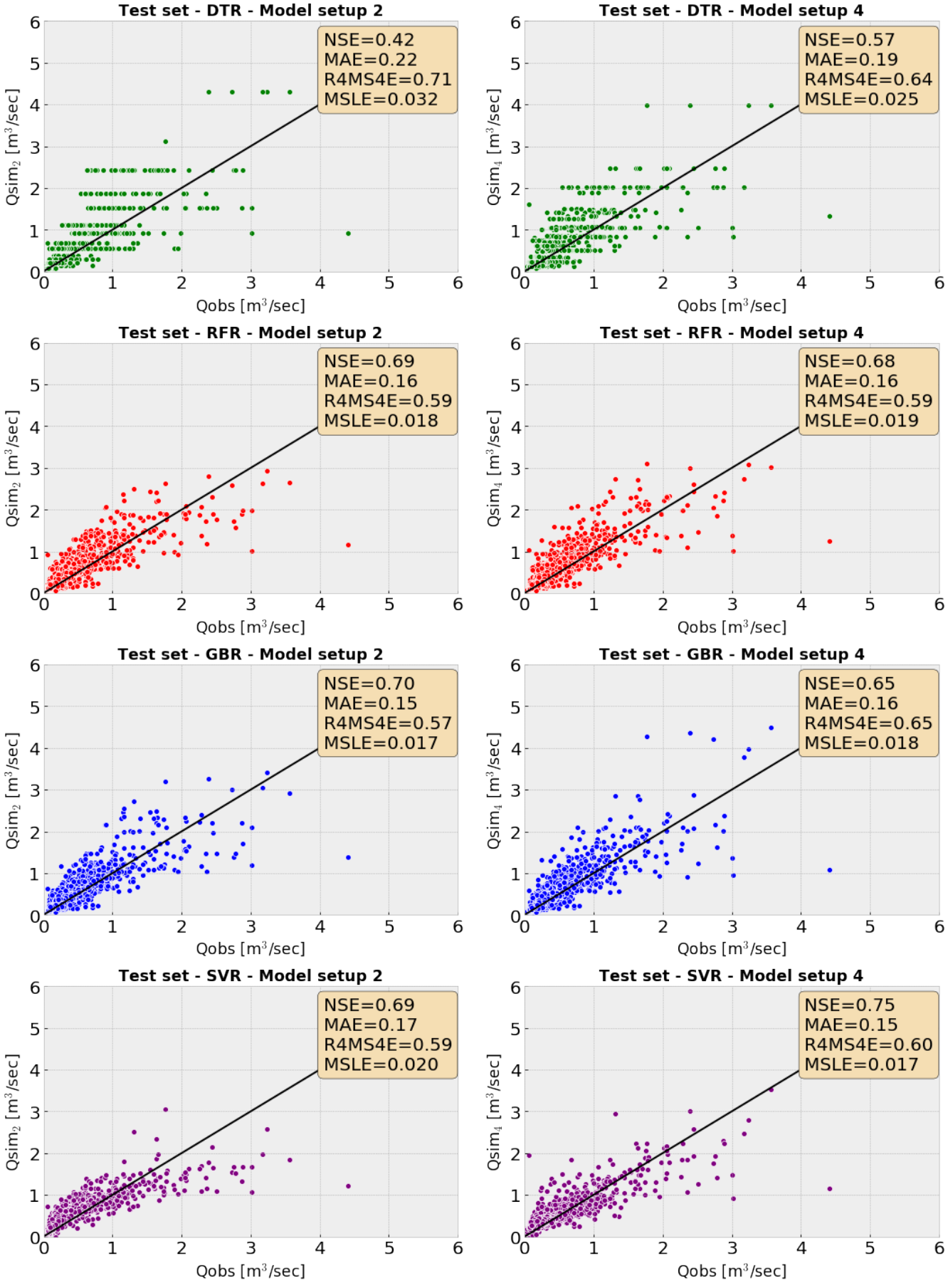


Figure 59. Test Results for research question 1c, including model setup 2 and model setup 4 for machine learning algorithms DTR, RFR, GBR and SVR

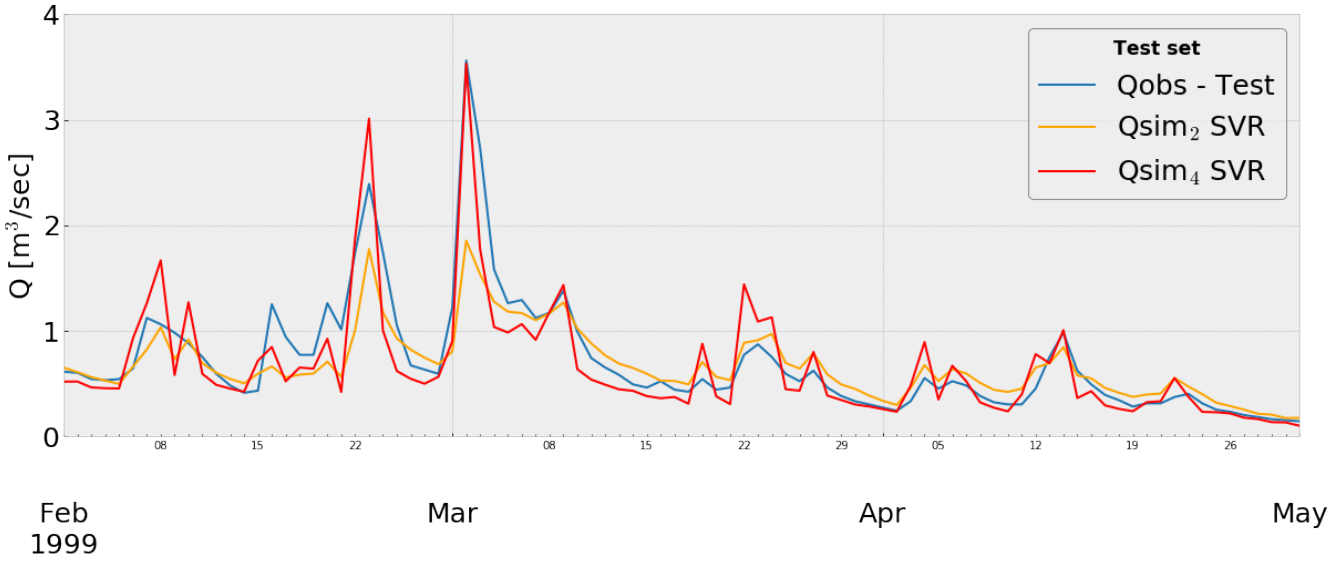


Figure 60. $Qsim_2$ and $Qsim_4$ time series of the machine learning algorithm SVR by using model setup 2 and 4, zooming in on a part of the test set

wanted. It was expected that the peak flows could be better simulated with P and Ep in the input variables set. Hence, this hypothesis can be refuted. Only for the algorithm DTR, the peak flows are better simulated when using model setup 4 instead of model setup 2. A possible reason for the disability of simulating peak flows with P and Ep in the input variables set, is the fact that these peak flows are subject to delays. A precipitation event at timestep t does not have to result in a peak flow at the same timestep t. The phenomena of delays will be examined in the next section of results.

Lastly when taking only low flows into account, adding P and Ep to the input variables set of model setup 2 does not hugely change the low flow model performance (MSLE stays approximately the same). This is due to the fact that P is mostly absent during low flows (Bosch et al., 2017) and can therefore also not hugely change the low flow model performance.

Taking the relative importances of the input variables of model setup 4 into account (Figure 61), it can be seen that P has the second largest relative importance, but is still small compared to the relative importance of well $X1_4$.

Apparently, P is important for simulating peak flows, but they are still not very well simulated (still over- and underestimation). Overall, it can be concluded that for simulating base flow of $Qobs$, the models with model setup 2 and 4 do not differ that much. However, when taking the total time series of $Qobs$ into account, the best model performance is obtained for SVR with model setup 4.

4.1.3.2 Limitations - research question 1c

The first limitation of this approach is again the number of hyperparameters. Moreover, in this approach $Qobs$ is not

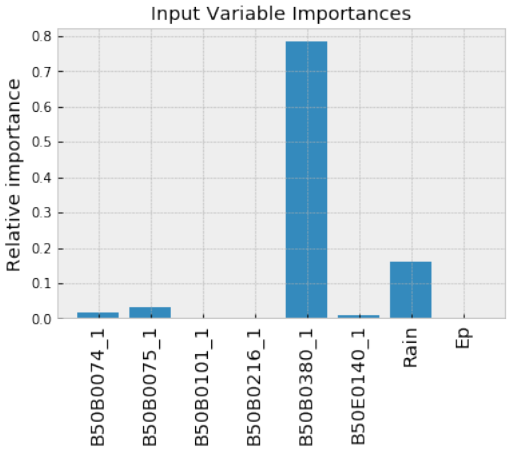


Figure 61. The relative importance of the input variables of model setup 4 when using RFR

simulated based on only the input variables P and Ep. It is recommended in additional research to simulate $Qobs$ with only P and Ep, to see if peak flows can be better simulated. It is expected however, that low flows will not be simulated well enough, as they depend on groundwater flow.

4.1.4 Results, discussion and limitations - research question 1d

This subsection covers the results of the approach defined for research question 1d, in which the chosen machine learning algorithms are performed with model setup 4 and 5, and are compared. For a short recap, the model setups are depicted

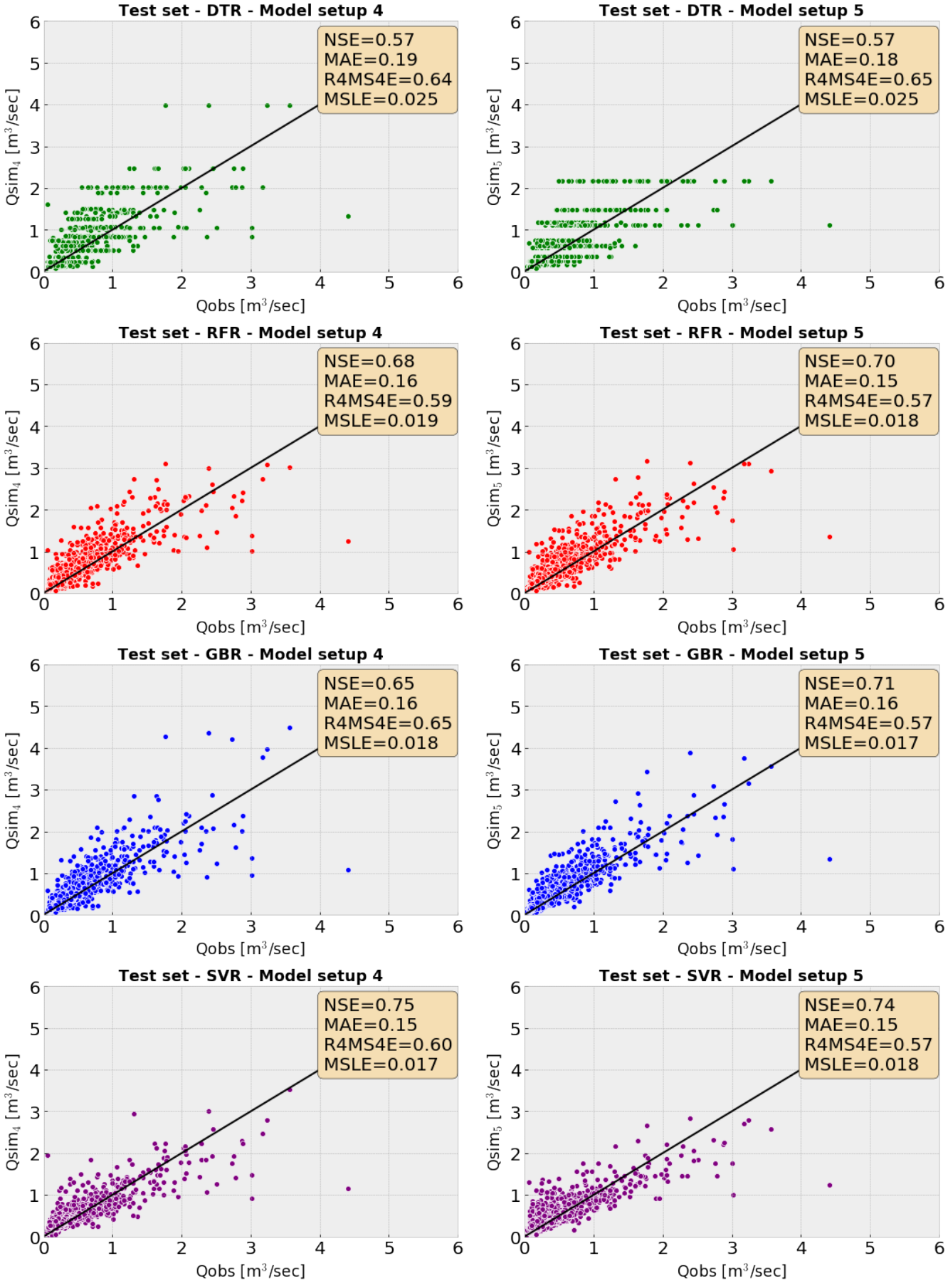


Figure 62. Test Results for research question 1d, including model setup 4 and model setup 5 for machine learning algorithms DTR, RFR, GBR and SVR

below:

Model setup 4:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t)) \quad (53)$$

Model setup 5:

$$Y(t) = F(X1_0(t), X1_1(t), ...X1_5(t), P(t), Ep(t), Mand\Delta t) \quad (54)$$

The results itself of the algorithms by using model setup 5 are collected in Appendix G, consisting of a total dataset overview in G1, plots of the $Qsim$ time series in appendix G2, the optimal hyperparameters for each algorithm in appendix G3 and the single regression tree of DTR in Appendix G4.

On the left of Figure 62, scatterplots of $Qobs$ versus $Qsim_4$ for the test set for model setup 4 are visualised. The right part of 62 shows scatterplots of $Qobs$ versus $Qsim_5$ for the test set for model setup 5. This is done for the four different machine learning algorithms DTR, RFR, GBR and SVR. Note that for RFR and GBR not all the input variables are used, but only the 10 with the largest relative importance (as in Figure G11) to reduce the computation time. The results are discussed in the next subsection, followed by a subsection giving a short overview of the limitations of this approach.

4.1.4.1 Results and discussion - research question 1d

The results of this approach show that adding memory and delays to the system (model setup 5) seems not to change a lot in the model performance for SVR and DTR, in comparison with model setup 4. The evaluation metrics stay approximately equal for model setup 4 and 5. Therefore, it can be concluded that adding memory and delays to the input variable set, does not have an effect on the model performance for SVR and DTR. For the algorithms RFR and GBR, the MAE does barely change, while the NSE shows a slight increase. The NSE of the RFR and GBR when using model setup 5 shows the best value so far and hence the best overall model performance for these algorithms. Apparently, the best overall model performance for algorithms RFR and GBR is with model setup 5: memory and delays are needed in the input variables set.

4.1.4.2 Limitations - research question 1d

The first limitation of this approach is again the number of hyperparameters. Moreover, the number of days taken into account for delays into the input variable set is restricted to 6 days for P and Ep. Furthermore for adding the M of the system, the rolling means are restricted to maximum 120 days and only the values 3,7,14,30 and 120 days are taken into account, but no values between them. Lastly, it is known that groundwater can stay for multiple years in the groundwater reservoir and therefore rolling means of longer than 120 days should be considered in additional research.

4.1.5 Overall comparison machine learning algorithms

In this section an overview of the model performances expressed in the evaluation metrics for DTR, RFR, GBR and SVR is depicted in Tables 14 to 18, taking into account the multiple model setups.

As can be seen in Table 14, using model setup 1 has a worse overall model performance for all machine learning algorithms. The NSE values are not larger than 0.30. Also, the peak and low flow model performances are not optimal, when considering the R4MS4E and MSLE values.

For machine learning algorithm DTR, model setup 3 to 5 can be chosen as the best model setups to simulate $Qobs$. However, the other machine learning algorithms perform clearly better when using model setup 2 to 5. Therefore, it can be concluded that DTR is not a very good machine learning algorithm to simulate $Qobs$ in this research.

For RFR, the best overall model performance is when using model setup 5 as can be seen in Table 18 (NSE of 0.70). For GBR, the best overall model performance is when using model setup 3 or 5 (NSE of 0.71). And for SVR, when using model setup 4 (Table 17). The NSE has a value of 0.75 and hence the model can be rated as very good, according to Cheng et al. (2017). It can be concluded that the best overall model performance is reached when using the algorithm SVR and model setup 4. This means an input variables set of the screen-1 wells $X1_0$-$X1_5$, P and Ep.

Therefore, it would be logical to choose this algorithm and model setup for the comparison with the conceptual GR4J model. However, the goal of this research is to simulate $Qobs$, but focused on the baseflow (low flows). When simulating the baseflow more accurately, the ecology-improving measures can also be more accurately, quantitatively examined. It is more important for this research to choose the combination of algorithm and model setup that has the lowest MSLE value.

The lowest MSLE value is for the combination of algorithm GBR and model setup 2. The MSLE has the lowest rounded value of 0.017 $\ln^2(m^3/sec)$. However, the MSLE value of algorithm SVR and model setup 4 does not differ that much from algorithm GBR and model setup 2. And since the overall model performance of SVR when using model setup 4 is better, this combination is again chosen as the best machine learning model for this research. Furthermore, note that the SVR has a much shorter computation time than the GBR algorithm, which is a huge advantage. Summarized, the SVR algorithm when using model setup 4 will be compared with the GR4J model in this research.

Lastly, it must be stated that when peak flows are the most important flows to model, none of these algorithms and model setups result in a very good model performance. The lowest R4MS4E value is 0.57 m^3/sec for algorithms RFR, GBR and SVR when using model setup 5.

Model setup 1	DTR	RFR	GBR	SVR
NSE	0.25	0.29	0.30	0.28
MAE	0.28	0.27	0.28	0.24
R4MS4E	0.83	0.81	0.80	0.86
MSLE	0.053	0.050	0.049	0.045

Table 14. Evaluation metrics for machine learning algorithms using model setup 1

Model setup 2	DTR	RFR	GBR	SVR
NSE	0.42	0.69	0.70	0.69
MAE	0.22	0.16	0.15	0.17
R4MS4E	0.71	0.59	0.57	0.59
MSLE	0.032	0.018	0.017	0.020

Table 15. Evaluation metrics for machine learning algorithms using model setup 2

Model setup 3	DTR	RFR	GBR	SVR
NSE	0.59	0.66	0.71	0.66
MAE	0.18	0.17	0.16	0.18
R4MS4E	0.69	0.58	0.59	0.62
MSLE	0.023	0.021	0.017	0.022

Table 16. Evaluation metrics for machine learning algorithms using model setup 3

Model setup 4	DTR	RFR	GBR	SVR
NSE	0.57	0.68	0.65	0.75
MAE	0.19	0.16	0.16	0.15
R4MS4E	0.64	0.59	0.65	0.60
MSLE	0.025	0.019	0.018	0.017

Table 17. Evaluation metrics for machine learning algorithms using model setup 4

Model setup 5	DTR	RFR	GBR	SVR
NSE	0.57	0.70	0.71	0.74
MAE	0.18	0.15	0.16	0.15
R4MS4E	0.65	0.57	0.57	0.57
MSLE	0.025	0.018	0.017	0.018

Table 18. Evaluation metrics for machine learning algorithms using model setup 5

4.2 Results, discussion and limitations - GR4J model

This section covers the results of the GR4J model, in which the two different objective functions are used for calibration, as described in section 3.2.2 of the chapter "Methods". Results of the GR4J model for using each single objective function are more extensively visualised in Appendix H.

4.2.1 Results and discussion - GR4J model

The results of the GR4J model for the two different objective functions for the validation set are depicted in Figure 63. At first glance, it seems that these models with different objective functions have almost a similar output. The evaluation metrics of the overall model performance show also similar values. However, when zooming in on peak flows, the GR4J model calibrated with the objective function MAE gives slightly higher values for $Qsim$ than the GR4J model calibrated with the NSE function. This is also visible in Figure 64 when zooming in on the $Qsim$ time series of the validation period. The model with the objective function NSE can better simulate $Qobs$ for peak flows. This is also as expected, since models with objective functions NSE focus more on fitting peak flows than the other objective function MAE does (Buzacott et al., 2019). However, both models still do not perform optimal for simulating peak flows: the R4MS4E values of both models are still larger than 0.5 m^3/sec.

On the other hand, zooming in on dry periods where the baseflow becomes more dominant (Figure 65), it becomes obvious that the model with objective function MAE is the one that has the best model performance in simulating $Qsim$ during baseflow conditions. The MSLE value of this model is also the lowest: 0.0084 $\ln^2(m^3/sec)$. Therefore, this model is chosen as the baseline model for this research. The optimized parameters of the GR4J model are as follows: a production store maximal capacity ($X1$) of 456 mm, a catchment water exchange coefficient ($X2$) of -5 mm/day (loss to other catchments), an one-day maximal capacity of the routing reservoir ($X3$) of 27 mm and the HU1 unit hydrograph time base ($X4$) is 1.1 days. Especially, parameter $X2$ is remarkable, as it states that groundwater is exchanged to surrounding catchments in the form of a loss. In the machine learning algorithms, this is something that has not been taken into account.

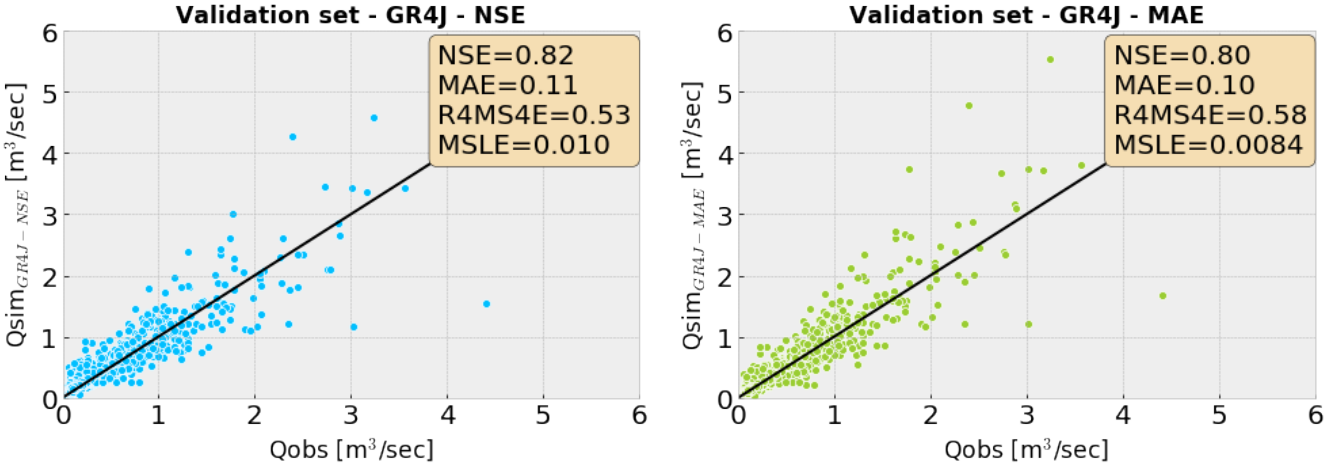


Figure 63. Test Results of GR4J model calibrated with objective functions NSE, MSE and MAE

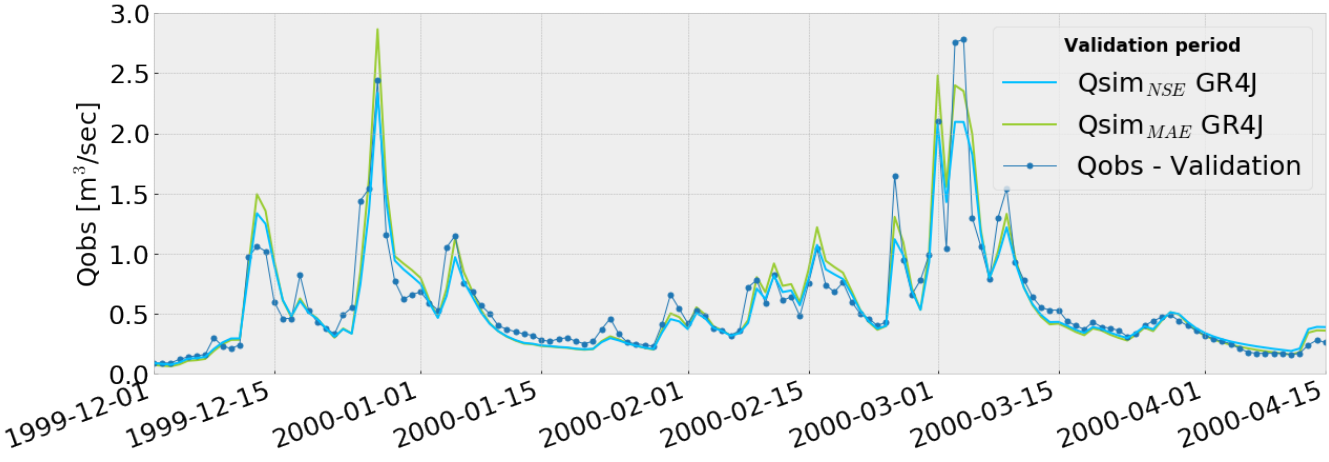


Figure 64. The $Qsim$ time series of the GR4J model calibrated with the objective functions NSE and MAE - zooming in on peak flows of the validation set

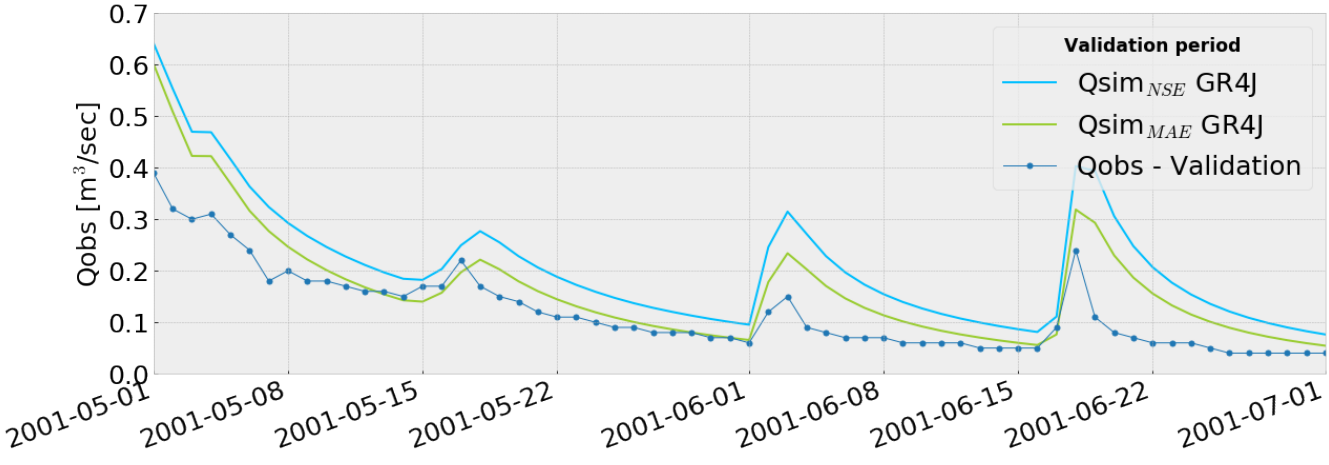


Figure 65. The $Qsim$ time series of the GR4J model calibrated with the objective functions NSE and MAE - zooming in on baseflows of the validation set

4.2.2 Limitations - GR4J model

The first limitation of the GR4J model is the fact that there is no interception store (Harlan et al., 2010): there is either a net rainfall Pn or a net potential evaporation capacity of En. In reality, the interception store for the Chaamse Beken is not equal to zero as the land uses within Chaamse Beken show possibilities for interception. Furthermore, the model assumes that 90% of the water that reaches the routing part (Pr) is converted to a slow flow infiltrating into the routing store, while only 10% is converted to a fast flow that flows on the soil surface. It should be examined in additional research if these percentages can also be calibrated to get an even better model performance.

Furthermore, note that the GR4J model is a lumped model. In future researches, also distributed versions of this model can be used to even further improve the model performance of this conceptual hydrological model in the Chaamse Beken.

4.3 Comparison machine learning algorithm & GR4J model

In this section, the outputs of the best machine learning algorithm (SVR with model setup 4) are compared with the outputs of the GR4J model (calibrated on the objective function MAE). These 2 models are chosen because of their best performance in simulating $Qobs$ during baseflow conditions.

In Figure 66, scatterplots of the test/validation set are depicted of these 2 models. Moreover, in Figure 67 the $Qsim$ time series of the test/validation set are visualised, also zoomed in on a part of the set where peakflow and baseflow are dominant. The evaluation metrics in Figure 66 show that both models can be rated as a very good overall model. The NSE of the machine learning model is 0.75, whereas the NSE for the conceptual GR4J model is slightly higher (0.80).

Moreover, the MAE is larger in the machine learning model than in the GR4J model. Especially during low flows (0-1 m^3/sec) the errors are larger in the machine learning model, compared to the GR4J model. As a result, the MSLE is higher for the machine learning model than for the GR4J model. This means that the GR4J model has a better model performance during baseflow conditions. It must be mentioned though that this difference is not that significant.

Lastly, both models do not perform well during peakflow conditions: the R4MS4E is for both models around 0.58-0.60 m^3/sec. The machine learning model underestimates peak flows more, while the conceptual GR4J model mostly overestimates peak flows.

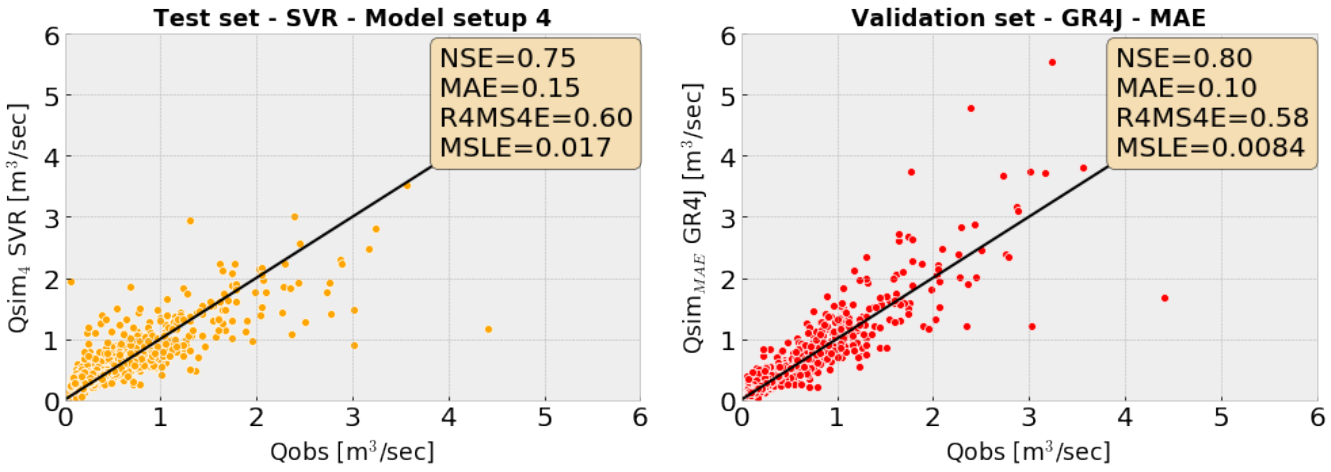


Figure 66. Test results of machine learning model SVR performed with model setup 4 (left) and test results of GR4J model calibrated with MAE as objective function (right)

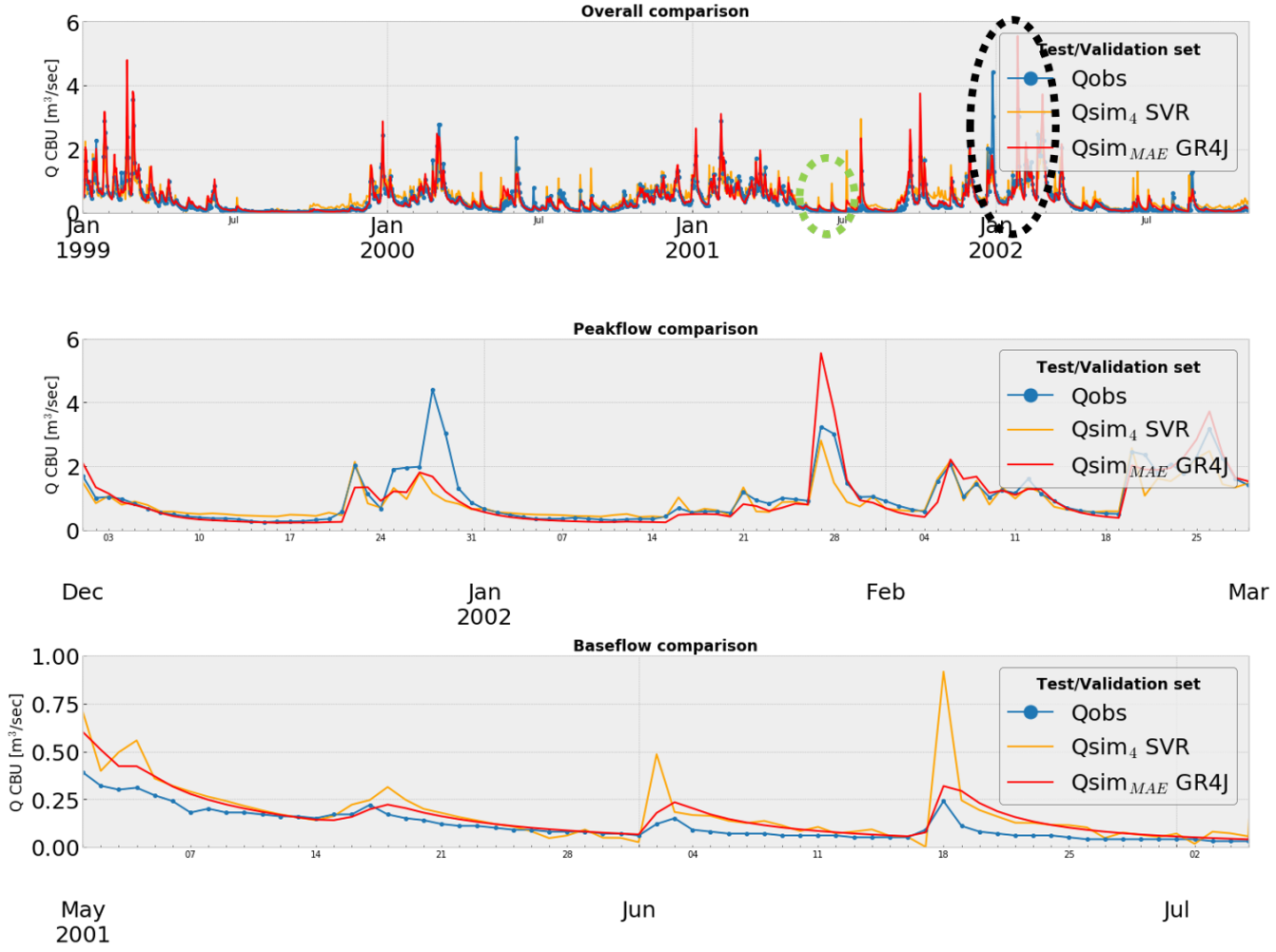


Figure 67. Overall comparison, peakflow comparison and baseflow comparison of the $Qsim$ time series of the GR4J model calibrated with the objective function MSE and of the machine learning model SVR performed with model setup 4. The green dotted circle in the overall comparison visualises the baseflow comparison and the black dotted circle the peakflow comparison.

5 Recommendations and future research

In the previous chapter already some limitation and recommendations are given. In this chapter, some ideas for future researches are further elaborated.

5.1 Screen wells outside Chaamse Beken

For the machine learning algorithms, the screen wells only within the Chaamse Beken are considered in this research. However, wells outside the Chaamse Beken can also play a role in the production of the baseflow. The boundaries of the subcatchment Chaamse Beken do not necessarily have to be the boundaries of the groundwater system itself of the Chaamse Beken. For example all the groundwater head time series of the screen-1 and screen-2 wells from the shapefile of Dinoloket used in this research, can be used as input variables for the machine learning algorithms in future research.

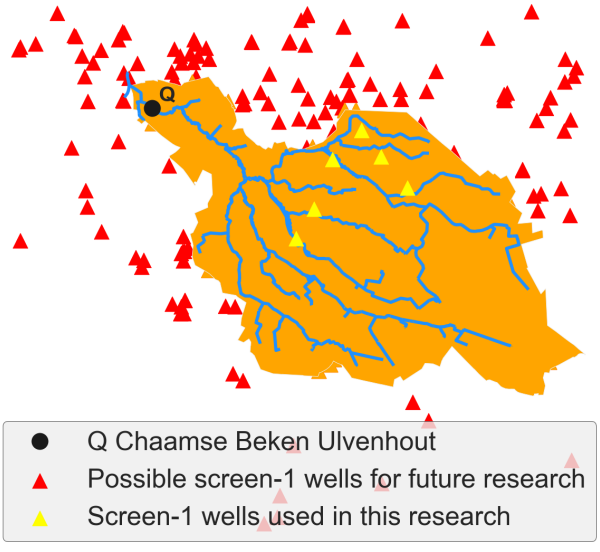


Figure 68. The screen-1 wells within Chaamse Beken used in this research and possible other screen-1 wells outside Chaamse Beken for future research)

Figure 68 shows that there are a lot of other wells around the stream discharge weir CBU. The wells used in this research are all to the right of CBU. Wells on the other side of CBU can also play a role in the groundwater system of Chaamse Beken. All the red marked wells in this figure that do have data from 1985-2003 and until now (2019) can be used for future research. Note that the more wells are used as input variables in the machine learning algorithms, the longer the computation time will be (Raschka and Mirjalli, 2017). As an advantage, by using first the relative importance of the machine learning algorithm RFR, for example the 10 wells with the largest relative importance can be considered for the other algorithms. In this way the computation time is reduced and it can be seen which wells are important for the simulation of the stream discharge CBU.

5.2 Baseflow separation

In this research, the **total** stream discharge is simulated by using different machine learning algorithms. As said before, it must be noted that the simulation of the baseflow is the most important in this research. An idea for future research is to **filter** the **baseflow** from the total stream discharge (Figure 69), before it is used in the machine learning models. Multiple baseflow separation techqniques exists according to the paper of Duncan (2019). Examples are baseflow recession and the Eckhardt digital filter.

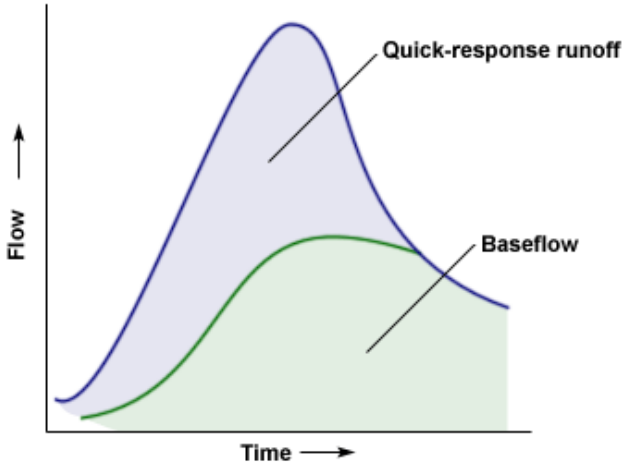


Figure 69. A division of the total stream discharge into a quick-response runoff (overland flow and part of interflow) and the baseflow (groundwater flow and part of interflow)

It is expected that the machine learning algorithms will result in a better model performance when only considering the baseflow. Note that there are also conceptual hydrological models that can simulate the baseflow part of the total stream discharge (Pelletier and Andréassian, 2019).

5.3 Manual separation of low and high flows

Instead of using baseflow separation as described above, the original time series of $Qobs$ can be broken down manually in low flows and high flows (Figure 70). For example, the low flows are defined as the values of $Qobs$ between 0 and 1 m^3/sec (Figure 71). And the high flows are defined as flows larger than 1 m^3/sec. The methods in this research for the machine learning algorithms can now be used for only the low flows. The high flows can then be neglected: simulating low flows is more important than high flows for the purpose of this research. Note that this procedure can not be applied for the GR4J model as this model can only simulate full time series (Perrin et al., 2003). However, for the GR4J model the evaluation metrics can be calculated for only the low flows.

In this way, the GR4J model can be compared with the machine learning algorithm for low flows.

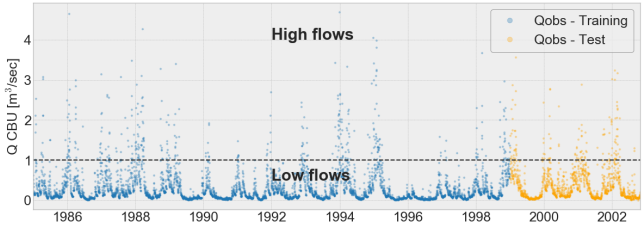


Figure 70. The original time series $Qobs$ manually broken down in high (> 1 m^3/sec) and low flows (< 1 m^3/sec)

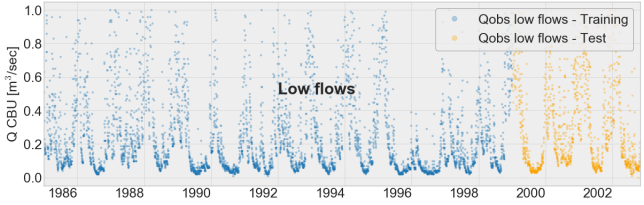


Figure 71. The manually broken down time series $Qobs$ in low flows (0-1 m^3/sec) of Figure 70

To show an example, the SVR machine learning algorithm by using model setup 4 is applied for only these low flows. A scatterplot showing $Qobs$ versus $Qsim_4$ for these low flows by using SVR is depicted in Figure 72. By using SVR for these low flows, the stream discharge can also not be simulated as larger than 1 m^3/sec. The very low flows up to 0.4 m^3/sec are mostly overestimated, while the flows between 0.4 and 1.0 m^3/sec are particularly underestimated. It is therefore recommended that this procedure is repeated for taking only the flows up to 0.4 m^3/sec. Nevertheless, this machine learning model for low flows is still rated as "good" according to the NSE of 0.72 (Cheng et al., 2017).

A scatterplot showing only the $Qobs$ for low flows versus the corresponding $Qsim_{MAE}GR4J$ is depicted in Figure 73. It is visible that for some low flows the stream discharge is simulated as larger than 1.0 m^3/sec. For this GR4J model and taking only the low flows into account, the model can be rated as "very good" according to the NSE of 0.78 (Cheng et al., 2017).

It is recommended to follow this procedure also for the other machine learning algorithms in future research. Moreover, other manually divisions in low and high flows can be examined.

5.4 Logarithmic transformation of Qobs

In the machine learning algorithms of this research, a relation is found between the input variables and the target $Qobs$. A non-linear relation exists between the variables and target. A common way to deal with this non-linearity and to make

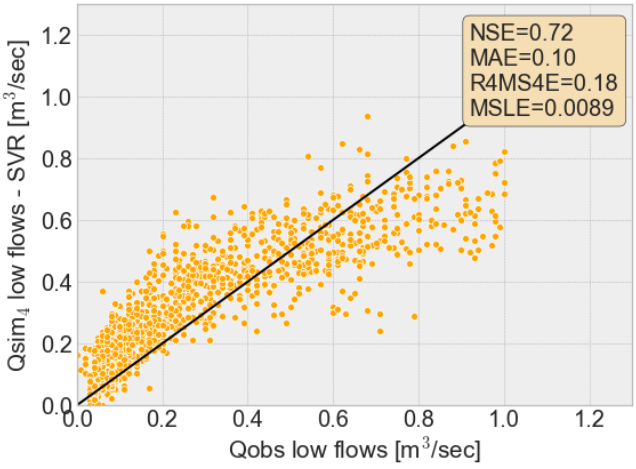


Figure 72. The manually broken down time series $Qobs$ in low flows in Figure 71 used for the machine learning algorithm SVR with model setup 4

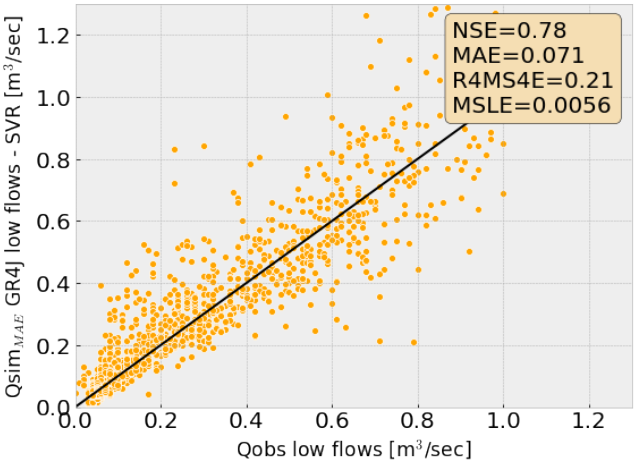


Figure 73. Test Results of GR4J model calibrated with objective functions MAE. The evaluation metrics are only computed for the low flows.

the data more interpretable is the logarithmic transformation of the target. By applying the logarithmic transformation the data will become less skewed. A generally known example of applying a logarithmic transformation within hydrology is the rating curve (Fenton, 2018). The rating curve finds a relation between the stream discharge and the water stage. By applying a logarithmic transformation to the stream discharge, the rating curve becomes a straight line. With this straight line, it is easier to find a relation and to extrapolate the data (Fenton, 2018).

The logarithmic transformation of the stream discharge can also be applied for the target $Qobs$ in the machine learning algorithms. The $Qobs$ time series becomes then the $ln(Qobs)$ time series (Figure 74).

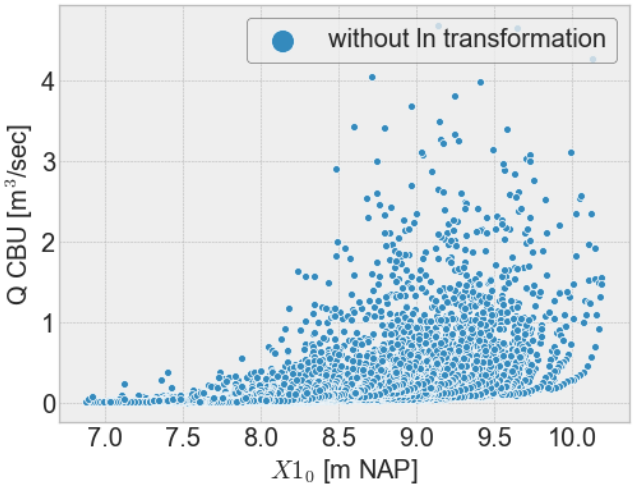


Figure 75. The relation between the groundwater heads of well $X1_0$ and the target $Qobs$ of CBU

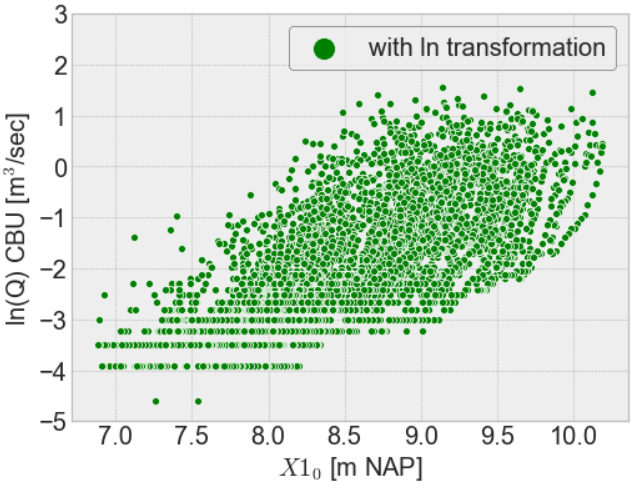


Figure 76. The relation between the groundwater heads of well $X1_0$ and the target $Qobs$ of CBU, when applying a logarithmic transformation to $Qobs$

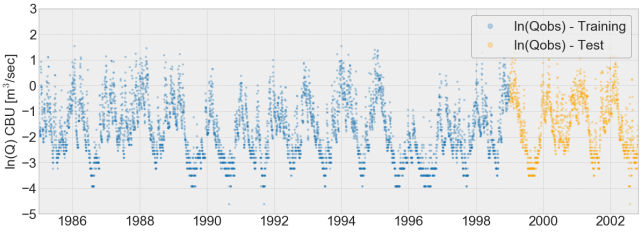


Figure 74. The logarithmic transformation of the target $Qobs$, with a division in training and test set

Figure 74 does not immediately shows what a logarithmic transformation of the stream discharge implies. To better visualize what the effect of the logarithmic transformation of $Qobs$ is, two scatterplots are visualised in respectively Figure 75 and 76. The first figure shows the relation between the groundwater heads of well $X1_0$ and the target $Qobs$ of the training set. As can be seen, a non-linear relation exist. However, when the stream discharge is transformed to $ln(Qobs$, a more linear relation exist between the stream discharge and the groundwater heads of well $X1_0$ (Figure 76). Moreover, the data becomes less skewed in comparison with the data of Figure 75.

It is recommended to further examine if using $ln(Qobs)$ as a target in the machine learning algorithms (instead of $Qobs$), is improving the model performances.

5.5 Implement other splitting criteria in MT modelling

In the applied machine learning algorithms DTR, RFR and GBR of this research, only two different splitting criteria are considered: 'mse' and 'mae'. In fact, this means that the regression trees in these algorithms are calibrated on this criteria. However, for the GR4J model the NSE is also used as a objective function. For more fair comparison with the GR4J model, the splitting criterion NSE should also be considered in the algorithms DTR, RFR and GBR.

Moreover, the evaluation metric MSLE can also be considered as the splitting criterion for the MT algorithms. With this MSLE function, the model focuses more on the fit of low flows. And since low flows are the most important in this research, it is recommended to examine this in further research. Note that for fair comparison with the GR4J model, the MSLE should then also be used as objective function in the GR4J model.

5.6 Validation of groundwater heads

The groundwater heads used in this research are validated with Pastas TSA. These simulated heads with Pastas TSA are used as an input for the machine learning algorithms. This means that the machine learning model can only recognize these simulated groundwater heads in order to simulate stream discharge. For example, well $X1_0$ is calibrated with Pastas TSA for the period until 2010. If groundwater heads after 2010 are input for the machine learning models, the data can not be simulated again with Pastas TSA with other model parameters for this period. This means that the model parameters of the calibrated groundwater heads before 2010 need to be used to extrapolate the groundwater heads to the period after 2010. If this is not done correctly, the model can fail to recognize the groundwater heads as inputs and the model performance will decrease. This is known as data leakage in machine learning (Luigi, 2019).

In order to overcome this problem of data leakage, the interpolated groundwater head time series can also be used as inputs for the machine learning models. Note that these interpolated series can only be used if Pastas TSA showed that the groundwater head time series can be validated.

5.7 Validation of Qobs

Lastly, the validation of the stream discharge time series ($Qobs$) of CBU should be more highlighted. In this research, the correlation of the time series of CBU with other surrounding time series is examined. This is a correlation technique and is often used as a validation tool. However, it is recommended for future research to further examine possible validation tools of stream discharge.

6 Conclusions

Three main research questions provided the basis for this study. These research questions are briefly discussed in this chapter. Subsequently, an answer to these questions is formulated.

6.1 Conclusions of research question 1

Can we analyze if there is/are (a) relation(s) between groundwater heads (X) and stream discharge (Y) in the same subcatchment by using machine learning algorithms, such that we can simulate the stream discharge time series from these groundwater heads in the future, or to fill in gaps in the historical stream discharge time series?

To answer this research question, the subcatchment Chaamse Beken is used for the groundwater head time series (X) and the stream discharge gauge Chaamse Beken Ulvenhout for the stream discharge time series (Y). In total six different wells within Chaamse Beken were used, that all have a screen-1 and a screen-2. This means six screen-1 wells ($X1$) and six screen-2 wells ($X2$). Moreover, four different machine learning algorithms are used in this research: decision tree regression (DTR), random forest regression (RFR), gradient boosting regression (GBR), and support vector regression (SVR). For these algorithms different combinations of input variables were used: $X1$, $X2$, precipitation, and potential evaporation. Note that it is attended to find a link for inputs at day i and the output also at day i. The most important findings are summarized below:

- By using only the most representative screen-1 well of the six screen-1 wells as input, no relation was found between the groundwater heads of this well and the stream discharge for all four algorithms. The NSE values of these models did not reach values larger than 0.30.

- When using all six screen-1 wells, for the algorithms RFR, GBR and SVR a relation was found. The overall model performance when using all six screen-1 wells resulted in a NSE of 0.69-0.70 for these 3 algorithms.

- Addition of the six screen-2 wells to the screen-1 wells, did not or did not significantly improve the overall model performance of the algorithms RFR, GBR and SVR. This means that screen-2 wells were not needed for the relation between groundwater heads and stream discharge within the Chaamse Beken. Also, the low flow model performance (expressed in evaluation metric MSLE) did not improve. In other words, the screen-2 wells are not needed for the simulation of the baseflow of the stream discharge in the Chaamse Beken. Lastly, note that the NSE value of the algorithm DTR did increase, but is still very small in comparison with the other algorithms.

- Addition of precipitation and potential evaporation to the six screen-1 wells, did not result in a better overall model performance for RFR and GBR, but the overall model performance of DTR and SVR did increase. The overall performance of the DTR algorithm is still poor in comparison with the other algorithms. The overall model performance of SVR significantly increased to a NSE of 0.75, of which it can be said that the model can be rated as good (according to Cheng et al. (2017)).

- Taking peak flow model performance into account, it is expected that precipitation plays a significant role in the simulation of peak flows. The peak flow model performance is evaluated with the metric R4MS4E. It can be said that the peak flow model performance does not improve when also precipitation is taken into account in addition to the screen-1 wells. Apparently, these machine learning models can not be used when simulation of the peak flows is important.

- It is known that precipitation events are subject to delays before they effect the stream discharge, as each hydrological system has a certain memory or state. Therefore, also some shifts (1 to 6 days) of the precipitation time series were added as inputs in the machine learning models. In addition, rolling means (3,7,14,30,120 days) of the groundwater head time series were added as input series to take the memory of the system into account. It is concluded that for the algorithms RFR and GBR the delays and memory of the system play a small role in the overall model performance (NSE increases from 0.69-0.70 to 0.70-0.71). For the algorithm SVR it does not improve overall model performance. Moreover, the peak flows are still not well simulated for all the algorithms.

6.2 Conclusions of research question 2

Which DDM is the most suitable for finding a relation with the stream discharge?

The SVR algorithm is able to get the best overall model performance of all four algorithms: a NSE value of 0.75 can be reached. For this model the variables $X1$, precipitation and potential evaporation are needed as input variables. The second best overall model performance is obtained when using algorithm GBR and only the six screen-1 wells as inputs (a NSE of 0.70). An important difference between these 2 algorithms is the computation time. Where the computation time of SVR is only seconds to a few minutes, the computation time of GBR is a couple of hours. Note that for the subcatchment Chaamse Beken it is mostly important to simulate the low flows of the stream discharge. The MSLE value of these two above-mentioned algorithms is for both algorithms 0.017 $\ln^2(m^3/sec)$. This is also the best score for the low flow model performance when considering all algorithms and all possible input variables. Since SVR has the best overall model performance and has one of the best low flow model performances, this algorithm is selected for the final research question.

Lastly, note that the algorithm DTR is not an appropriate machine learning algorithm for this research. It can not be used to find an accurate relation between groundwater heads and stream discharge.

6.3 Conclusions of research question 3

Does the best performing DDM perform as well or better than a conceptual hydrological model?

The GR4J model is chosen in this research as the conceptual hydrological model. This model is calibrated with the objective function MAE. The results of the GR4J model showed a very good overall model performance: a NSE of 0.80. This means that the GR4J model outperforms the SVR model, when considering overall model performance of stream discharge simulation. For low flows, the GR4J model shows slightly better model performance: a MSLE of 0.0084 $\ln^2(m^3/sec)$ instead of 0.017 $\ln^2(m^3/sec)$ for the SVR algorithm. Neither model performs well for peak flow. Overall, the conceptual model shows somewhat better results than the machine learning model SVR.

It must be noted that the SVR model and the GR4J model are completely different in their practical application:

- Where physical understanding of the hydrological system is needed for the GR4J model, the machine learning algorithm (a data-driven model) can be applied without these physics of the system.
- In the conceptual GR4J model the memory/state of the system is included. Furthermore, delays (for example P resulting in stream discharge) within the system are directly included in the conceptual model. The memory of the system is not included in the the machine learning model, but the delays are indirectly included in the machine learning model. The groundwater heads are namely a weighted moving average of recharge and do indirectly include the delays within the system.
- Groundwater head time series play an important role for the machine learning model. These groundwater heads are not directly included in the GR4J model. It is concluded that groundwater heads play an important part in the simulation of stream discharge, especially for baseflow.

References

Adnan, R., Liang, Z., Trajkovic, S., Zounemat-Kermani, M., Li, B., and Kisi, O.: Daily streamflow prediction using optimally pruned extreme learning machine, Journal of Hydrology, 577, 2019.

Beersma, J., Buishand, T., and Buiteveld, H.: Droog, droger, droogst:KNMI/RIZA-bijdrage aan De tweede fase van de Droogtestudie Nederland, Tech. rep., KNMI, 2004.

Behnamian, A., Millard, K., Banks, S., White, L., Richardson, M., and Pasher, J.: A Systematic Approach for Variable Selection with Random Forests: Achieving Stable Variable Importance Values, IEEE Geoscience and Remote Sensing Letters, 2017.

Bhatnagar, S.: Introduction to Regression Trees[lecture slides], 2018.

Bosch, D., Arnold, J., Allen, P., Lim, K., and Park, Y.: Temporal variations in baseflow for the Little River experimental watershed in South Georgia, USA, Journal of Hydrology: Regional studies, 10, 110–121, 2017.

Broekhoven, F., Smidt, J., and Biesheuvel, A.: Klimaatrobuuste Bovenlopen Beeksysteem Hoge Zandgronden, Tech. rep., WBD, 2019.

Buckingham, D., Skalka, C., and Bongard, J.: Inductive machine learning for improved estimation of catchment-scale snow water equivalent, Journal of Hydrology, 524, 311–325, http://dx.doi.org/10.1016/j.jhydrol.2015.02.042, 2015.

Buzacott, A., Tran, B., van Ogtrop, F., and Vervoort, R.: Conceptual Models and Calibration Performance - Investigating Catchment Bias, Water, 11, https://doi.org/10.3390/w11112424, 2019.

Cheng, K., Lien, Y., Wu, Y., and Su, Y.: On the criteria of model performance evaluation for real-time flood forecasting, Stochastic Environmental Research and Risk Assessment, 31, 1123–1146, https://doi.org/10.1007/s00477-016-1322-7, 2017.

Chiew, F., Zheng, H., and Potter, N.: Rainfall-Runoff Modelling Considerations to Predict Streamflow Characteristics in Ungauged Catchments and under Climate Change, water, 1319, 7624–7629, 2018.

Collenteur, R., Bakker, M., Calje, R., and Schaars, F.: Source code for pastas.modelstats, https://pastas.readthedocs.io/en/latest/_modules/pastas/modelstats.html#Statistics.many, 2019a.

Collenteur, R., Bakker, M., Caljé, R., Klop, S., and Schaars, F.: Pastas: Open Source Software for the Analysis of Groundwater Time Series, Groundwater, 57, 877–885, https://ngwa.onlinelibrary.wiley.com/doi/epdf/10.1111/gwat.12925, 2019b.

Dawson, C., Abrahart, R., and See, L.: HydroTest: A web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts, Environmental Modelling & Software, 22, 1034–1052, 2007.

Duncan, H.: Baseflow separation - A practical approach, Journal of Hydrology, 575, 308–313, https://doi.org/10.1016/j.jhydrol.2019.05.040, 2019.

Fenton, J.: On the generation of stream rating curves, Journal of Hydrology, 564, 748–757, https://doi.org/10.1016/j.jhydrol.2018.07.025, 2018.

Giri, S., Zhang, Z., Krasnuk, D., and Lathrop, R.: Evaluating the impacts of land uses on stream integrity using machine learning algorithms, Science of the total environment, 696, https://doi.org/10.1016/j.scitotenv.2019.133858, 2019.

Gonzales, A., Nonner, J., Heijkers, J., and Uhlenbrook, S.: Comparison of different base flow separation methods in a lowland catchment, Hydrology and Earth System Sciences, 13, 2055–2068, 2009.

Harlan, D., Wangsadipura, M., and Munajat, C., eds.: Rainfall-Runoff Modelinf of Citaram Hulu River Basin by Using GR4J, vol. 2, Proceedings of the World Congress on Engineering 2010, http://www.iaeng.org/publication/WCE2010/WCE2010_pp1607-1611.pdf, 2010.

Hartong, H. and Termes, P.: Handboek debietmeten in open waterlopen, stowa, https://edepot.wur.nl/14699, 2009.

Knoben, W., Freer, J., and Woods, R.: Technical note: Inherent benchmark or not? Comparing Nash-Sutcliffe and Kling-Gupta efficiency scores, Hydrology and Earth System Sciences, 23, 4323–4331, 2019.

Krause, P., Boyle, D., and Bäse, F.: Comparison of different efficiency criteria for hydrological model assessment, Advances in Geosciences, 2005.

Kunnath-Poovakka, A. and Eldho, T.: A comparative study of conceptual rainfall-runoff models GR4J, AWBM and Sacramento at catchments in the upper Godavari river basin, India, Journal of Earth System Science, 226, https://doi.org/10.1007/s12040-018-1055-8, 2019.

Le Moine, N., V. Andre assian, C. P., and Michel, C.: How can rainfall-runoff models handle intercatchment groundwater flows? Theoretical study based on 1040 French catchments., Water Resources Research, 43, https://agupubs.onlinelibrary.wiley.com/doi/epdf/10.1029/2006WR005608, 2007.

Li, C.: A Gente Introduction to Gradient Boosting [lecture slides], 2018.

Ligtvoet, W., Beugelink, G., and Franken, R.: Evaluation of the Water Framework Directive in the Netherlands; costs and benefits, Tech. Rep. 500140004, Netherlands Environmental Assessment Agency, 2008.

Liu, Z., Wang, Y., Xu, Z., and Duan, Q.: Handbook of Hydrometeorological Ensemble Forecasting, chap. Conceptual Hydrological Models, pp. 195–224, Springer, Berlin, Heidelberg, 2017.

Lu, H., Zou, N., Jacobs, R., B., A., Lu, X., and Morgan, D.: Computational Materials Science, Computational Materials Science, 169, https://doi.org/10.1016/j.commatsci.2019.06.010, 2019.

Luigi: How Data Leakage Impacts Machine Learning Models, https://mlinproduction.com/data-leakage/, 2019.

Luxemburg, W. and Coenders, A.: CIE4440 Hydrological processes and measurments [Lecture Notes], 2017.

Mishra, S. and Datta-Gupta, A.: Applied Statistical Modeling and Data Analytics, chap. 8, pp. 195–224, Elsevier, 2018.

Msiza, I., Nelwamondo, F., and Marwala, T.: Artificial Neural Networks and Support Vector Machines for Water Demand Time Series Forecasting, Journal of Hydraulic Engineering, 2007.

Muste, M., Vermeyen, T., Hotchkiss, R., and Oberg, K.: Acoustic Velocimetry for Riverine Environments, Journal of Hydraulic Engineering, 133, 1297–1298, 2007.

Paisitkriangkrai, P.: Linear Regression and Support Vector Regression], 2012.

Pelletier, A. and Andréassian, V.: Hydrograph separation: an impartial parametrization for an imperfect method, Hydrology and Earth System Sciences, 503, https://doi.org/10.5194/hess-2019-503, 2019.

Perrin, C., Michel, C., and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, Journal of Hydrology,

279, 275–289, https://doi.org/10.1016/S0022-1694(03)00225-7, 2003.

Probst, P., Boulesteix, A., and Bischl, B.: Tunability: Importance of Hyperparameters of Machine Learning Algorithms, Journal of Machine Learning Research, 20, 1–32, 2019.

Raschka, S. and Mirjalli, V.: Python Machine Learning, Packt Publishing, 2017.

Ritzema, H., Heuvelink, G., Heinen, M., Bogaart, P., van der Bolt, F., Hack-ten Broeke, M., Hoogland, T., Knotters, M., Massop, H., and Vroon, H.: Meten en interpreteren van grondwaterstanden: analyse van methodieken nauwkeurigheid, Tech. rep., Alterra Wageningen, https://library.wur.nl/WebQuery/wurpubs/fulltext/215081, 2012.

Sachindra, D., Ahmed, K., Mamunur Rashid, M., Shahid, S., and Perera, B.: Statistical downscaling of precipitation using machine learning techniques, Atmospheric Research, 212, 240–258, https://doi.org/10.1016/j.atmosres.2018.05.022, 2018.

Salas, J., Delleur, J., Yevjevich, V., and Lane, W.: Applied modelling of hydrologic time series, Water Resources Publications, LLC, 1980.

Shin, M. and Kim, C.: Analysis of the effect of uncertainty in rainfall-runoff models on simulation results using a simple uncertainty-screening method, water, 2019.

Sitterson, J., Knightes, C., Parmar, R., Wolfe, K., Muche, M., and Avant, B.: An Overview of Rainfall-Runoff Model Types, Tech. rep., United States Environmental Protection Agency (EPA), 2017.

Solomatine, D. and Ostfeld, A.: Data-Driven Modelling: Some Past Experiences and New Approaches, Journal of Hydroinformatics, 133, 2008.

Steenbergen, N. and Willems, P.: Method for testing the accuracy of rainfall–runoff models in predicting peak flow changes due to rainfall changes, in a climate changing context, Journal of Hydrology, s 414–415, 425–434, https://doi.org/10.1016/j.jhydrol.2011.11.017, 2012.

Stravs, L. and Brilly, M.: Development of a low-flow forecasting model using the M5 machine learning method, Hydrological Sciences Journal, pp. 466–477, 2007.

Valipour, M.: Long-term runoff study using SARIMA and ARIMA models in the United States, Meteorological Applications 22, pp. 529–598, 2015.

Van Loon, A., Van Lanen, H., Seibert, J., and Torfs, P.: Adaptation of the HBV model for the study of drought propagation in European catchment, Geophysical Research Abstracts, 11, https://meetingorganizer.copernicus.org/EGU2009/EGU2009-9589.pdf, 2009.

Vink, K.: Personal conversation, 2019.

Waajen, G., van Heemskerk, J., Oosthoek, C., Lambregts, C., Lambregts-Van de Clundert, F., Rijsdijk, N., and Göbel, M.: Watersysteemanalyse De Agger, Tech. rep., Waterschap Brabantse Delta, https://www.brabantsedelta.nl/mgd/files/watersysteemanalyse-agger.pdf, 2018.

WBD: Hoofdrapport Evaluatie Droogte 2018 waterschap Brabantse Delta, Tech. Rep. 500140004, Waterschap Brabantse Delta, https://edepot.wur.nl/467636, 2018.

Westerhoff, W., Menkovic, A., and de Lang, F.: A revised lithostratigraphy of Upper Pliocene and Lower Pleistocene fluvial deposits from Rhine, Meuse and Belgian rivers in the Netherlands, Netherlands Journal of Geosciences, https://research.vu.nl/ws/portalfiles/portal/42184389/chapter+2.pdf, 2009.

Yaseen, Z., El-shafie, A., Jaafar, O., Afan, H., and Sayl, K.: Artificial intelligence based models for stream-flow forecasting: 2000-2015, Journal of Hydrology, 530, 829–844, http://dx.doi.org/10.1016/j.jhydrol.2015.10.038, 2015.

Zhang, Y.-K. and Schilling, K.: Increasing streamflow and baseflow in Mississippi River since the 1940 s: Effect of land use change, Journal of Hydrology, 324, 412–422, 2006.

Zhao, G., Pang, B., Xu, Z., and Xu, L.: A Hybrid Machine Learning Framework for Real-Time Water Level Prediction, Journal of Hydrology, 13, https://doi.org/10.1016/j.jhydrol.2019.124422, 2019.

Appendix A: The performance of monitoring wells with different screens

In this Appendix, it is explained why different screens are used in monitoring wells. Different screens are needed to see if infiltration or seepage between different layers occurs. Therefore, the screens are placed in different layers. When the water level of a screen-1 well is higher than that of a screen-2 well, infiltration will occur to deeper layers (Figure A1). On the other hand, when the water level of the screen-2 well is higher than the level of the screen-1 well, seepage will occur and water from a deeper layer will be pushed toward the screen-1 well layer (Figure A2).

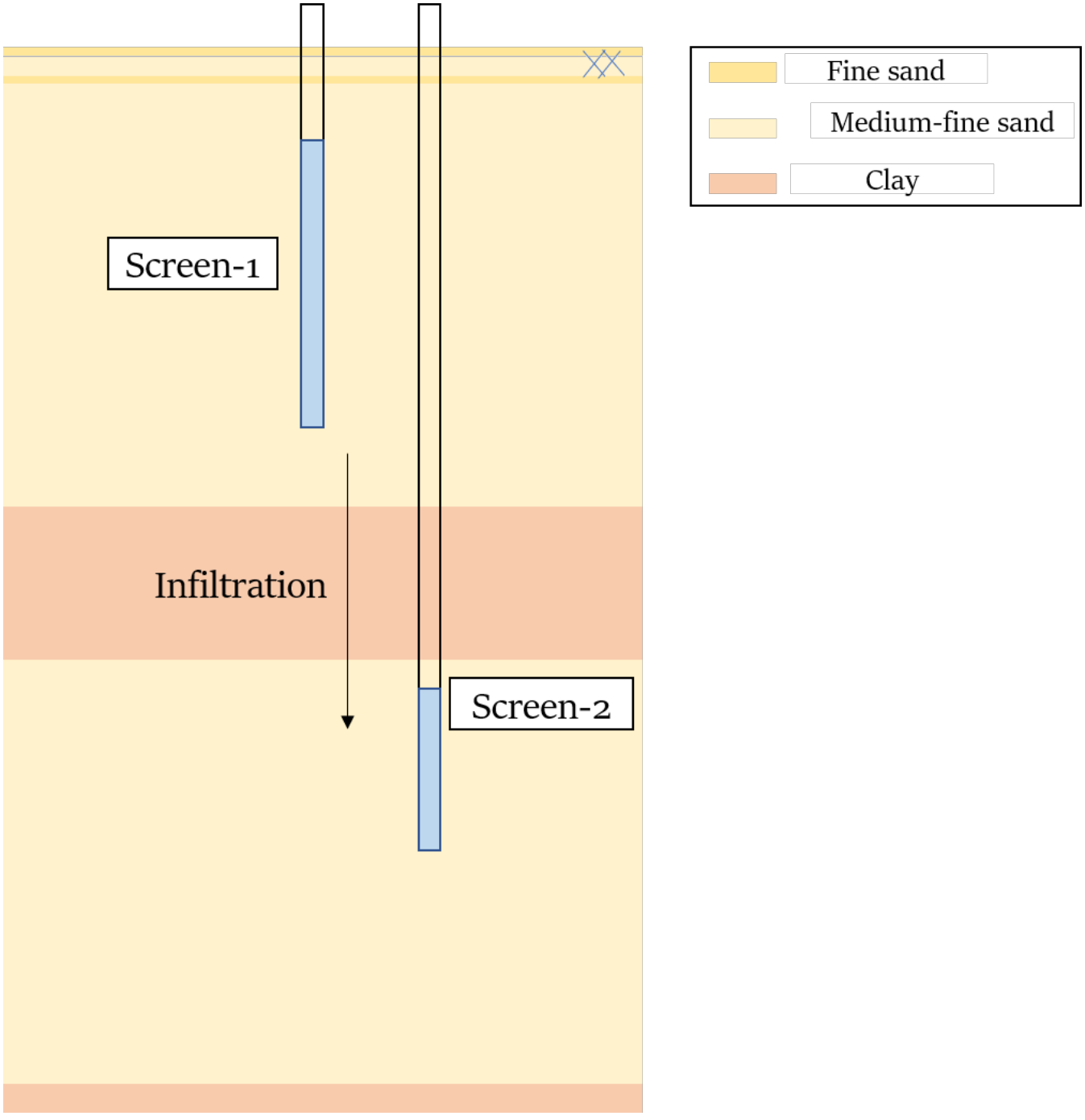


Figure A1. The phenomena **infiltration** between 2 layers

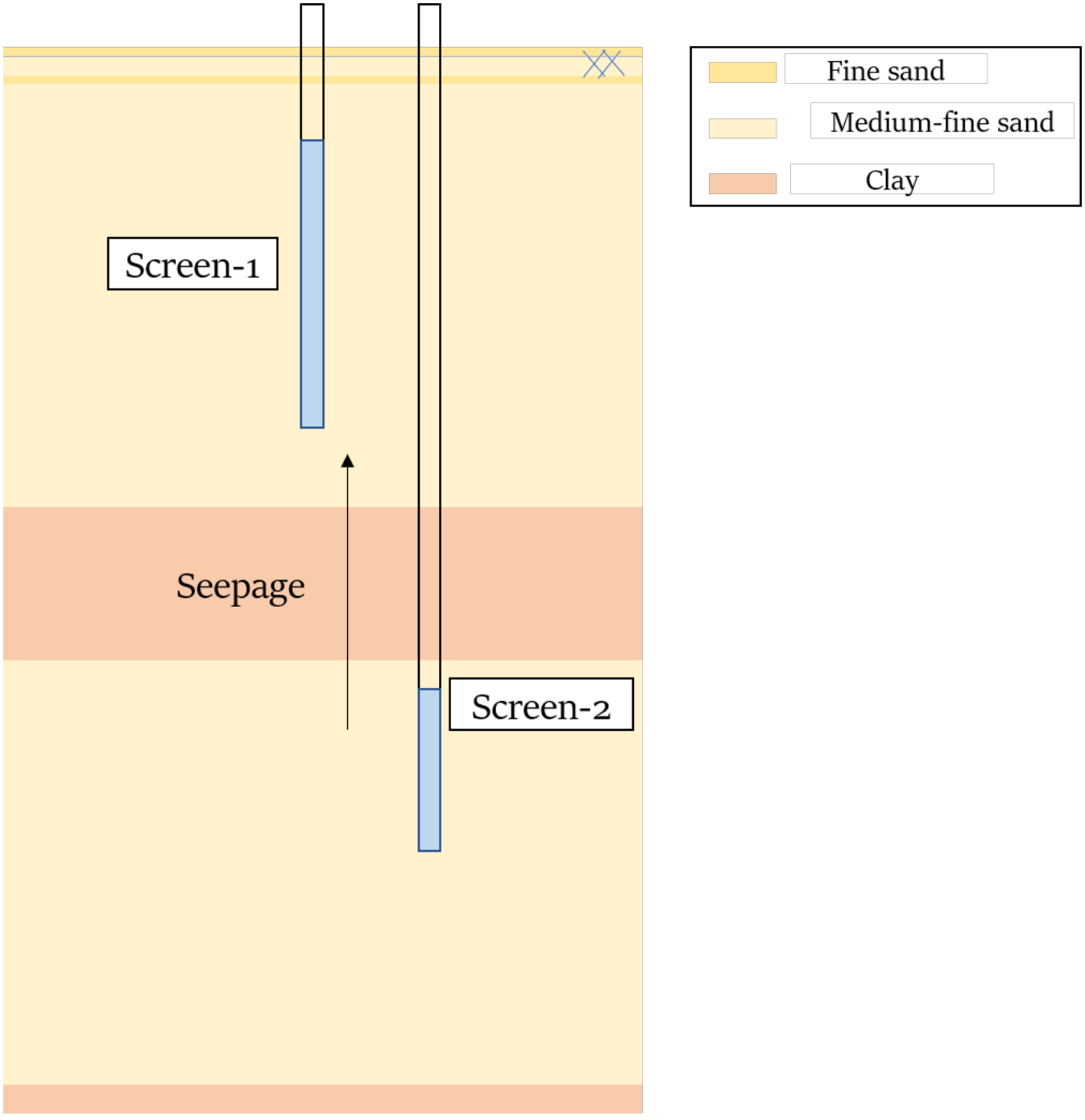


Figure A2. The phenomena **seepage** between 2 layers

Appendix B: TSA with Pastas

In this Appendix, first the time series of the explanatory variables used within TSA Pastas are given, followed by the results of the simulated groundwater head time series with Pastas.

B1 Explanatory variables Chaamse Beken

The explanatory variables used for simulating the groundwater head time series of screen-1 wells ($X1$) and screen-2 wells ($X2$) are: precipitation P, potential evaporation Ep, $Qpumping$ Prinsenbosch and the $Qeffluent$ of RWZI Chaam. The time series are depicted below.

B1.1 Explanatory variables P and Ep

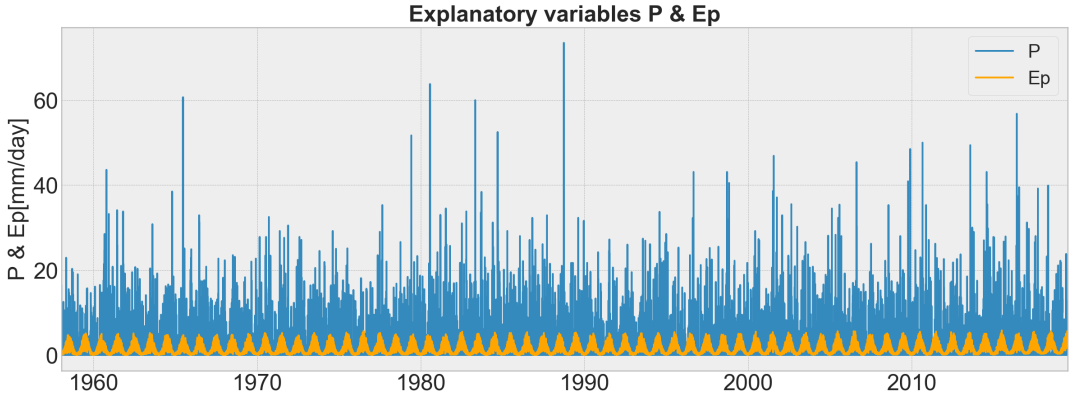


Figure B1. Time series of explanatory variables P and Ep

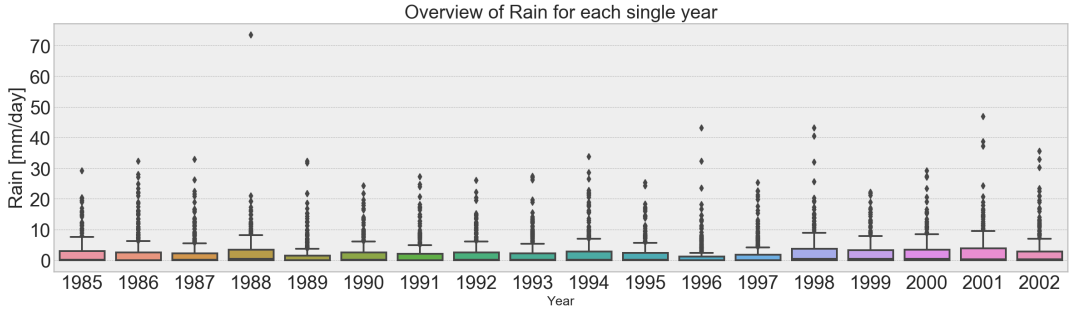


Figure B2. Overview time series of explanatory variable P per year

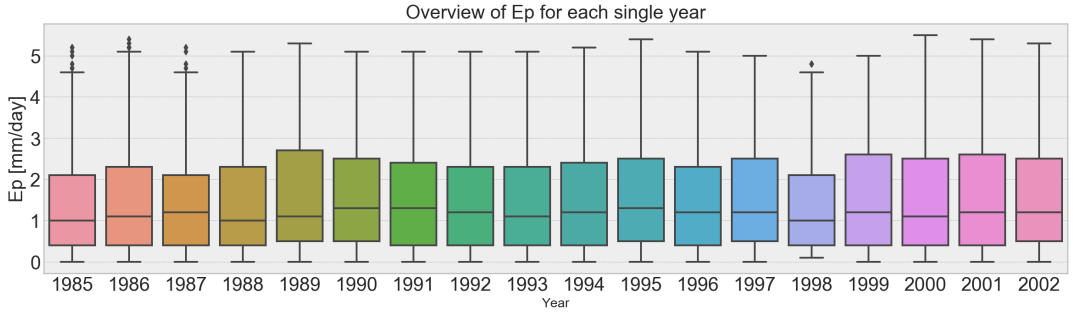


Figure B3. Overview time series of explanatory variable Ep per year

B1.2 Explanatory variable Qpumping Prinsenbosch

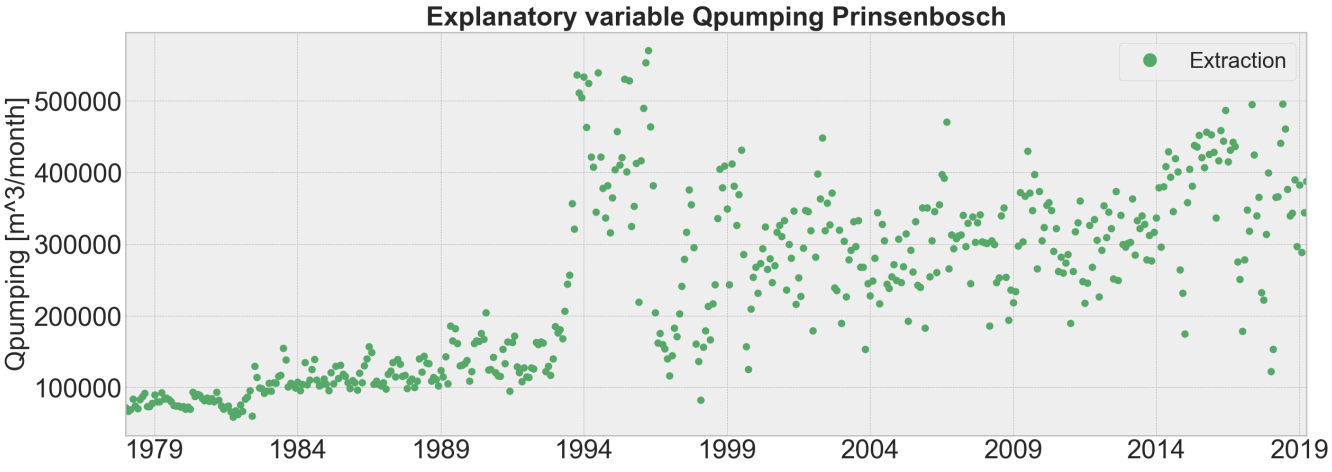


Figure B4. Time series of explanatory variable Qpumping Prinsenbosch

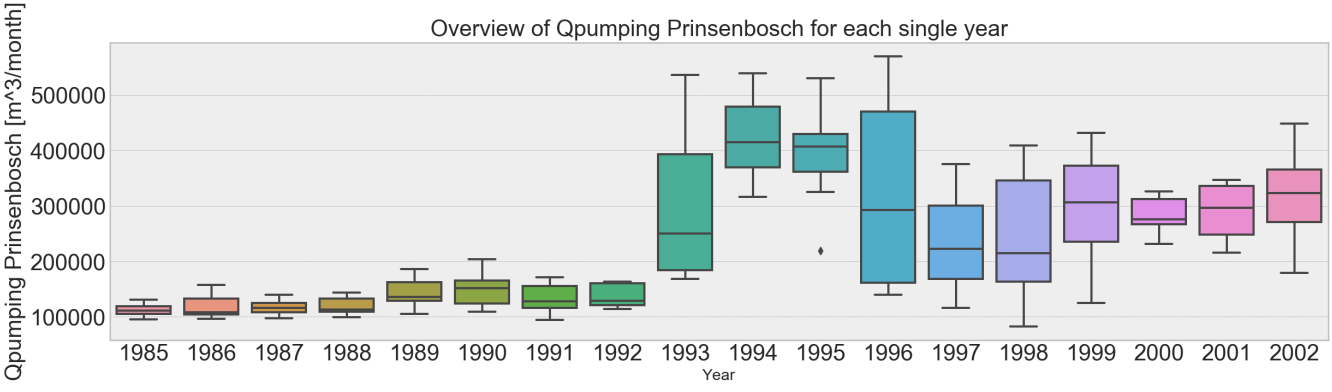


Figure B5. Overview time series of explanatory variable Qpumping Prinsenbosch per year

B1.3 Explanatory variable Qeffluent of RWZI Chaam

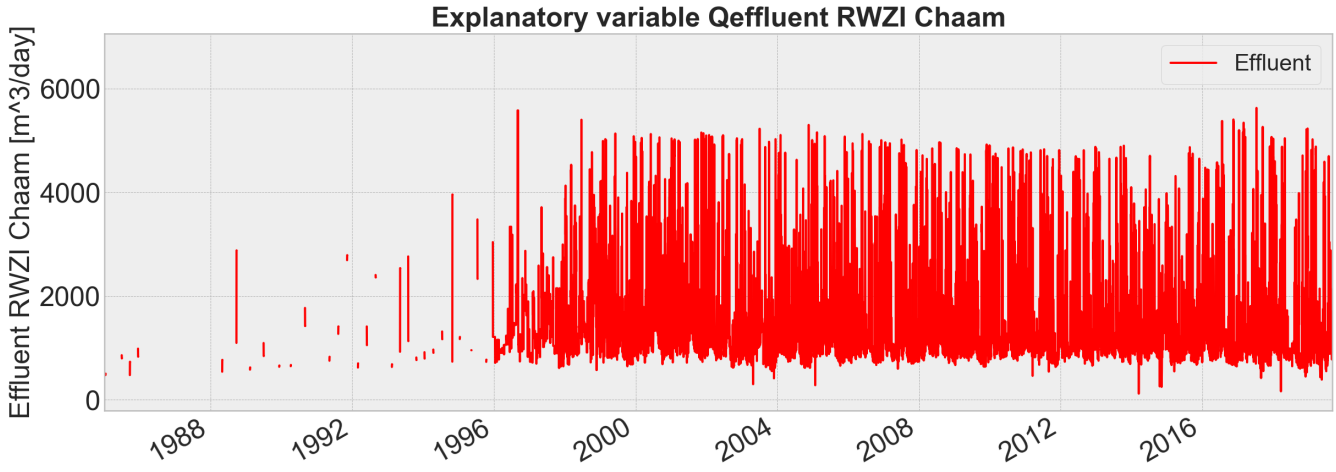


Figure B6. Time series of explanatory variable Qeffluent RWZI Chaam

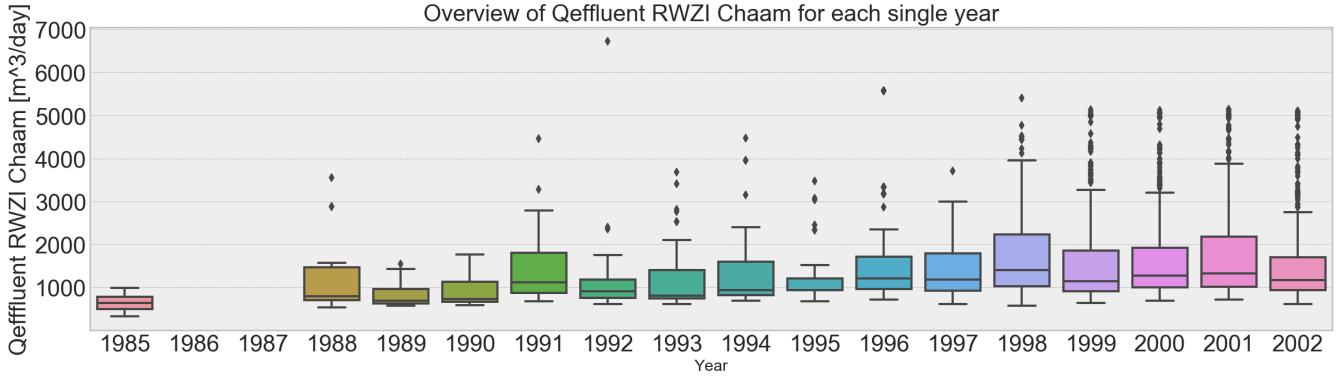


Figure B7. Overview time series of explanatory variable Qeffluent RWZI Chaam

B2 Simulated groundwater time series Pastas TSA - screen-1 wells

In this Appendix, the results of the simulated groundwater head time series with Pastas TSA for the screen-1 wells are depicted.

B2.1 $X1_0 : B50B0074_1$

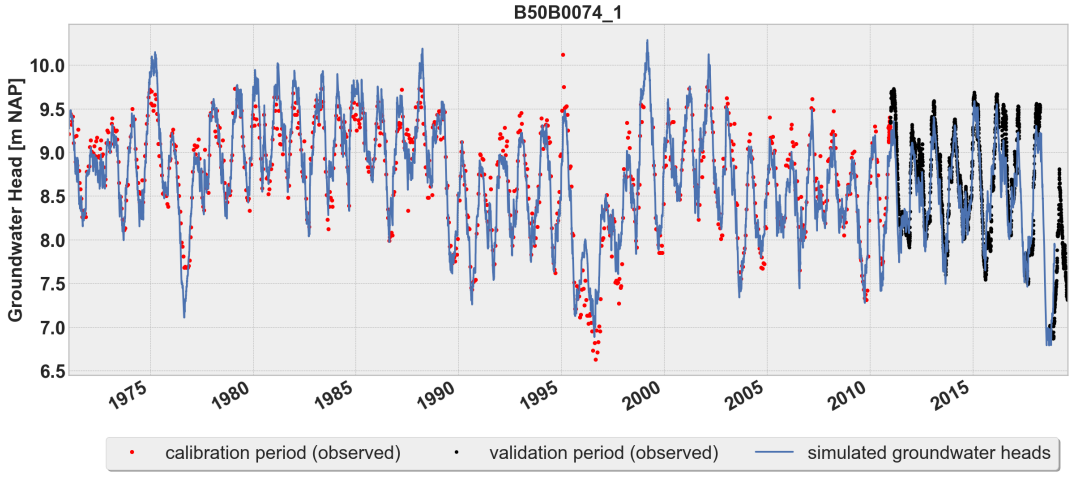


Figure B8. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_0$: $B50B0074_1$

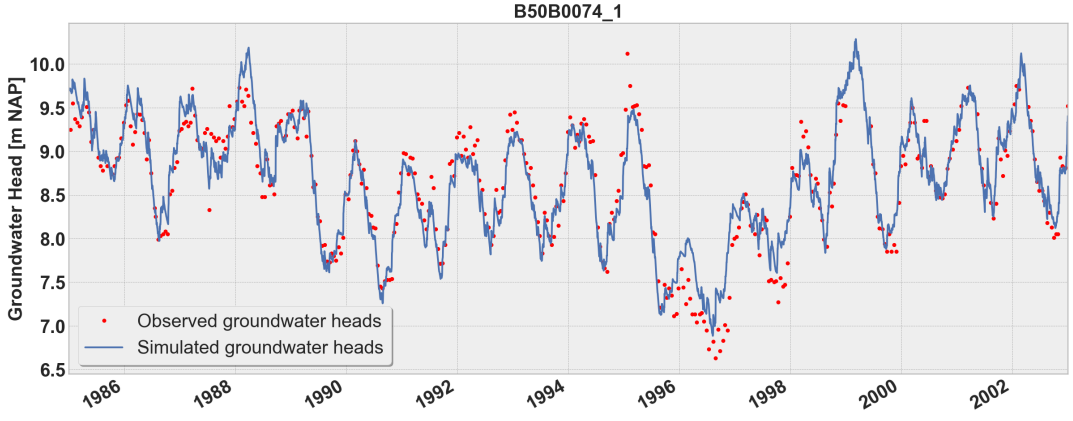


Figure B9. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_0$: $B50B0074_1$

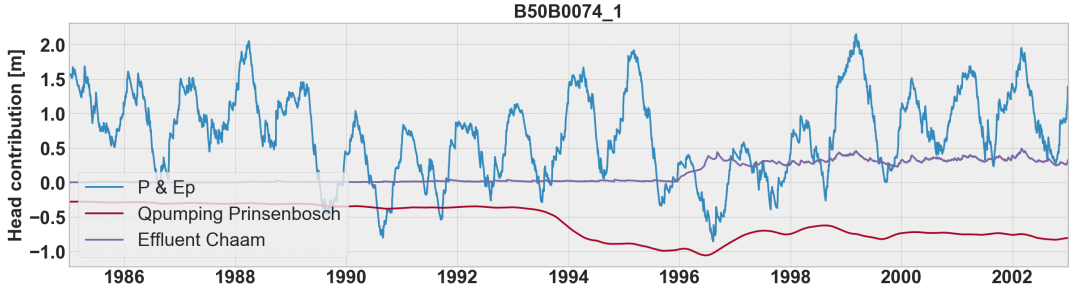


Figure B10. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_0$: $B50B0074_1$

```
Model Results B50B0074_1                          Fit Statistics
================================================================
nfev     43                      EVP                      88.78
nobs     900                     R2                        0.89
noise    True                    RMSE                      0.21
tmin     1971-01-15 00:00:00     AIC                      16.96
tmax     2010-01-01 00:00:00     BIC                      69.79
freq     D                       ___
warmup   3650 days 00:00:00      ___
solver   LeastSquares            ___

Parameters (11 were optimized)
================================================================
                             optimal   stderr     initial  vary
Rain & Evaporation_A        1.190120   ±3.10%    0.208808  True
Rain & Evaporation_n        0.966127   ±0.84%    1.000000  True
Rain & Evaporation_a      224.640546   ±4.49%   10.000000  True
Rain & Evaporation_f       -1.075833   ±2.85%   -1.000000  True
Qpumping Prinsenbosch_A    -0.000080   ±9.54%   -0.000201  True
Qpumping Prinsenbosch_rho   0.708516  ±43.86%    1.000000  True
Qpumping Prinsenbosch_cS  933.836217  ±60.81%  100.000000  True
Effluent Chaam_A            0.000201  ±20.37%    0.001011  True
Effluent Chaam_a           59.965154  ±26.65%   10.000000  True
constant_d                  8.424330   ±0.77%    8.590603  True
noise_alpha                82.003893   ±6.12%    1.000000  True

Parameter correlations |rho| > 0.5
================================================================
Rain & Evaporation_A      Rain & Evaporation_a       0.71
Rain & Evaporation_n      Rain & Evaporation_a      -0.70
Rain & Evaporation_f      constant_d                -0.91
Qpumping Prinsenbosch_A   Qpumping Prinsenbosch_rho  0.67
                          Qpumping Prinsenbosch_cS  -0.78
                          Effluent Chaam_A          -0.91
                          Effluent Chaam_a          -0.67
Qpumping Prinsenbosch_rho Qpumping Prinsenbosch_cS  -0.95
                          Effluent Chaam_A          -0.60
Qpumping Prinsenbosch_cS  Effluent Chaam_A           0.73
                          Effluent Chaam_a           0.56
Effluent Chaam_A          Effluent Chaam_a           0.76
```

Figure B11. Fit report Pastas TSA $X1_0$: $B50B0074_1$

B2.2 $X1_1 : B50B0075_1$

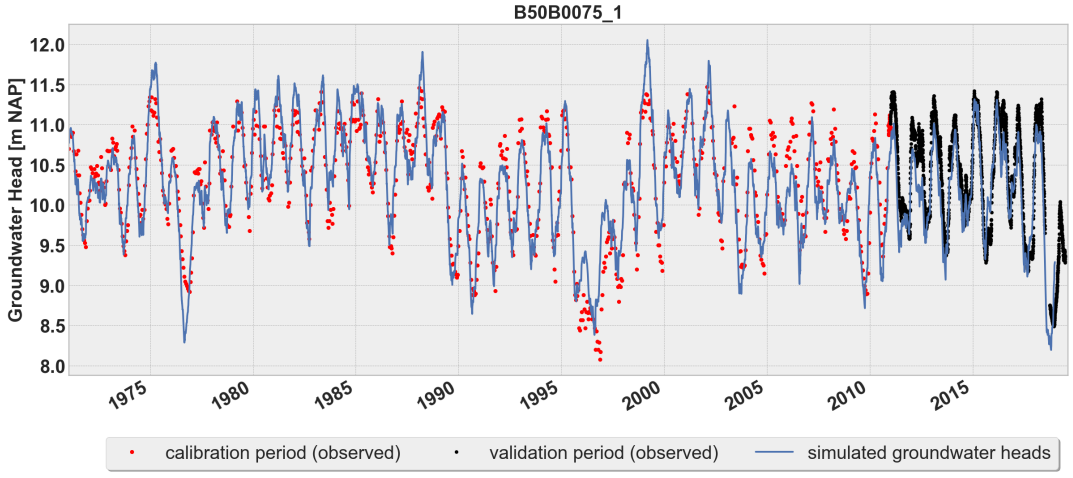


Figure B12. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_1$: $B50B0075_1$

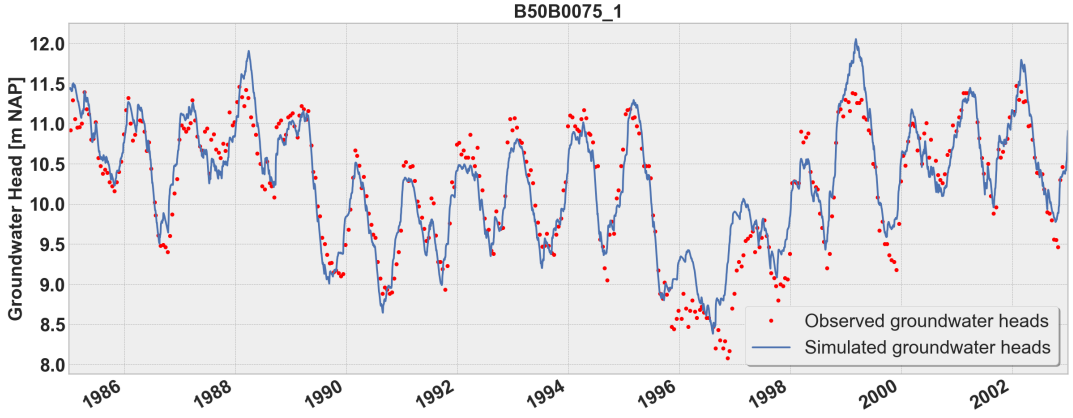


Figure B13. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_1$: $B50B0075_1$

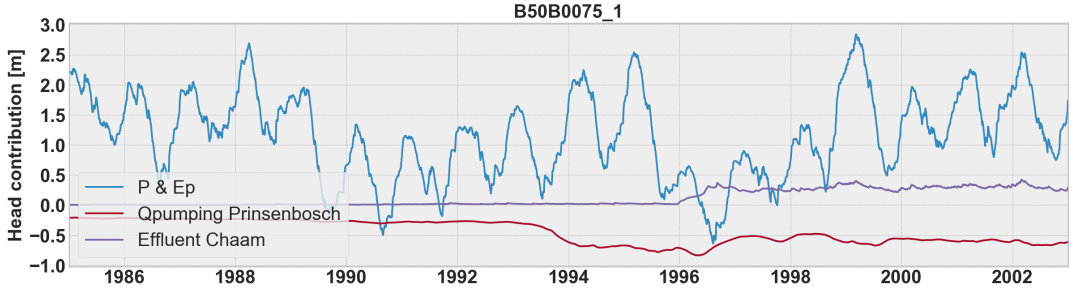


Figure B14. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_1$: $B50B0075_1$

```
Model Results B50B0075_1                                  Fit Statistics
================================================================================
nfev     35                       EVP                                  84.85
nobs     906                      R2                                    0.85
noise    True                     RMSE                                  0.27
tmin     1971-01-15 00:00:00      AIC                                  16.49
tmax     2010-01-01 00:00:00      BIC                                  69.39
freq     D                        ___
warmup   3650 days 00:00:00       ___
solver   LeastSquares             ___

Parameters (11 were optimized)
================================================================================
                                optimal    stderr     initial  vary
Rain & Evaporation_A           1.470609    ±3.55%    0.208808  True
Rain & Evaporation_n           1.161429    ±0.96%    1.000000  True
Rain & Evaporation_a         179.654304    ±4.44%   10.000000  True
Rain & Evaporation_f          -0.938772    ±3.16%   -1.000000  True
Qpumping Prinsenbosch_A       -0.000064   ±19.77%   -0.000201  True
Qpumping Prinsenbosch_rho      0.237858   ±85.95%    1.000000  True
Qpumping Prinsenbosch_cS    2106.659613  ±144.20%  100.000000  True
Effluent Chaam_A               0.000180   ±34.09%    0.001011  True
Effluent Chaam_a              74.619572   ±42.26%   10.000000  True
constant_d                     9.439611    ±0.93%   10.295529  True
noise_alpha                  115.552542    ±7.10%    1.000000  True

Parameter correlations |rho| > 0.5
================================================================================
Rain & Evaporation_A      Rain & Evaporation_a       0.71
                          constant_d                -0.60
Rain & Evaporation_n      Rain & Evaporation_a      -0.72
Rain & Evaporation_f      constant_d                -0.86
Qpumping Prinsenbosch_A   Qpumping Prinsenbosch_rho  0.77
                          Qpumping Prinsenbosch_cS  -0.84
                          Effluent Chaam_A          -0.92
                          Effluent Chaam_a          -0.72
Qpumping Prinsenbosch_rho Qpumping Prinsenbosch_cS  -0.97
                          Effluent Chaam_A          -0.70
                          Effluent Chaam_a          -0.54
Qpumping Prinsenbosch_cS  Effluent Chaam_A           0.77
                          Effluent Chaam_a           0.61
Effluent Chaam_A          Effluent Chaam_a           0.81
```

Figure B15. Fit report Pastas TSA $X1_1 : B50B0075_1$

B2.3 $X1_2 : B50B0101_1$

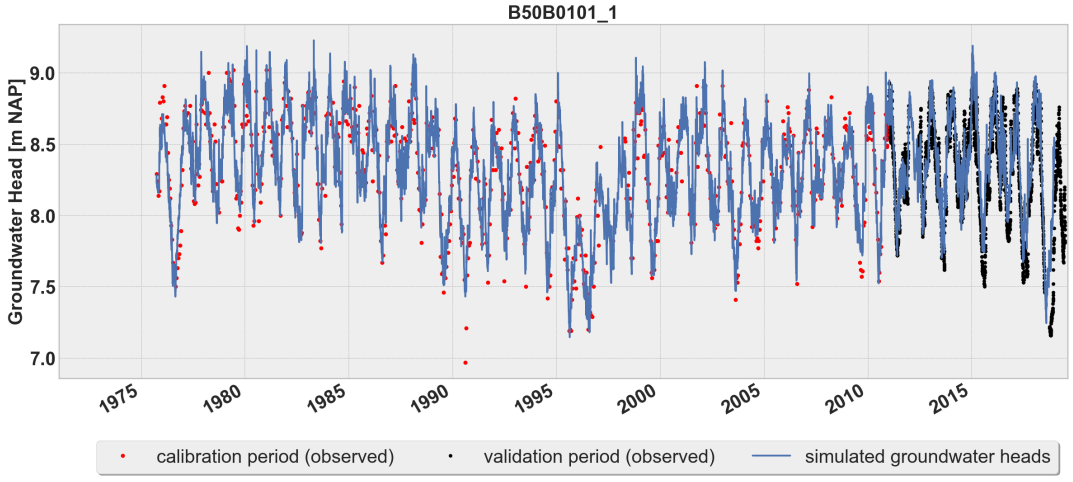


Figure B16. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_2$: $B50B00101_1$

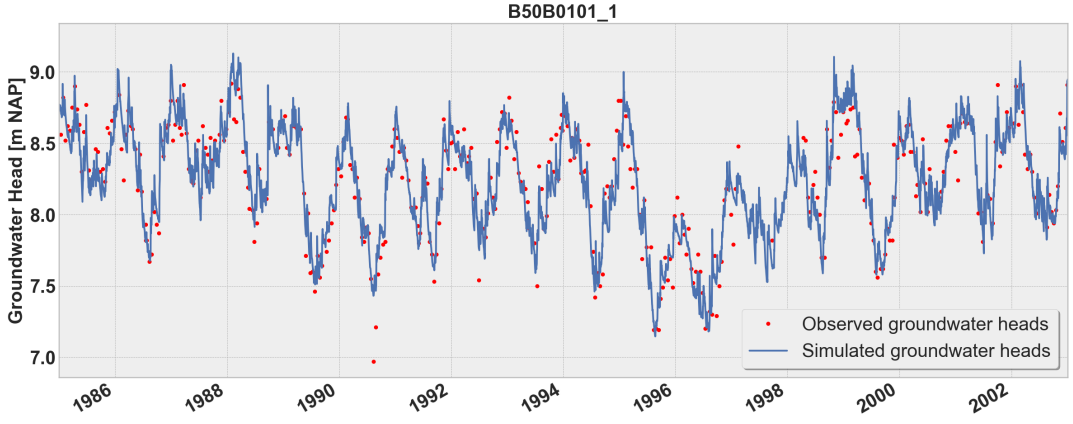


Figure B17. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_2$: $B50B0101_1$

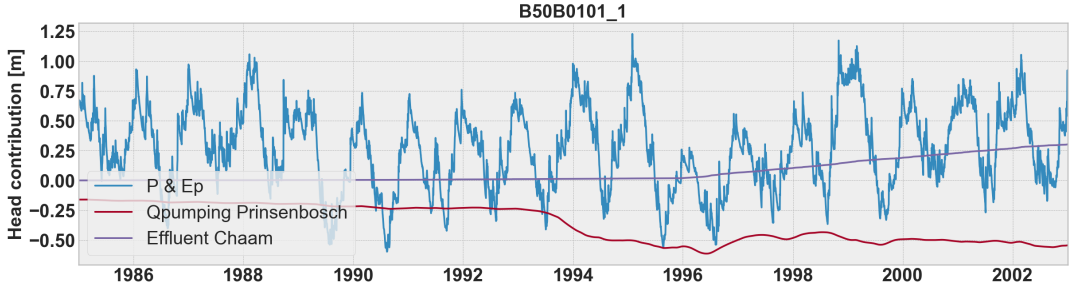


Figure B18. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_2$: $B50B0101_1$

Model Results B50B0101_1 Fit Statistics

nfev	30	EVP	88.85
nobs	786	R2	0.89
noise	True	RMSE	0.13
tmin	1975-09-29 00:00:00	AIC	17.45
tmax	2010-01-01 00:00:00	BIC	68.79
freq	D	___	
warmup	3650 days 00:00:00	___	
solver	LeastSquares	___	

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	0.407457	±2.41%	0.208808	True
Rain & Evaporation_n	0.782931	±1.13%	1.000000	True
Rain & Evaporation_a	124.039818	±4.49%	10.000000	True
Rain & Evaporation_f	-1.038484	±2.38%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000062	±23.13%	-0.000201	True
Qpumping Prinsenbosch_rho	0.185889	±80.96%	1.000000	True
Qpumping Prinsenbosch_cS	7616.594906	±143.43%	100.000000	True
Effluent Chaam_A	0.000448	±21.83%	0.001011	True
Effluent Chaam_a	4999.999993	±26.35%	10.000000	True
constant_d	8.258979	±0.24%	8.290922	True
noise_alpha	20.600378	±4.13%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.63
	constant_d	-0.64
Rain & Evaporation_n	Rain & Evaporation_a	-0.75
Rain & Evaporation_f	constant_d	-0.89
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.96
	Qpumping Prinsenbosch_cS	-0.98
	Effluent Chaam_A	-0.66
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.99
	Effluent Chaam_A	-0.71
Qpumping Prinsenbosch_cS	Effluent Chaam_A	0.70
Effluent Chaam_A	Effluent Chaam_a	0.76

Figure B19. Fit report Pastas TSA $X1_2 : B50B0101_1$

B2.4 $X1_3 : B50B0216_1$

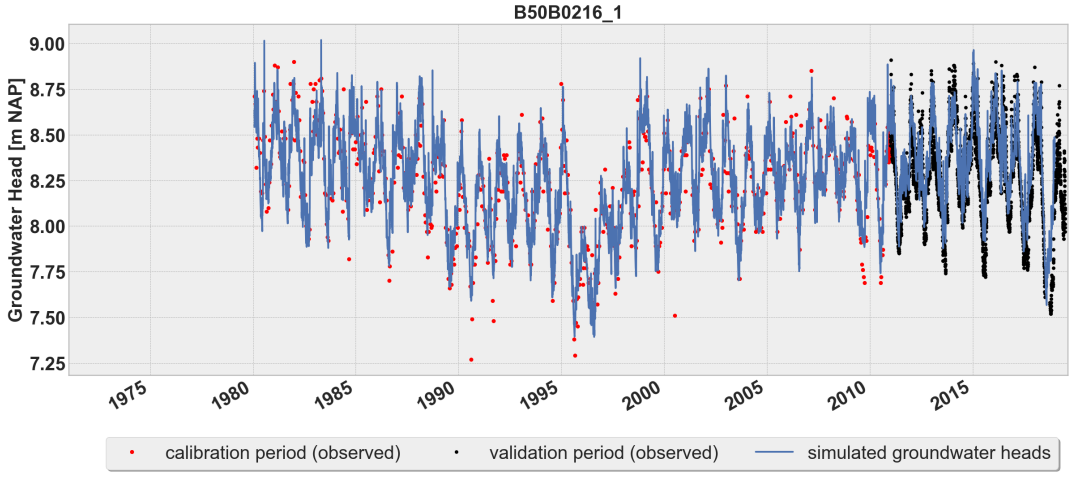


Figure B20. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_3$: $B50B0216_1$

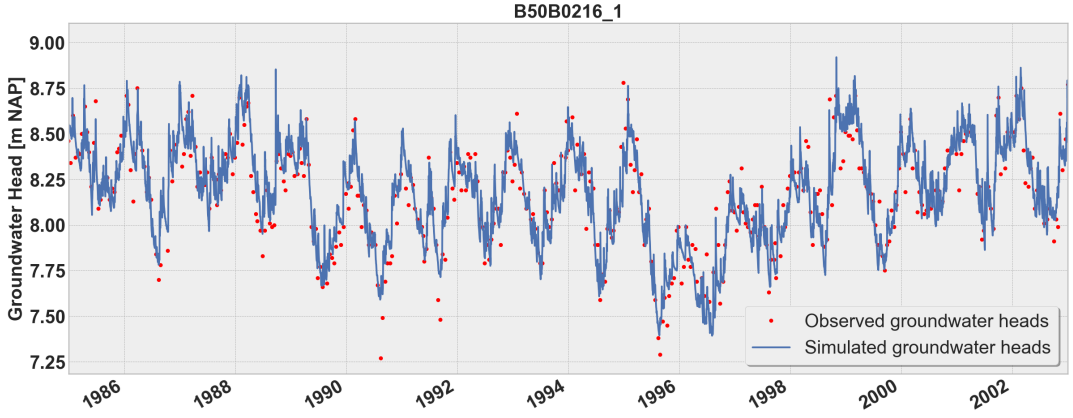


Figure B21. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_3$: $B50B0216_1$

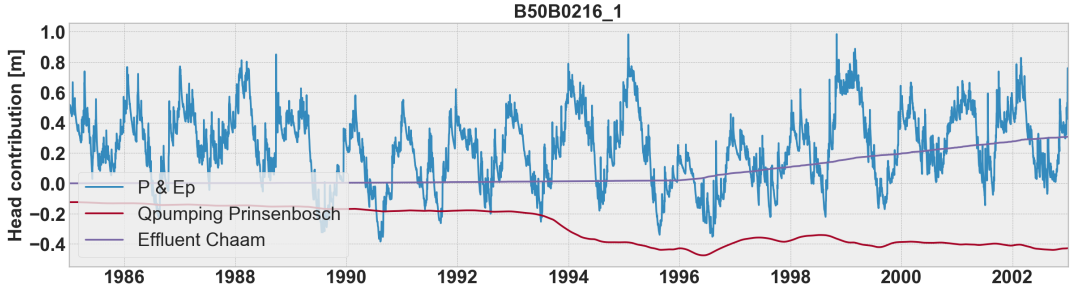


Figure B22. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_3$: $B50B0216_1$

```
Model Results B50B0216_1                                Fit Statistics
================================================================================
nfev     31                     EVP                                  83.65
nobs     704                    R2                                    0.84
noise    True                   RMSE                                  0.11
tmin     1980-01-28 00:00:00    AIC                                  17.92
tmax     2010-01-01 00:00:00    BIC                                  68.04
freq     D                      ___
warmup   3650 days 00:00:00     ___
solver   LeastSquares           ___

Parameters (11 were optimized)
================================================================================
                               optimal    stderr     initial  vary
Rain & Evaporation_A          0.316034    ±3.32%    0.208808  True
Rain & Evaporation_n          0.699926    ±1.30%    1.000000  True
Rain & Evaporation_a        164.883821    ±6.19%   10.000000  True
Rain & Evaporation_f         -0.982188    ±2.96%   -1.000000  True
Qpumping Prinsenbosch_A      -0.000050   ±31.40%   -0.000201  True
Qpumping Prinsenbosch_rho     0.163420  ±102.21%    1.000000  True
Qpumping Prinsenbosch_cS   9984.774659  ±186.49%  100.000000  True
Effluent Chaam_A              0.000402   ±17.42%    0.001011  True
Effluent Chaam_a           4198.738424   ±18.80%   10.000000  True
constant_d                    8.153002    ±0.26%    8.269179  True
noise_alpha                  15.979940    ±4.17%    1.000000  True

Parameter correlations |rho| > 0.5
================================================================================
Rain & Evaporation_A      Rain & Evaporation_a       0.68
                          constant_d                -0.72
Rain & Evaporation_n      Rain & Evaporation_a      -0.69
                          Rain & Evaporation_f       0.54
Rain & Evaporation_f      constant_d                -0.83
Qpumping Prinsenbosch_A   Qpumping Prinsenbosch_rho  0.96
                          Qpumping Prinsenbosch_cS  -0.99
                          Effluent Chaam_A          -0.81
Qpumping Prinsenbosch_rho Qpumping Prinsenbosch_cS  -0.99
                          Effluent Chaam_A          -0.85
Qpumping Prinsenbosch_cS  Effluent Chaam_A           0.85
Effluent Chaam_A          Effluent Chaam_a           0.71
```

Figure B23. Fit report Pastas TSA $X1_3 : B50B0216_1$

B2.5 $X1_4 : B50B0380_1$

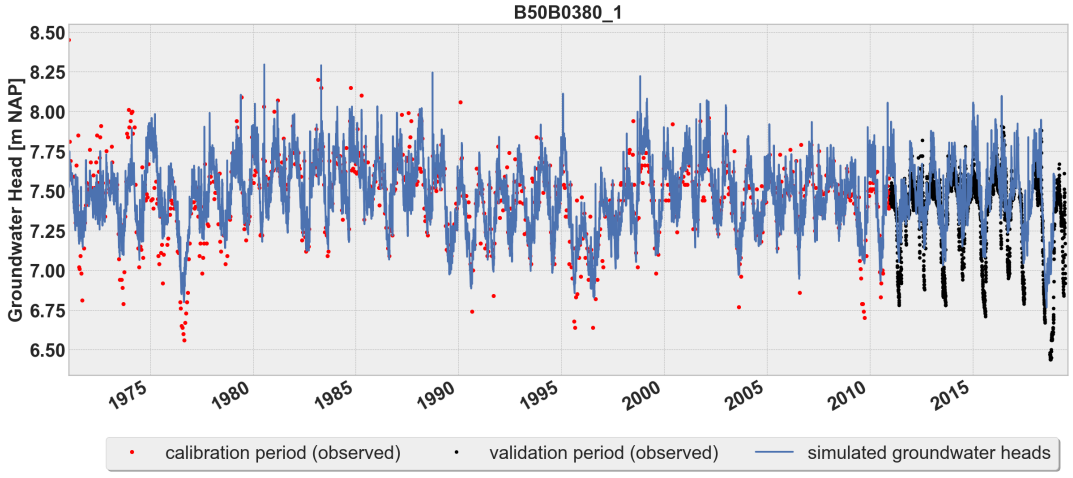


Figure B24. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_4$: $B50B0380_1$

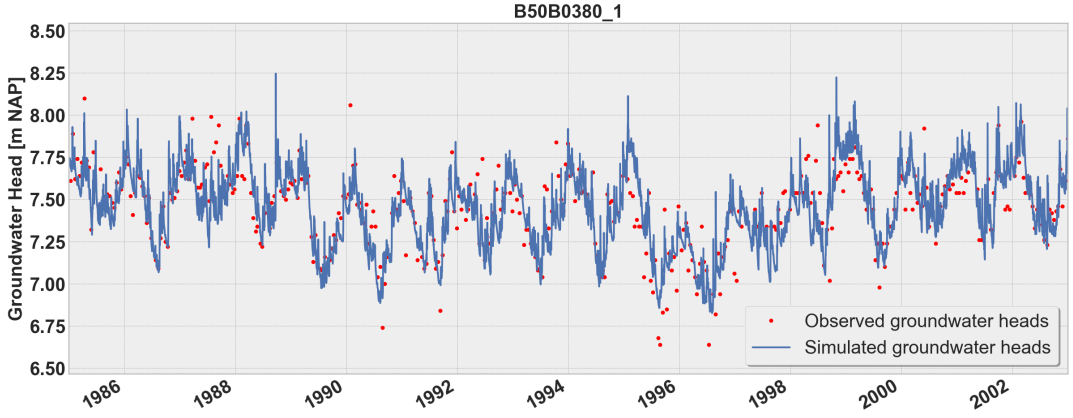


Figure B25. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_4$: $B50B0380_1$

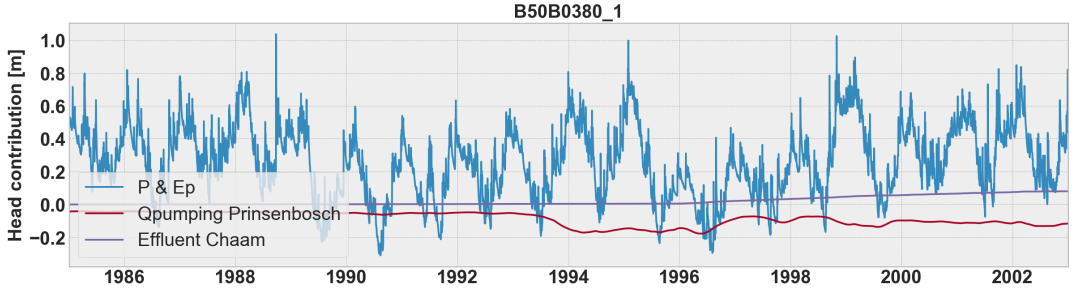


Figure B26. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_4$: $B50B0380_1$

```
Model Results B50B0380_1                                Fit Statistics
================================================================
nfev     38                     EVP                           66.96
nobs     896                    R2                             0.67
noise    True                   RMSE                           0.15
tmin     1971-01-15 00:00:00    AIC                           16.87
tmax     2010-01-01 00:00:00    BIC                           69.65
freq     D                      ___
warmup   3650 days 00:00:00     ___
solver   LeastSquares           ___

Parameters (11 were optimized)
================================================================
                              optimal    stderr     initial  vary
Rain & Evaporation_A         0.338392    ±4.69%    0.208808  True
Rain & Evaporation_n         0.621324    ±1.64%    1.000000  True
Rain & Evaporation_a       276.533087   ±10.28%   10.000000  True
Rain & Evaporation_f        -0.964031    ±4.35%   -1.000000  True
Qpumping Prinsenbosch_A     -0.000012   ±13.08%   -0.000201  True
Qpumping Prinsenbosch_rho    1.257874  ±108.85%    1.000000  True
Qpumping Prinsenbosch_cS   190.761311  ±120.32%  100.000000  True
Effluent Chaam_A             0.000067   ±25.82%    0.001011  True
Effluent Chaam_a          1953.027169   ±65.30%   10.000000  True
constant_d                   7.255612    ±0.36%    7.390087  True
noise_alpha                 24.081026    ±4.07%    1.000000  True

Parameter correlations |rho| > 0.5
================================================================
Rain & Evaporation_A      Rain & Evaporation_a      0.74
                          constant_d               -0.52
Rain & Evaporation_n      Rain & Evaporation_a     -0.68
                          Rain & Evaporation_f      0.51
Rain & Evaporation_f      constant_d               -0.89
Qpumping Prinsenbosch_rho Qpumping Prinsenbosch_cS -0.95
Effluent Chaam_A          Effluent Chaam_a          0.79
```

Figure B27. Fit report Pastas TSA $X1_4 : B50B0380_1$

B2.6 $X1_5 : B50E0140_1$

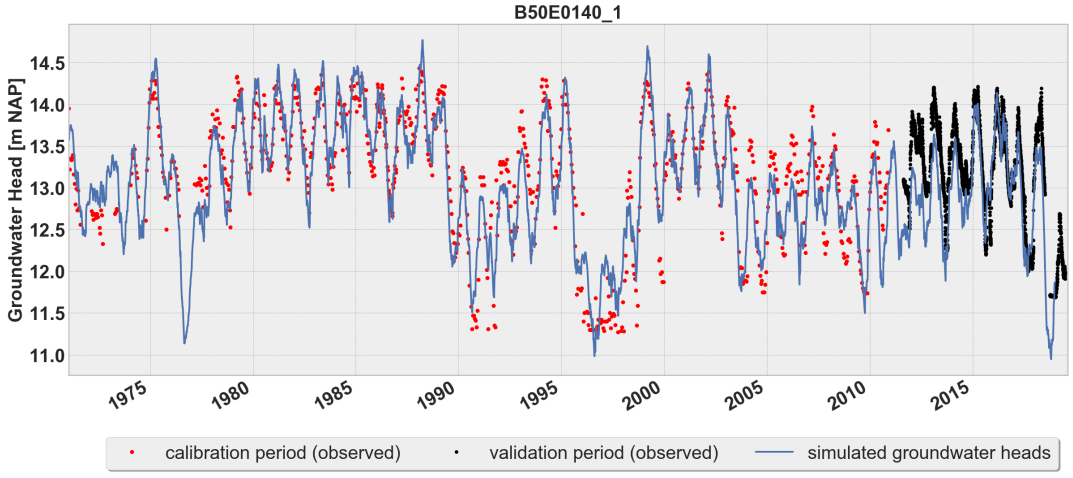


Figure B28. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X1_5$: $B50E0140_1$

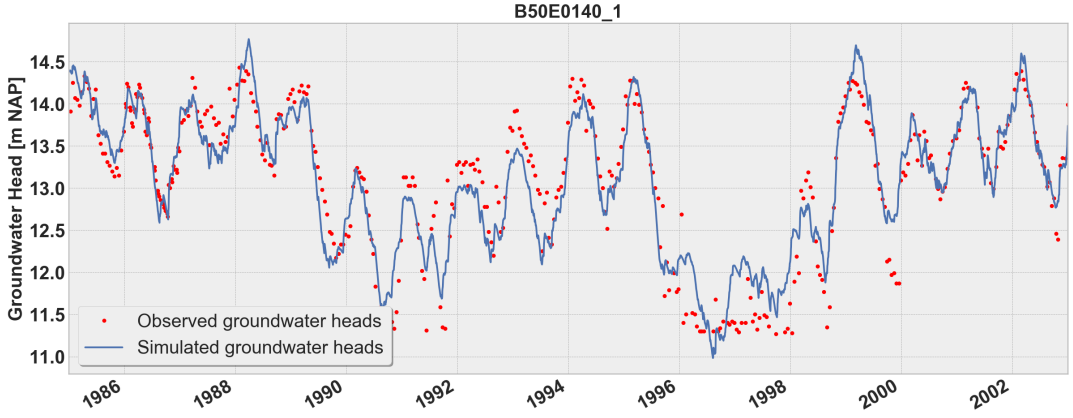


Figure B29. Observed and simulated groundwater heads for further research period 1985-2003, for well $X1_5$: $B50E0140_1$

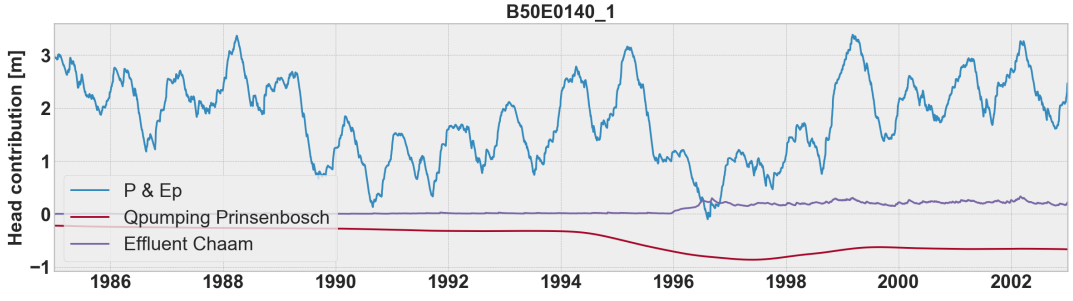


Figure B30. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X1_5$: $B50E0140_1$

Model Results B50E0140_1 Fit Statistics

nfev	29	EVP	85.13
nobs	856	R2	0.85
noise	True	RMSE	0.29
tmin	1971-01-15 00:00:00	AIC	15.27
tmax	2010-01-01 00:00:00	BIC	67.54
freq	D		
warmup	3650 days 00:00:00		
solver	LeastSquares		

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	1.972257	±3.22%	0.208808	True
Rain & Evaporation_n	1.139278	±1.28%	1.000000	True
Rain & Evaporation_a	255.912018	±5.02%	10.000000	True
Rain & Evaporation_f	-0.855451	±4.25%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000071	±13.49%	-0.000201	True
Qpumping Prinsenbosch_rho	3.558551	±46.97%	1.000000	True
Qpumping Prinsenbosch_cS	500.397672	±51.38%	100.000000	True
Effluent Chaam_A	0.000133	±39.45%	0.001011	True
Effluent Chaam_a	55.155073	±56.39%	10.000000	True
constant_d	11.658396	±1.13%	13.166809	True
noise_alpha	62.367707	±5.66%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.55
	constant_d	-0.56
Rain & Evaporation_n	Rain & Evaporation_a	-0.76
Rain & Evaporation_f	constant_d	-0.90
Qpumping Prinsenbosch_A	Effluent Chaam_A	-0.91
	Effluent Chaam_a	-0.63
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.96
Effluent Chaam_A	Effluent Chaam_a	0.72

Figure B31. Fit report Pastas TSA $X1_5 : B50E0140_1$

B3 Simulated groundwater time series Pastas TSA - screen-2 wells

In this Appendix, the results of the simulated groundwater head time series with Pastas TSA for the screen-2 wells are depicted.

B3.1 $X2_0 : B50B0074_2$

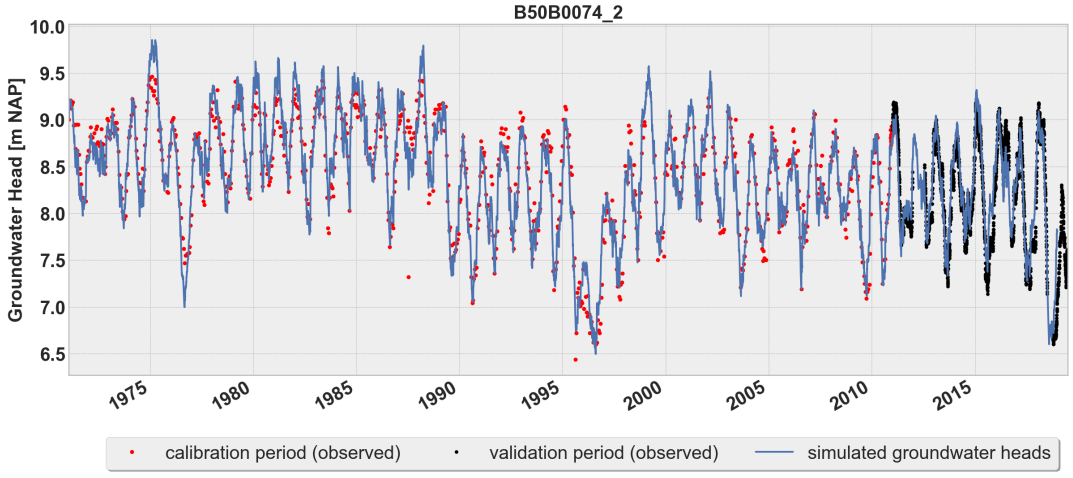


Figure B32. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X2_0$: $B50B0074_2$

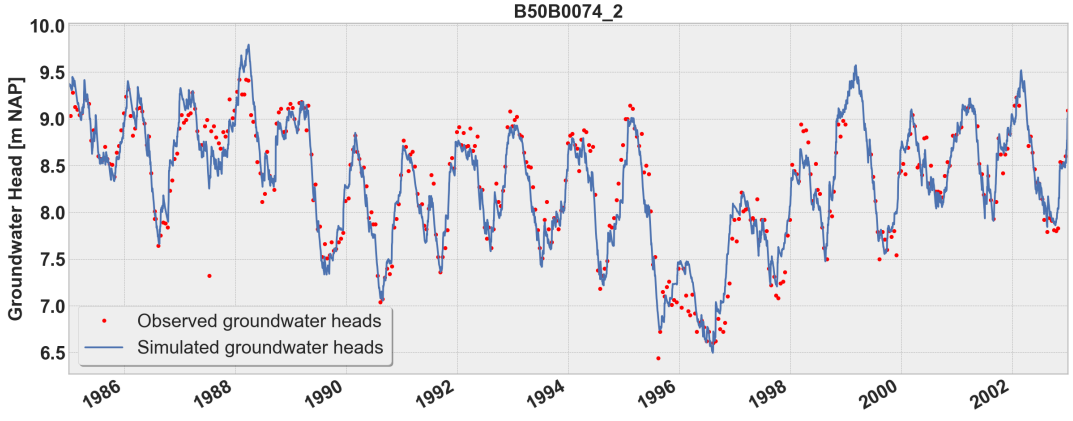


Figure B33. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_0$: $B50B0074_2$

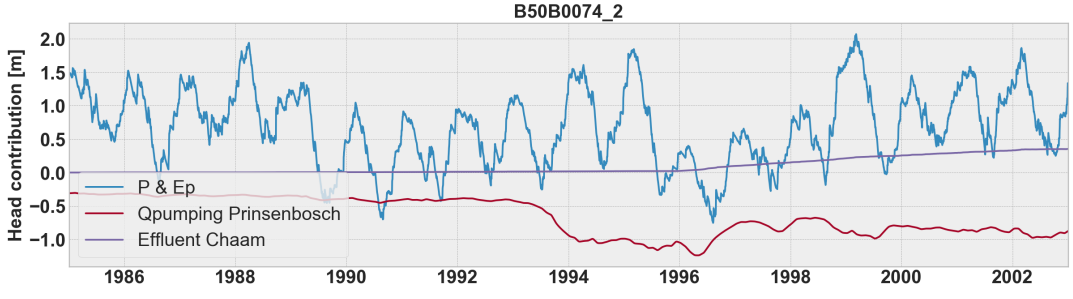


Figure B34. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_0$: $B50B0074_2$

Model Results B50B0074_2 Fit Statistics

nfev	33	EVP	91.81
nobs	900	R2	0.92
noise	True	RMSE	0.17
tmin	1971-01-15 00:00:00	AIC	16.93
tmax	2010-01-01 00:00:00	BIC	69.75
freq	D		
warmup	3650 days 00:00:00		
solver	LeastSquares		

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	1.007488	±2.43%	0.208808	True
Rain & Evaporation_n	0.990238	±0.87%	1.000000	True
Rain & Evaporation_a	178.929231	±3.81%	10.000000	True
Rain & Evaporation_f	-1.013019	±2.54%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000092	±6.54%	-0.000201	True
Qpumping Prinsenbosch_rho	0.234179	±37.05%	1.000000	True
Qpumping Prinsenbosch_cS	1778.019141	±56.93%	100.000000	True
Effluent Chaam_A	0.000276	±11.97%	0.001011	True
Effluent Chaam_a	1734.906947	±26.59%	10.000000	True
constant_d	8.192141	±0.57%	8.220625	True
noise_alpha	46.698698	±5.01%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.64
	constant_d	-0.52
Rain & Evaporation_n	Rain & Evaporation_a	-0.75
	Rain & Evaporation_f	0.50
Rain & Evaporation_f	constant_d	-0.91
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.81
	Qpumping Prinsenbosch_cS	-0.89
	Effluent Chaam_A	-0.70
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.96
	Effluent Chaam_A	-0.62
Qpumping Prinsenbosch_cS	Effluent Chaam_A	0.66

Figure B35. Fit report Pastas TSA $X2_0 : B50B0074_2$

B3.2 $X2_1 : B50B0075_2$

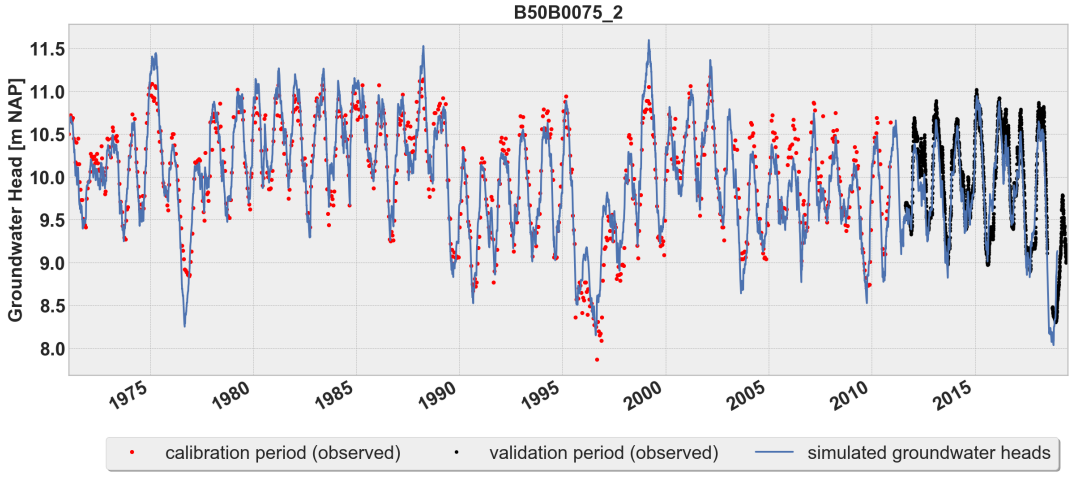


Figure B36. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X2_1$: $B50B0075_2$

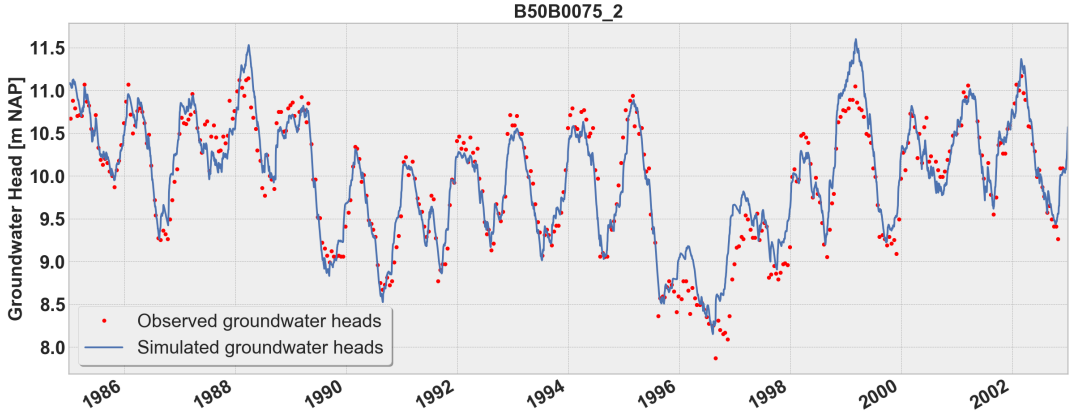


Figure B37. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_1$: $B50B0075_2$

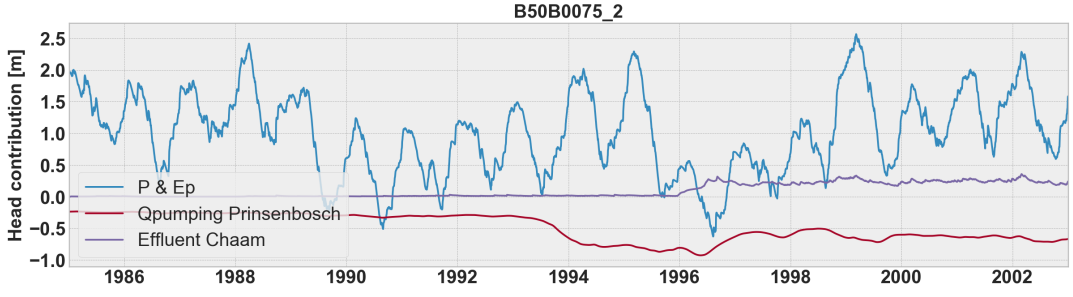


Figure B38. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_1$: $B50B0075_2$

Model Results B50B0075_2 Fit Statistics

nfev	40	EVP	88.84
nobs	909	R2	0.89
noise	True	RMSE	0.22
tmin	1971-01-15 00:00:00	AIC	16.67
tmax	2010-01-01 00:00:00	BIC	69.60
freq	D	___	
warmup	3650 days 00:00:00	___	
solver	LeastSquares	___	

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	1.268728	±2.86%	0.208808	True
Rain & Evaporation_n	1.128503	±0.97%	1.000000	True
Rain & Evaporation_a	170.035639	±4.01%	10.000000	True
Rain & Evaporation_f	-0.934037	±3.05%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000068	±11.15%	-0.000201	True
Qpumping Prinsenbosch_rho	0.471669	±54.03%	1.000000	True
Qpumping Prinsenbosch_cS	997.980269	±79.68%	100.000000	True
Effluent Chaam_A	0.000147	±27.41%	0.001011	True
Effluent Chaam_a	63.599562	±37.86%	10.000000	True
constant_d	9.358629	±0.73%	9.919234	True
noise_alpha	74.484463	±5.87%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.64
	constant_d	-0.56
Rain & Evaporation_n	Rain & Evaporation_a	-0.74
Rain & Evaporation_f	constant_d	-0.89
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.71
	Qpumping Prinsenbosch_cS	-0.81
	Effluent Chaam_A	-0.91
	Effluent Chaam_a	-0.62
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.96
	Effluent Chaam_A	-0.66
Qpumping Prinsenbosch_cS	Effluent Chaam_A	0.76
	Effluent Chaam_a	0.52
Effluent Chaam_A	Effluent Chaam_a	0.71

Figure B39. Fit report Pastas TSA $X2_1 : B50B0075_2$

B3.3 $X2_2 : B50B0101_2$

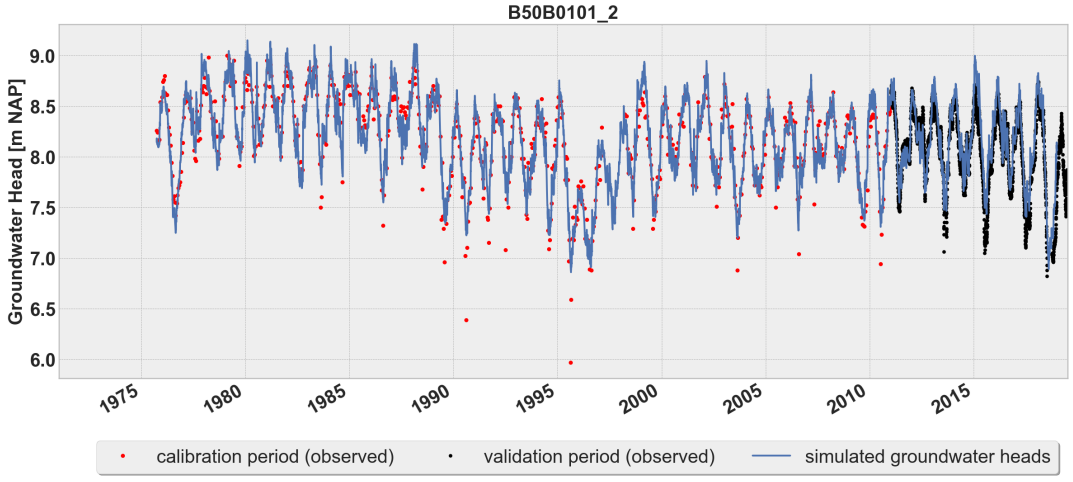


Figure B40. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X2_2$: $B50B0101_2$

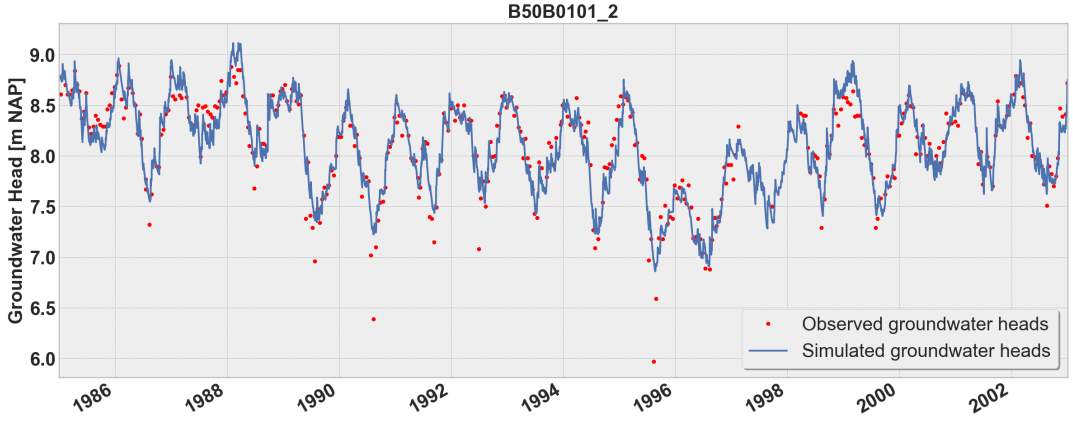


Figure B41. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_2$: $B50B0101_2$

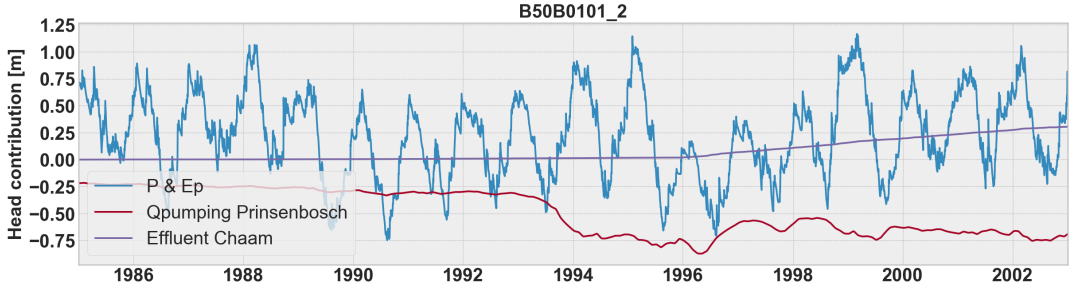


Figure B42. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_2$: $B50B0101_2$

Model Results B50B0101_2 Fit Statistics

nfev	32	EVP	91.02
nobs	785	R2	0.91
noise	True	RMSE	0.13
tmin	1975-09-29 00:00:00	AIC	17.44
tmax	2010-01-01 00:00:00	BIC	68.77
freq	D		
warmup	3650 days 00:00:00		
solver	LeastSquares		

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	0.509624	±2.26%	0.208808	True
Rain & Evaporation_n	0.854889	±1.07%	1.000000	True
Rain & Evaporation_a	149.615123	±4.02%	10.000000	True
Rain & Evaporation_f	-1.180532	±2.42%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000081	±21.38%	-0.000201	True
Qpumping Prinsenbosch_rho	0.090246	±85.31%	1.000000	True
Qpumping Prinsenbosch_cS	9977.615363	±155.81%	100.000000	True
Effluent Chaam_A	0.000392	±19.93%	0.001011	True
Effluent Chaam_a	4081.441816	±23.30%	10.000000	True
constant_d	8.297390	±0.30%	8.089305	True
noise_alpha	24.075412	±4.08%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.55
	constant_d	-0.53
Rain & Evaporation_n	Rain & Evaporation_a	-0.78
	Rain & Evaporation_f	0.52
Rain & Evaporation_f	constant_d	-0.93
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.97
	Qpumping Prinsenbosch_cS	-0.99
	Effluent Chaam_A	-0.74
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.99
	Effluent Chaam_A	-0.76
Qpumping Prinsenbosch_cS	Effluent Chaam_A	0.76
Effluent Chaam_A	Effluent Chaam_a	0.67

Figure B43. Fit report Pastas TSA $X2_2 : B50B0101_2$

B3.4 $X2_3 : B50B0216_2$

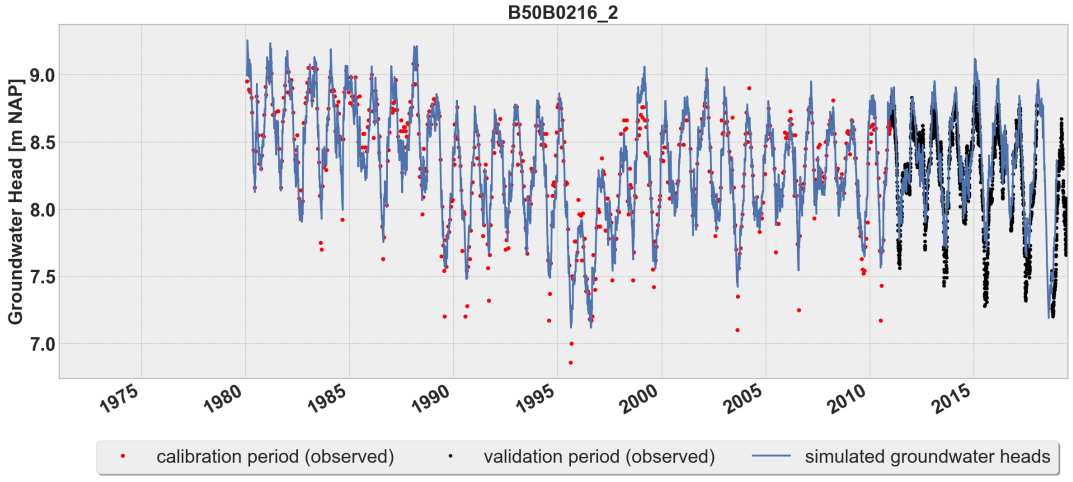


Figure B44. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X2_3$: $B50B0216_2$

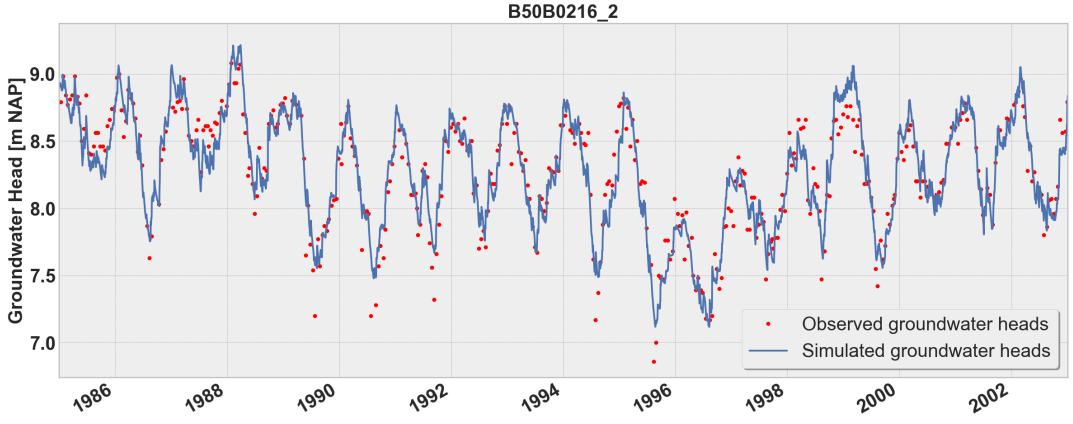


Figure B45. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_3$: $B50B0216_2$

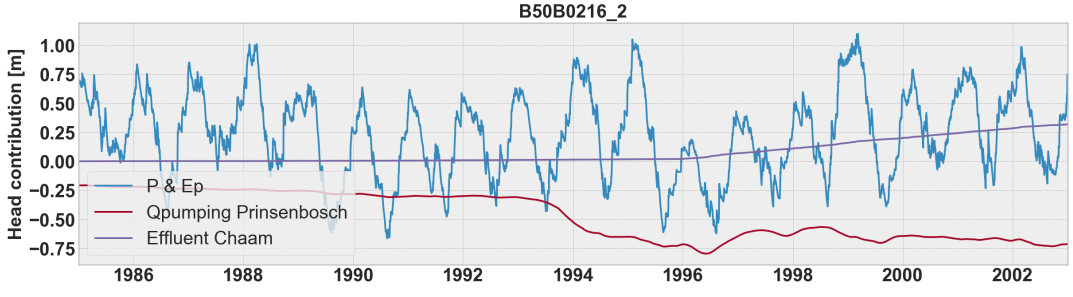


Figure B46. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_3$: $B50B0216_2$

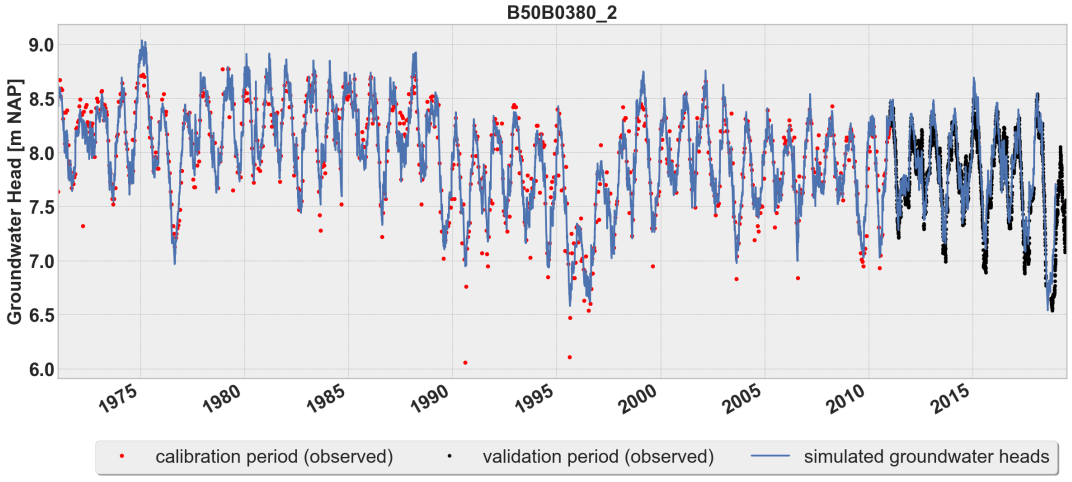
```
Model Results B50B0216_2                                Fit Statistics
================================================================
nfev     39                     EVP                           90.57
nobs     701                    R2                             0.91
noise    True                   RMSE                           0.13
tmin     1980-01-28 00:00:00    AIC                           17.63
tmax     2010-01-01 00:00:00    BIC                           67.71
freq     D                      ___
warmup   3650 days 00:00:00     ___
solver   LeastSquares           ___

Parameters (11 were optimized)
================================================================
                             optimal    stderr     initial  vary
Rain & Evaporation_A        0.447141    ±2.72%    0.208808  True
Rain & Evaporation_n        0.951551    ±1.24%    1.000000  True
Rain & Evaporation_a      111.033540    ±4.16%   10.000000  True
Rain & Evaporation_f       -1.122872    ±2.59%   -1.000000  True
Qpumping Prinsenbosch_A    -0.000083   ±23.70%   -0.000201  True
Qpumping Prinsenbosch_rho   0.158523   ±78.45%    1.000000  True
Qpumping Prinsenbosch_cS 9999.999947  ±142.86%  100.000000  True
Effluent Chaam_A            0.000473   ±22.58%    0.001011  True
Effluent Chaam_a         5000.000000   ±26.30%   10.000000  True
constant_d                  8.445102    ±0.33%    8.269346  True
noise_alpha                20.548135    ±4.26%    1.000000  True

Parameter correlations |rho| > 0.5
================================================================
Rain & Evaporation_A      Rain & Evaporation_a       0.57
                          Rain & Evaporation_f       0.54
                          constant_d                -0.73
Rain & Evaporation_n      Rain & Evaporation_a      -0.78
Rain & Evaporation_f      constant_d                -0.89
Qpumping Prinsenbosch_A   Qpumping Prinsenbosch_rho  0.96
                          Qpumping Prinsenbosch_cS  -0.99
                          Effluent Chaam_A          -0.71
Qpumping Prinsenbosch_rho Qpumping Prinsenbosch_cS  -0.99
                          Effluent Chaam_A          -0.78
Qpumping Prinsenbosch_cS  Effluent Chaam_A           0.76
Effluent Chaam_A          Effluent Chaam_a           0.76
```

Figure B47. Fit report Pastas TSA $X2_3 : B50B0216_2$

B3.5 $X2_4 : B50B0380_2$



and simulated groundwater heads with a division in calibration and validation period, for well $X2_4$: $B50B0380_2$

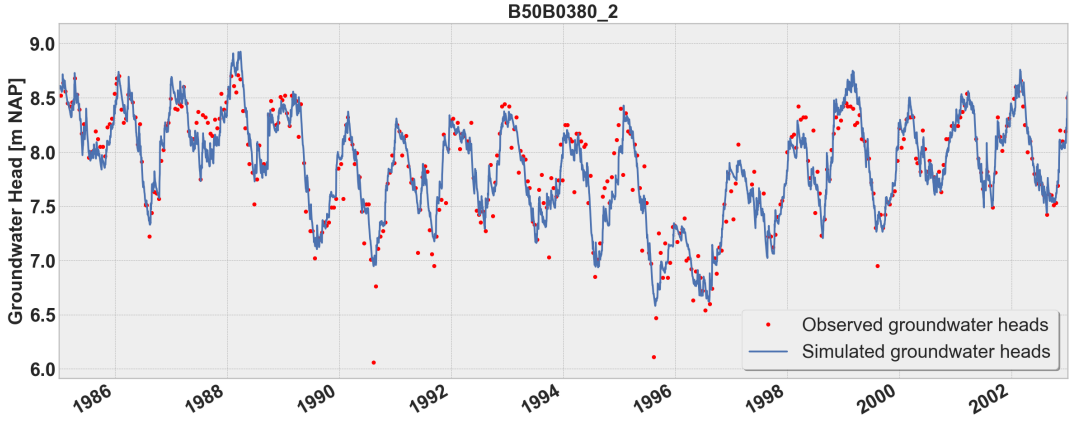


Figure B48. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_4$: $B50B0380_2$

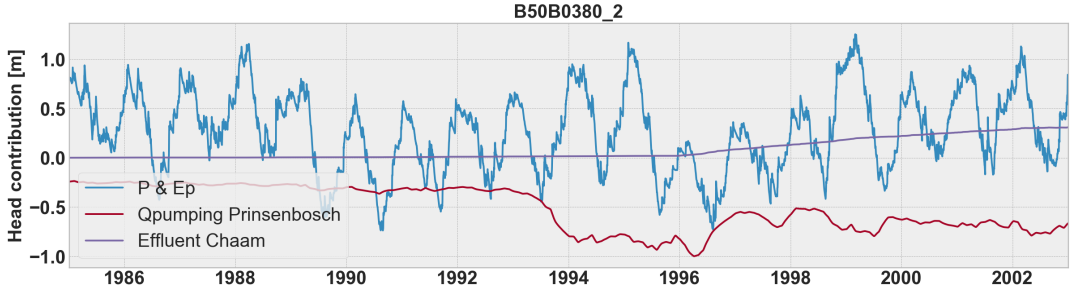


Figure B49. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_4$: $B50B0380_2$

Model Results B50B0380_2 Fit Statistics

nfev	33	EVP	89.72
nobs	884	R2	0.90
noise	True	RMSE	0.15
tmin	1971-01-15 00:00:00	AIC	16.42
tmax	2010-01-01 00:00:00	BIC	69.05
freq	D		
warmup	3650 days 00:00:00		
solver	LeastSquares		

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	0.607230	±1.91%	0.208808	True
Rain & Evaporation_n	0.878244	±1.14%	1.000000	True
Rain & Evaporation_a	178.455236	±4.09%	10.000000	True
Rain & Evaporation_f	-1.173146	±2.37%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000073	±4.25%	-0.000201	True
Qpumping Prinsenbosch_rho	0.158550	±28.69%	1.000000	True
Qpumping Prinsenbosch_cS	2016.433879	±41.94%	100.000000	True
Effluent Chaam_A	0.000246	±6.90%	0.001011	True
Effluent Chaam_a	1815.771563	±14.21%	10.000000	True
constant_d	8.037712	±0.34%	7.817342	True
noise_alpha	13.830117	±4.11%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.52
Rain & Evaporation_n	Rain & Evaporation_a	-0.83
	Rain & Evaporation_f	0.55
Rain & Evaporation_a	Rain & Evaporation_f	-0.57
Rain & Evaporation_f	constant_d	-0.96
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.81
	Qpumping Prinsenbosch_cS	-0.91
	Effluent Chaam_A	-0.72
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.95
	Effluent Chaam_A	-0.64
Qpumping Prinsenbosch_cS	Effluent Chaam_A	0.69

Figure B50. Fit report Pastas TSA $X2_4 : B50B0380_2$

B3.6 $X2_5$: $B50E0140_2$

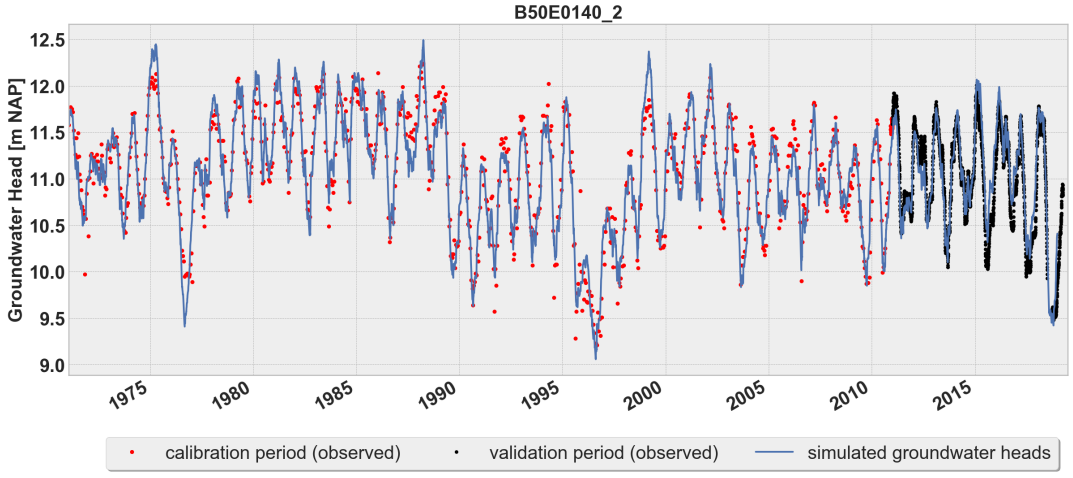


Figure B51. Observed and simulated groundwater heads with a division in calibration and validation period, for well $X2_5$: $B50B0140_2$

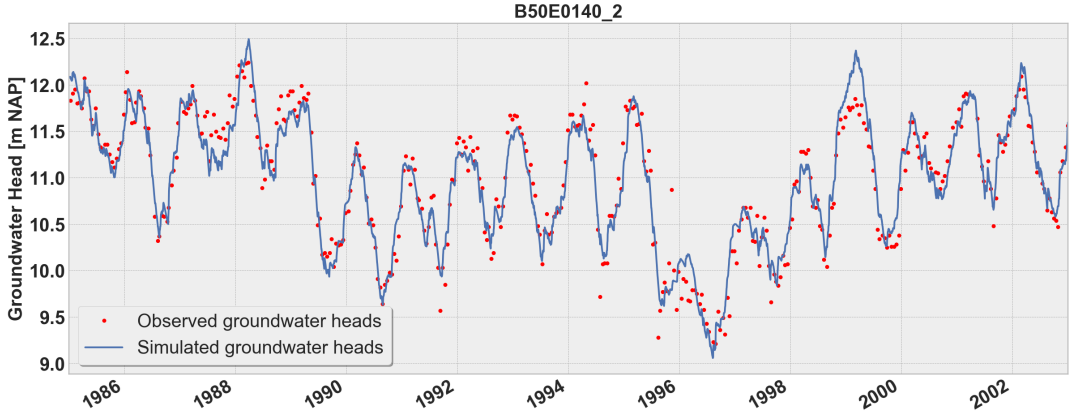


Figure B52. Observed and simulated groundwater heads for further research period 1985-2003, for well $X2_5$: $B50E0140_2$

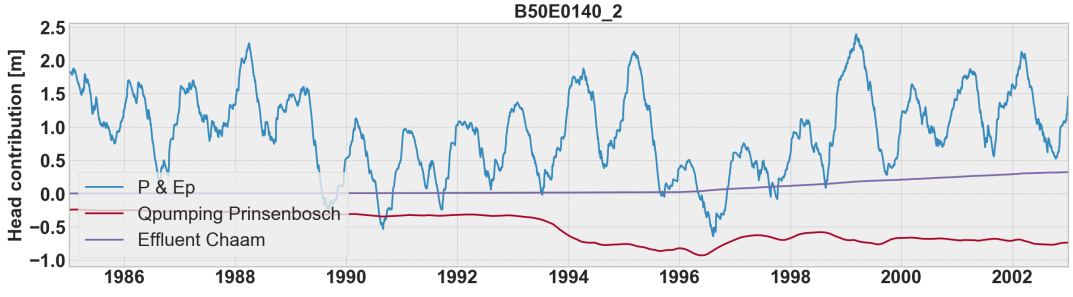


Figure B53. Contributions of the explanatory variables to the groundwater heads during the research period, for well $X2_5$: $B50E0140_2$

Model Results B50E0140_2 Fit Statistics

nfev	35	EVP	91.69
nobs	913	R2	0.92
noise	True	RMSE	0.18
tmin	1971-01-15 00:00:00	AIC	16.52
tmax	2010-01-01 00:00:00	BIC	69.51
freq	D	___	
warmup	3650 days 00:00:00	___	
solver	LeastSquares	___	

Parameters (11 were optimized)

	optimal	stderr	initial	vary
Rain & Evaporation_A	1.216437	±2.05%	0.208808	True
Rain & Evaporation_n	1.131923	±0.95%	1.000000	True
Rain & Evaporation_a	174.468319	±3.38%	10.000000	True
Rain & Evaporation_f	-0.961532	±2.62%	-1.000000	True
Qpumping Prinsenbosch_A	-0.000076	±10.17%	-0.000201	True
Qpumping Prinsenbosch_rho	0.313830	±51.12%	1.000000	True
Qpumping Prinsenbosch_cS	2297.612145	±78.58%	100.000000	True
Effluent Chaam_A	0.000395	±21.34%	0.001011	True
Effluent Chaam_a	3852.028324	±44.03%	10.000000	True
constant_d	10.505033	±0.51%	11.011405	True
noise_alpha	40.607430	±4.58%	1.000000	True

Parameter correlations |rho| > 0.5

Rain & Evaporation_A	Rain & Evaporation_a	0.54
	constant_d	-0.53
Rain & Evaporation_n	Rain & Evaporation_a	-0.80
Rain & Evaporation_f	constant_d	-0.93
Qpumping Prinsenbosch_A	Qpumping Prinsenbosch_rho	0.85
	Qpumping Prinsenbosch_cS	-0.92
	Effluent Chaam_a	0.56
Qpumping Prinsenbosch_rho	Qpumping Prinsenbosch_cS	-0.97
Effluent Chaam_A	Effluent Chaam_a	0.79

Figure B54. Fit report Pastas TSA $X2_5 : B50E0140_2$

Appendix C: Model setup 1

C1 Dataset for model setup 1

In this Appendix, the time series of the input variable $X1_0$ and the target $Qobs$ for model setup 1 are visualised. A division is made between the training set and the test set.

C1.1 Input variable for model setup 1

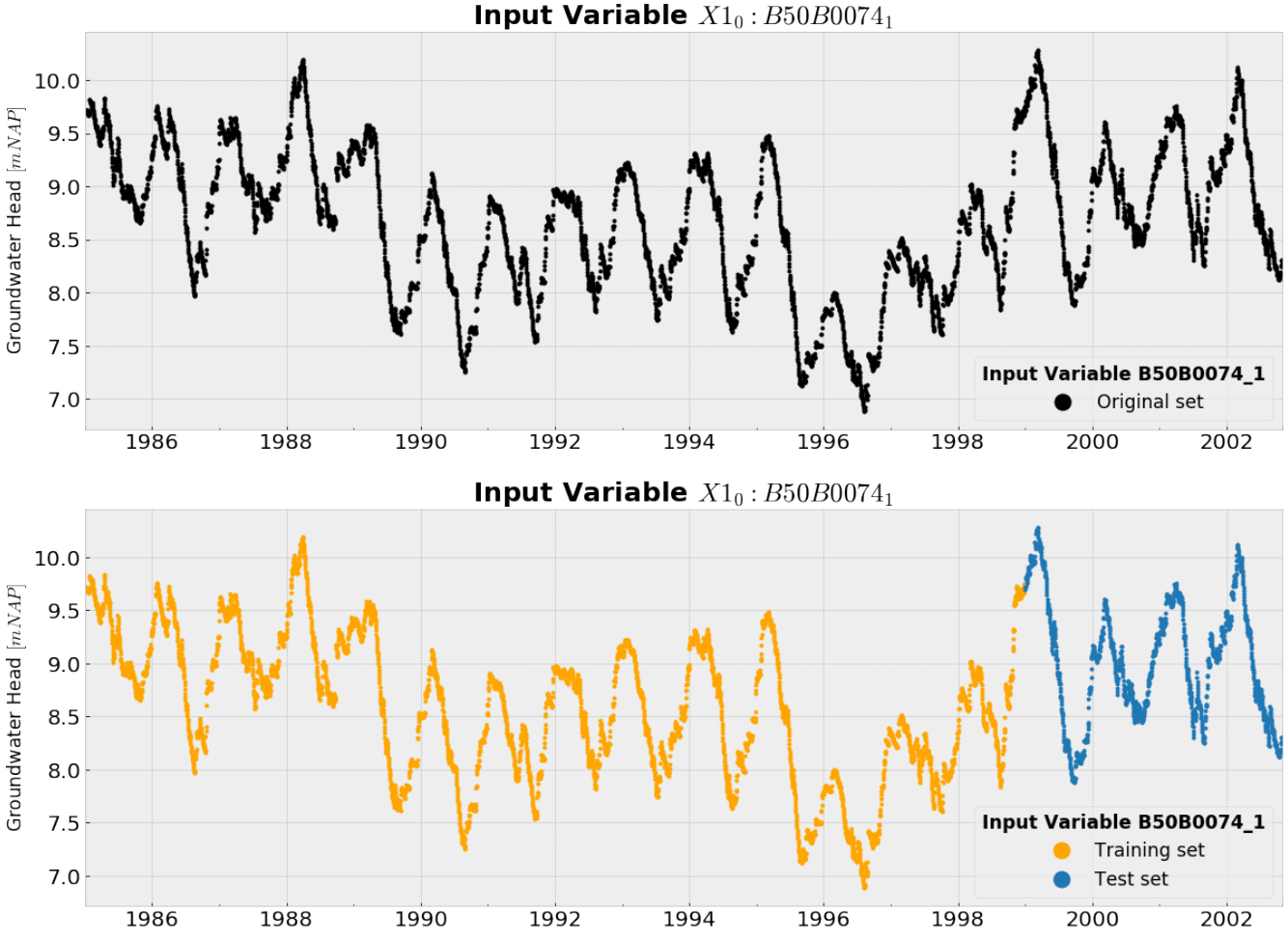


Figure C1. The time series of the input variable well B50B0074_1 for model setup 1, divided into the training set (1985-1999) and the test set (1999-2003)

C1.2 Target for model setup 1

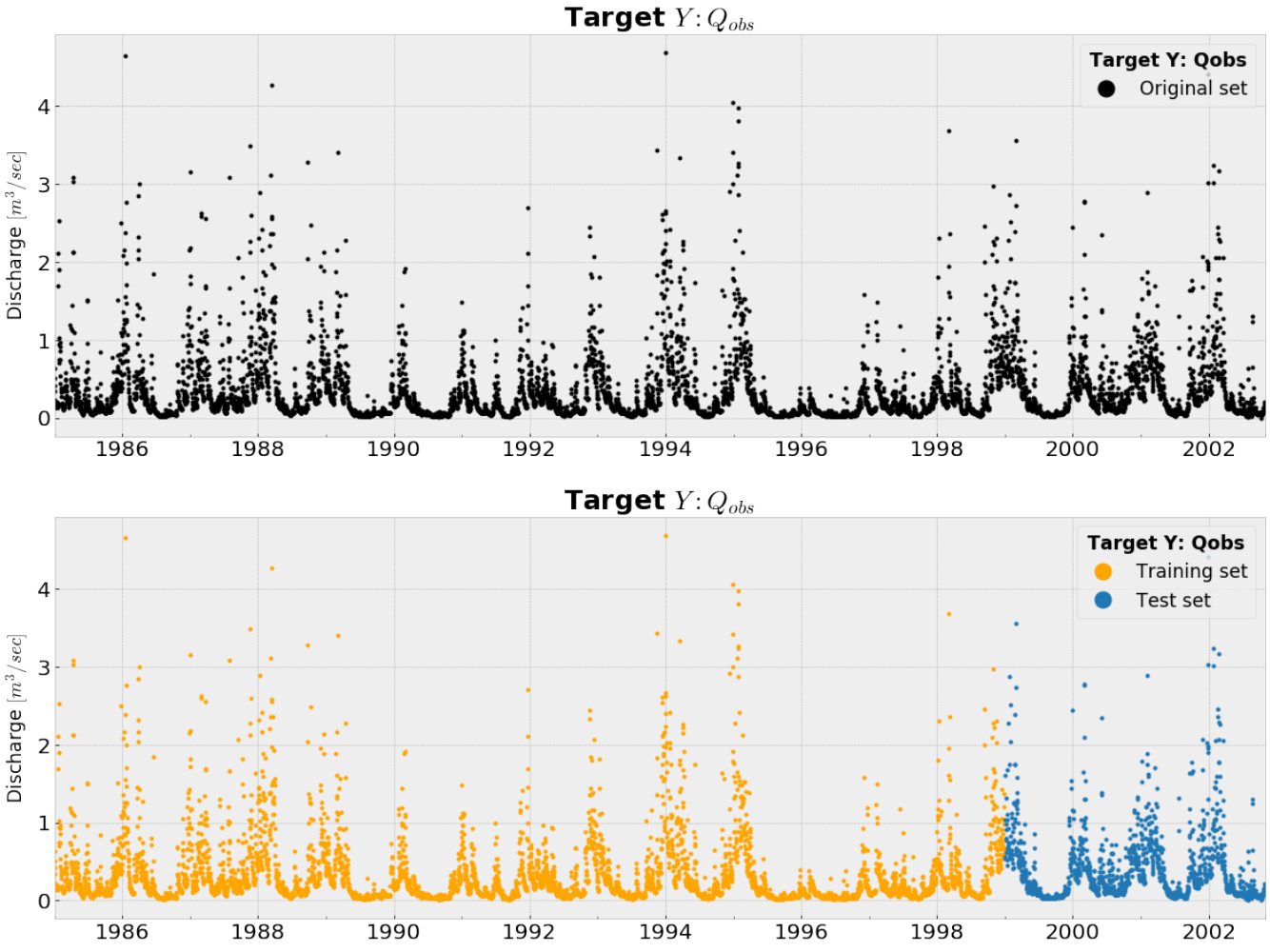


Figure C2. The time series of the target $Qobs$ for model setup 1, divided into the training set (1985-1999) and the test set (1999-2003)

C2 Results - model setup 1

In this Appendix, the results of the different machine learning algorithms of model setup 1 are separately visualised. First, the $Qsim$ time series is plotted for the training and test, followed by a plot of zooming in on the test set. The last figure of each machine learning algorithm is a scatterplot of $Qobs$ against $Qsim$ to easlity detect over- or underfitting.

C2.1 Results DTR - model setup 1

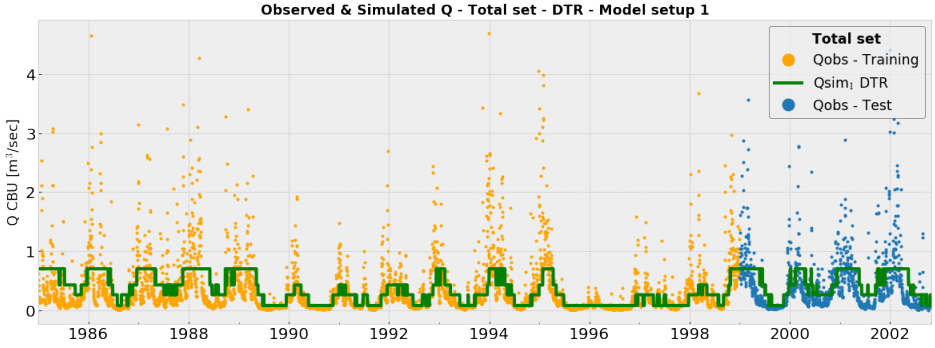


Figure C3. The time series of $Qsim$ and $Qobs$ for the training and test set, for DTR - model setup 1

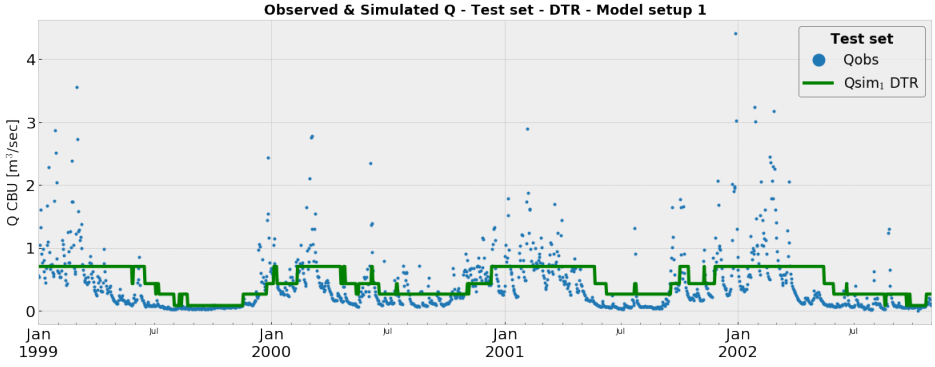


Figure C4. The time series of $Qsim$ and $Qobs$ for only the test set, for DTR - model setup 1

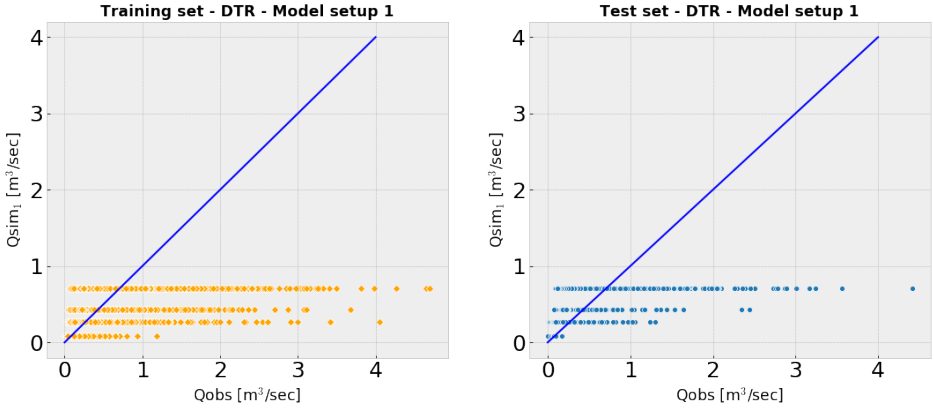


Figure C5. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for DTR - model setup 1. Underfitting is occuring when using the machine learning algorithm DTR for model setup 1.

C2.2 Results RFR - model setup 1

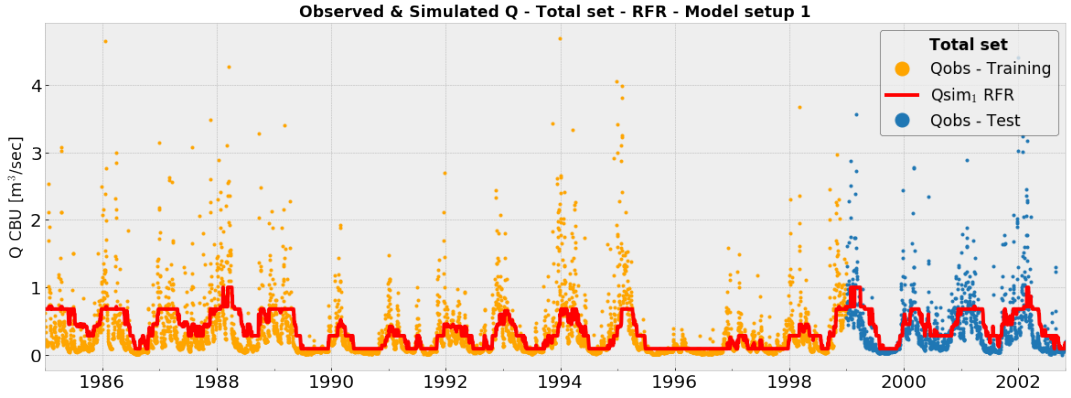


Figure C6. The time series of $Qsim$ and $Qobs$ for the training and test set, for RFR - model setup 1

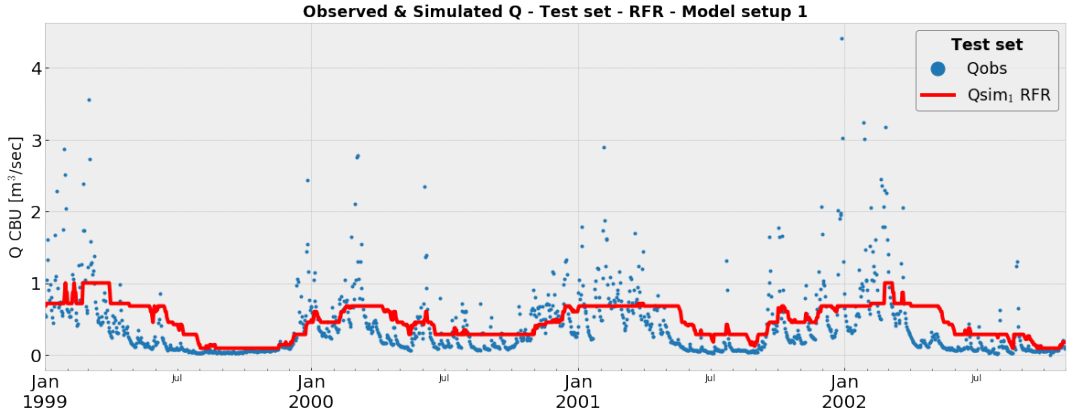


Figure C7. The time series of $Qsim$ and $Qobs$ for only the test set, for RFR - model setup 1. The results do not differ a lot from the result of DTR.

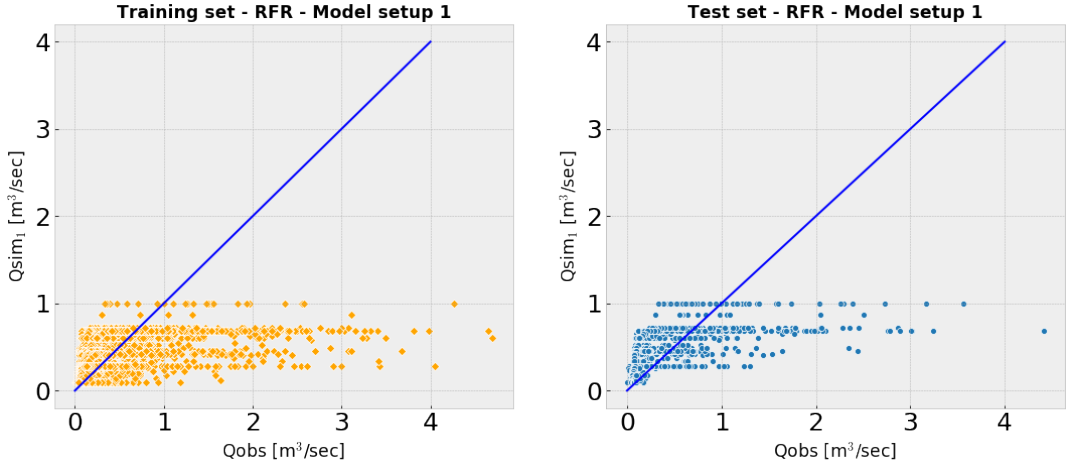


Figure C8. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for RFR - model setup 1. Underfitting is occuring.

C2.3 Results GBR - model setup 1

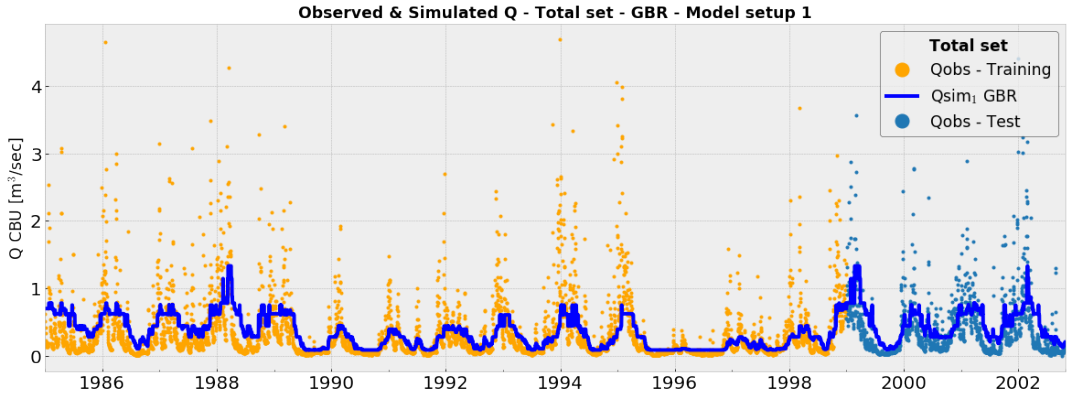


Figure C9. The time series of $Qsim$ and $Qobs$ for the training and test set, for GBR - model setup 1

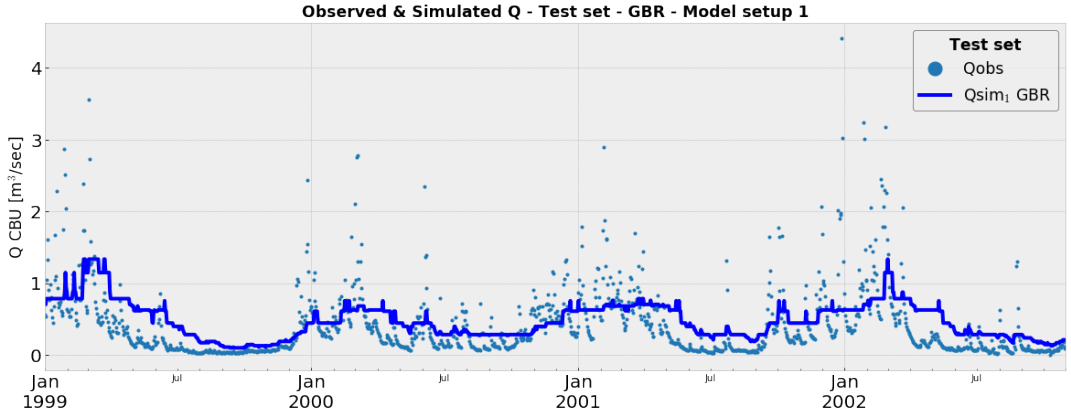


Figure C10. The time series of $Qsim$ and $Qobs$ for only the test set, for GBR - model setup 1.

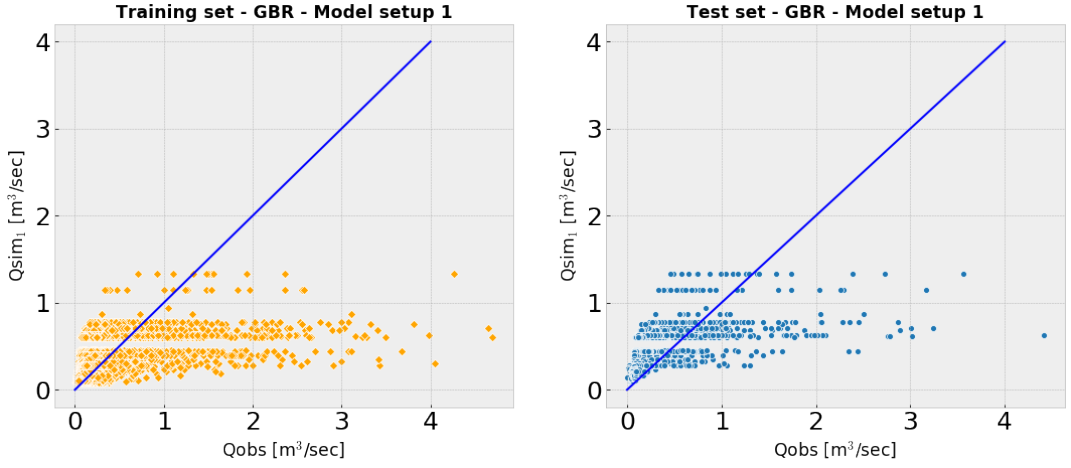


Figure C11. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for GBR - model setup 1. Underfitting is occuring.

C2.4 Results SVR - model setup 1

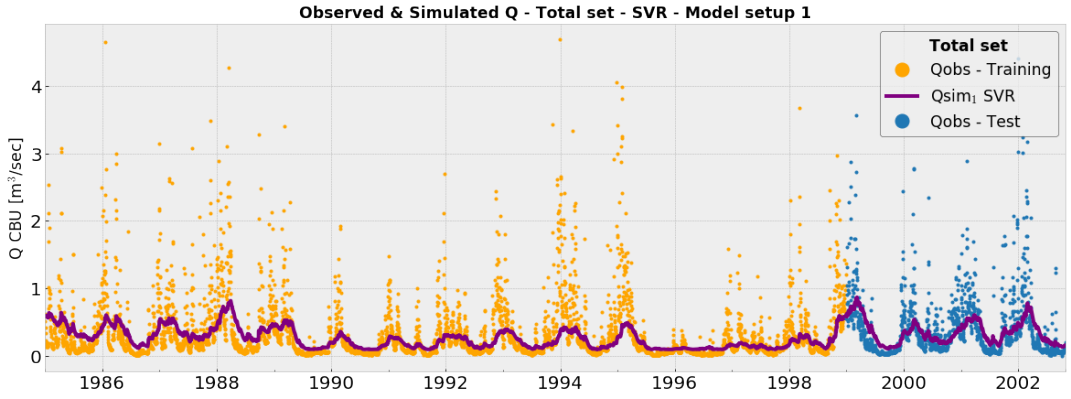


Figure C12. The time series of $Qsim$ and $Qobs$ for the training and test set, for SVR - model setup 1

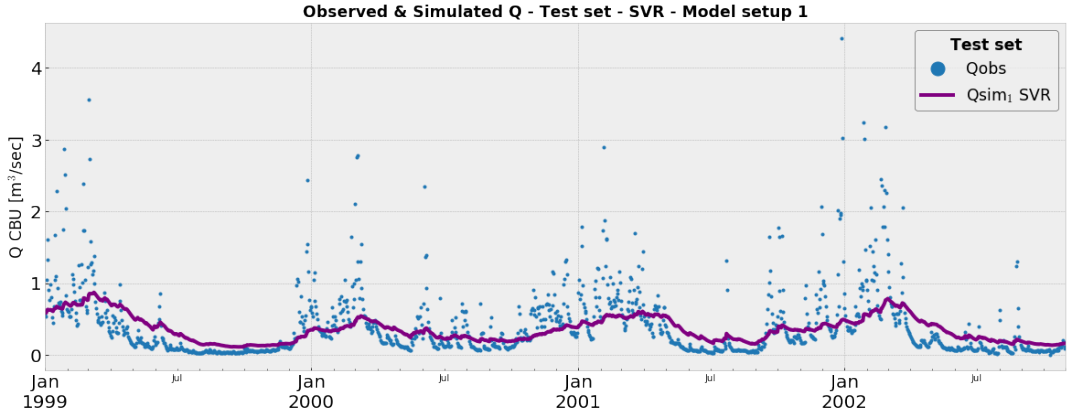


Figure C13. The time series of $Qsim$ and $Qobs$ for only the test set, for SVR - model setup 1

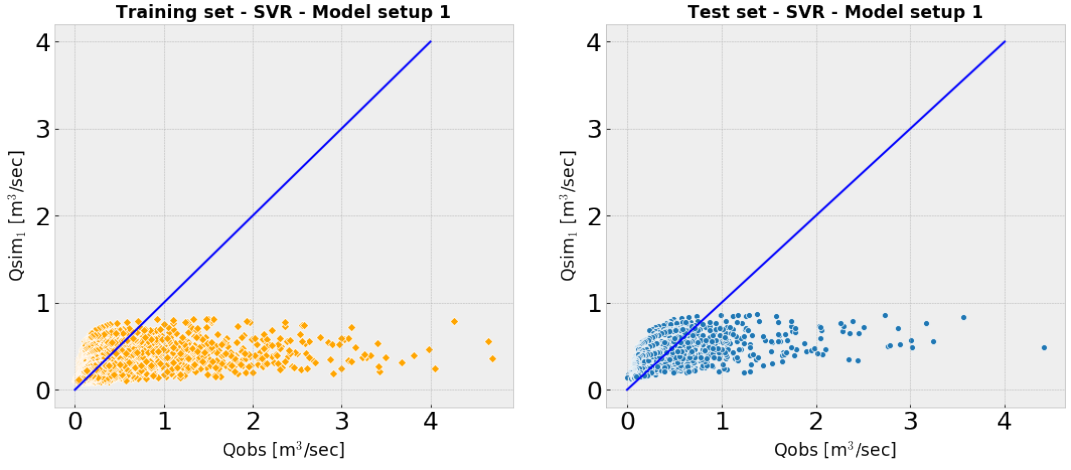


Figure C14. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for SVR - model setup 1. Underfitting is occuring

C3 Optimal hyperparameters - model setup 1

In this Appendix, the optimal hyperparameter set found with 5-folds grid search cross validation is depicted for each single machine learning algorithm in a Table. Moreover, the computation time for the hyperparameter tuning is given in the same table.

	5-folds grid search cross validation	
Hyperparameters DTR - model setup 1	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MSE
maximum tree depth	2, 4, 6, 8, 10	2
minimum samples in a leaf	1, 2, 4	1
minimum samples to obtain a split	2, 5, 10	2
Computation time	*30.1 sec*	

Table C1. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 1 DTR

	5-folds grid search cross validation	
Hyperparameters RFR - model setup 1	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MSE
maximum tree depth	2, 4, 6, 8, 10	2
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	10
Computation time	*44.4 min*	

Table C2. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 1 RFR

	5-folds grid search cross validation	
Hyperparameters GBR - model setup 1	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MSE
maximum tree depth	2, 4, 6, 8, 10	2
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	25
Computation time	*138.5 min*	

Table C3. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 1 GBR

	5-folds grid search cross validation	
Hyperparameters SVR - model setup 1	**Grid**	**Optimal Hyperparameter**
gamma (kernel coefficient)	0.001, 0.01, 0.1, 1	0.01
C (penalty error parameter)	0.001, 0.01, 0.1, 1, 10	10
Computation time	*23.9 sec*	

Table C4. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 1 SVR

C4 Regression trees RFR - model setup 1

In this Appendix, the 10 regression trees of the RFR of model setup 1 are visualised. In the end, the output for each sample of each regression tree is averaged to get the output $Qsim$ per sample of the model RFR.

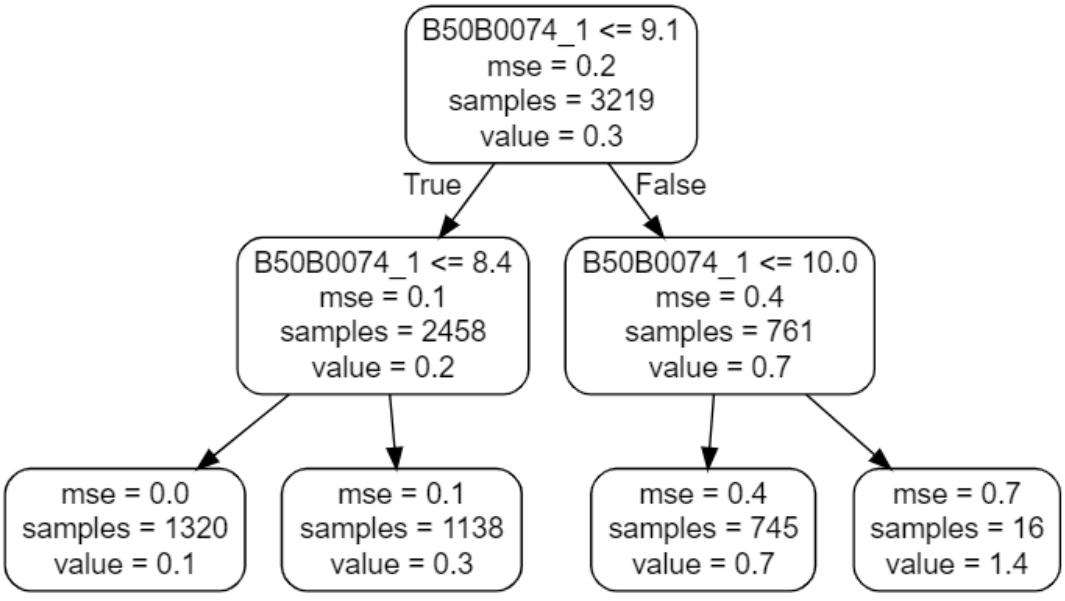


Figure C15. Regression tree 1 of RFR model setup 1

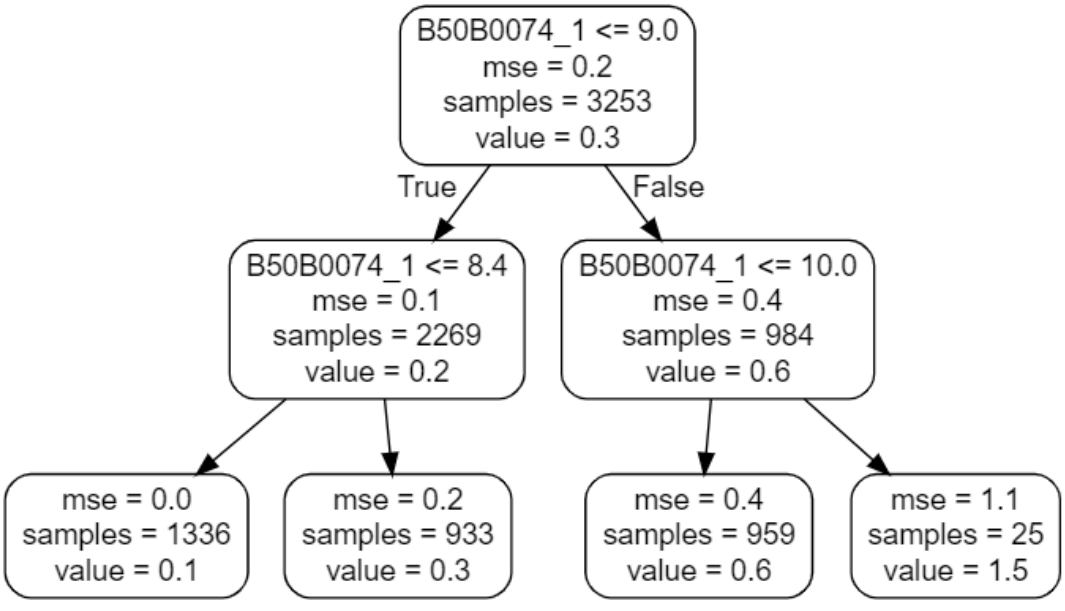


Figure C16. Regression tree 2 of RFR model setup 1

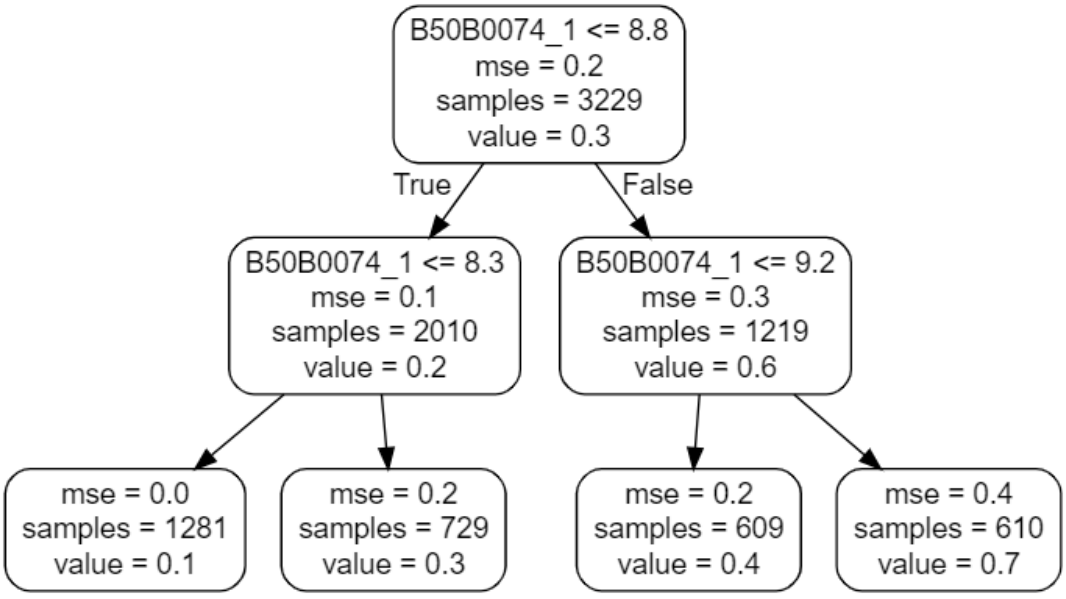


Figure C17. Regression tree 3 of RFR model setup 1

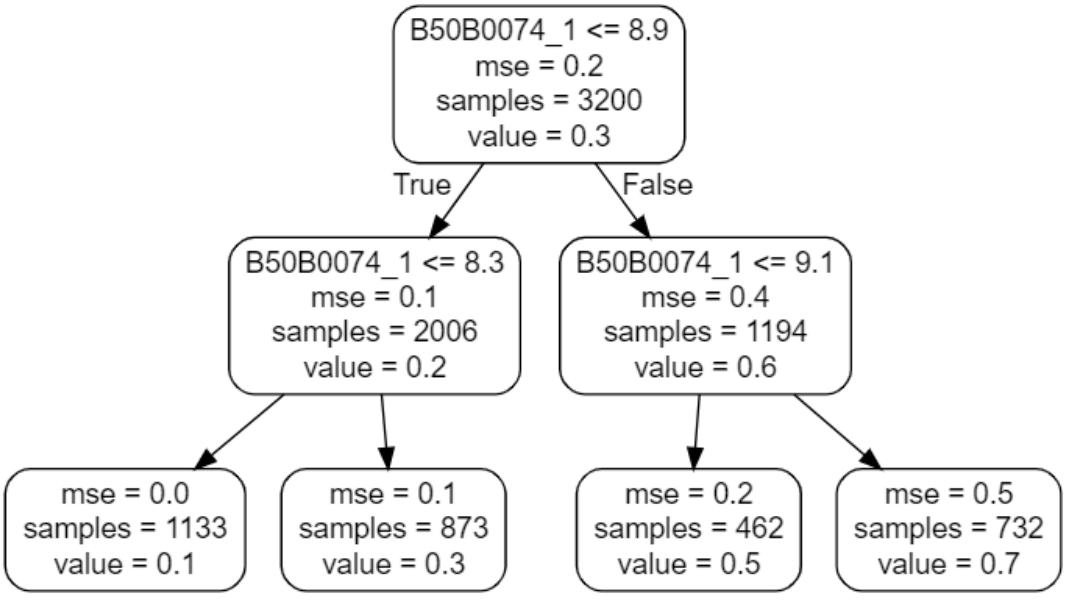


Figure C18. Regression tree 4 of RFR model setup 1

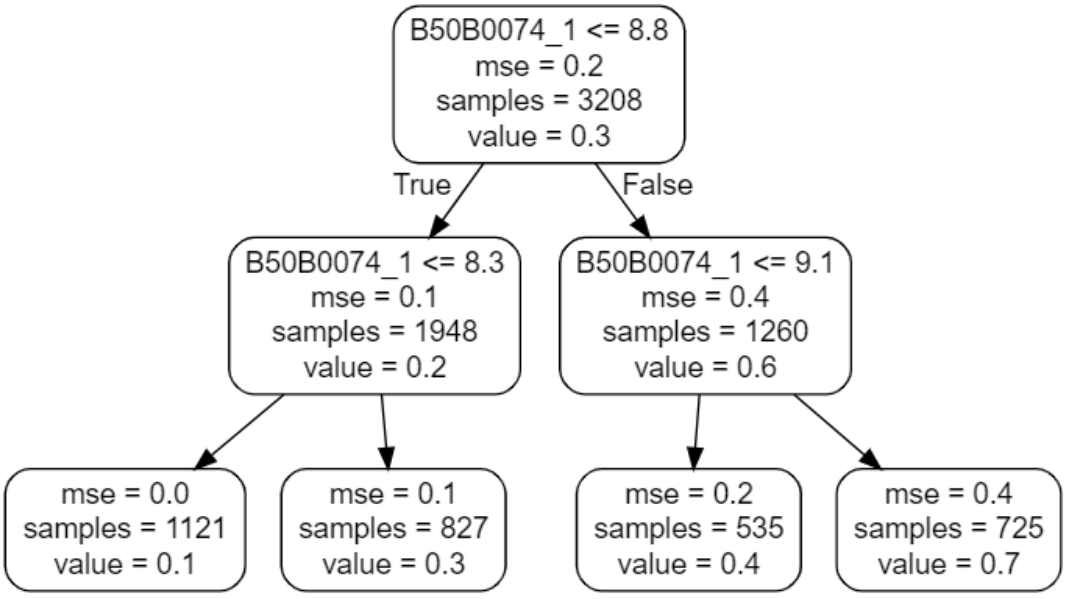


Figure C19. Regression tree 5 of RFR model setup 1

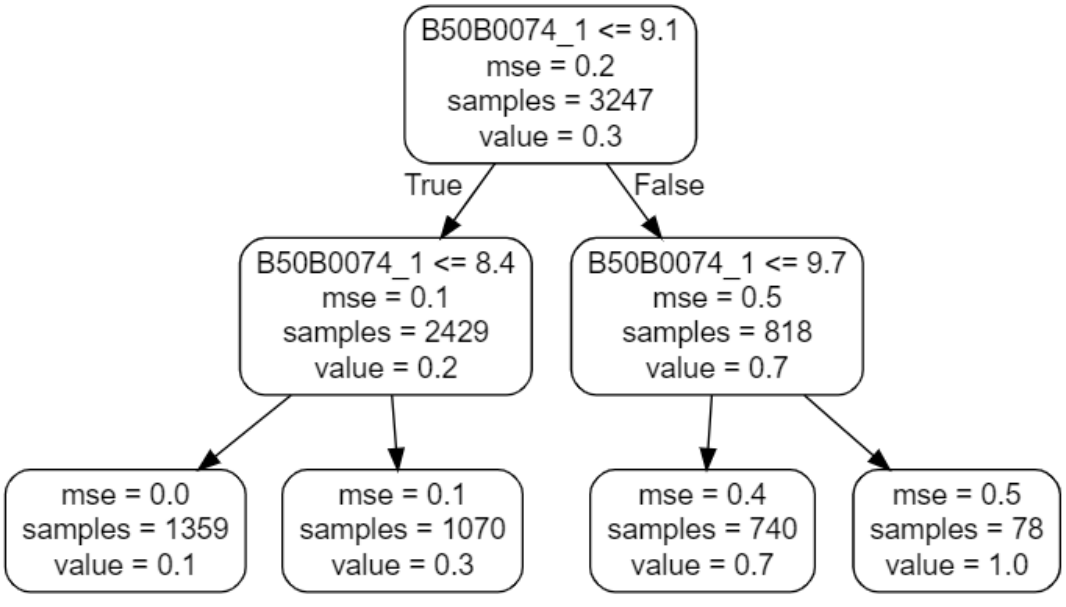


Figure C20. Regression tree 6 of RFR model setup 1

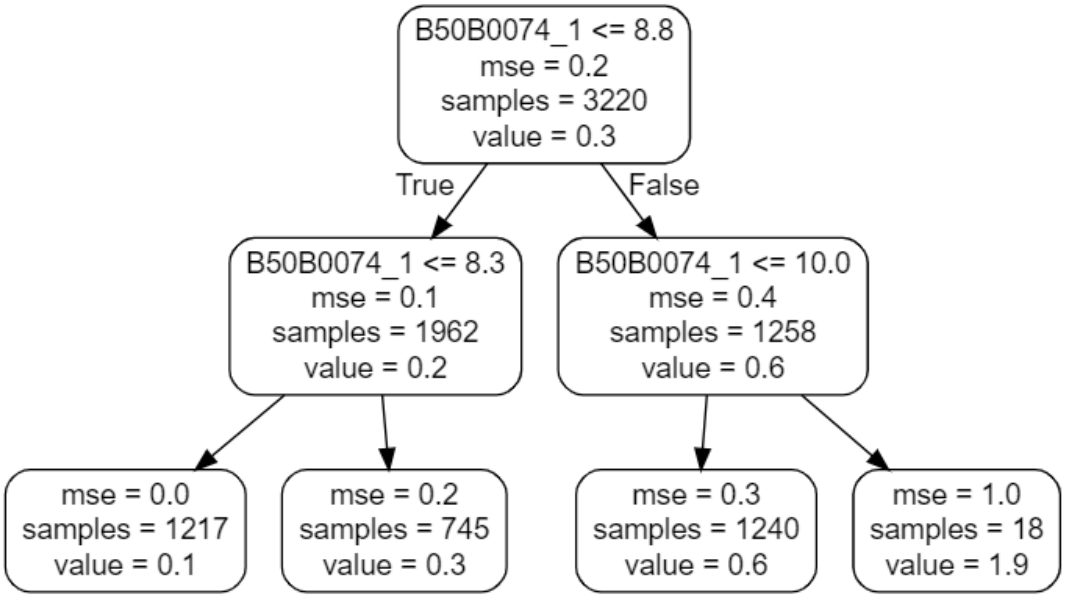


Figure C21. Regression tree 7 of RFR model setup 1

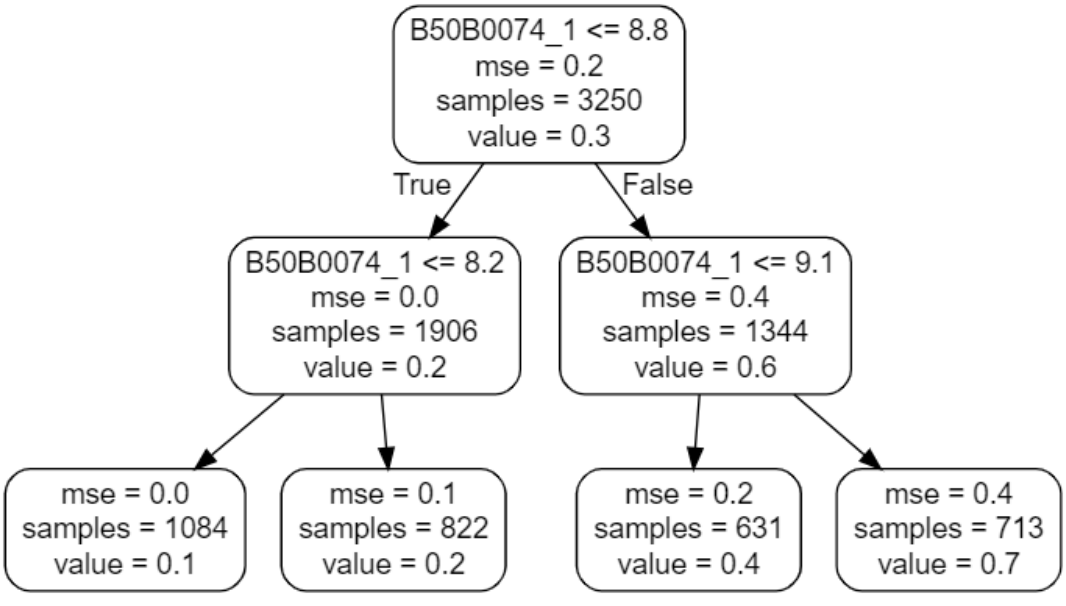


Figure C22. Regression tree 8 of RFR model setup 1

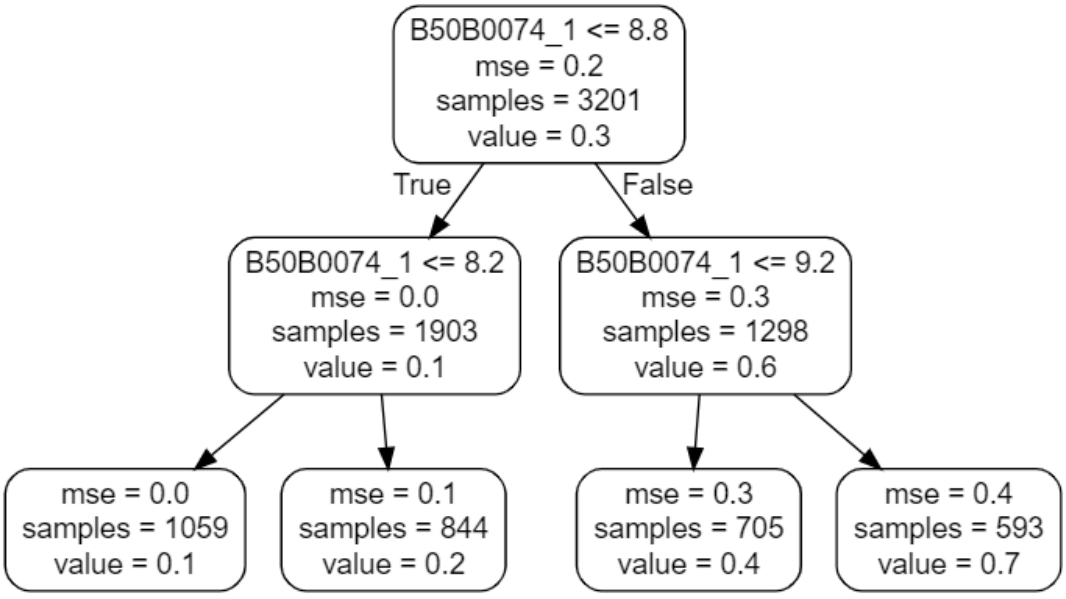


Figure C23. Regression tree 9 of RFR model setup 1

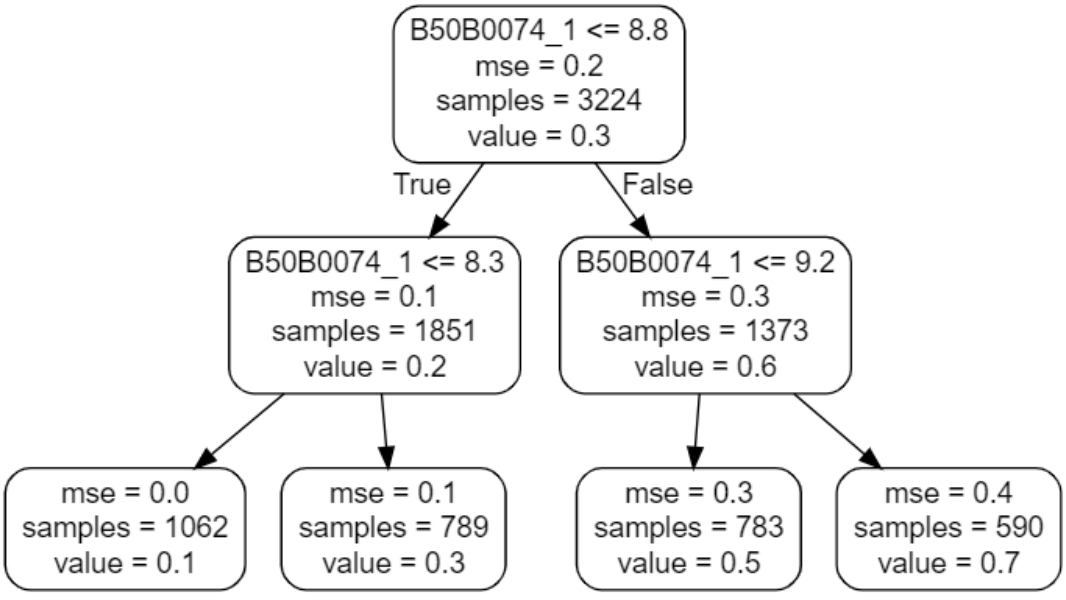


Figure C24. Regression tree 10 of RFR model setup 1

Appendix D: model setup 2

D1 Dataset for model setup 2

In this Appendix, the time series of the input variables $X1_0$-$X1_5$ and the target $Qobs$ for model setup 2 are visualised. A division is made between the training set and the test set. Also, a heatmap is depicted to visualise the correlations between the input variables and the target.

D1.1 Input variables for model setup 2

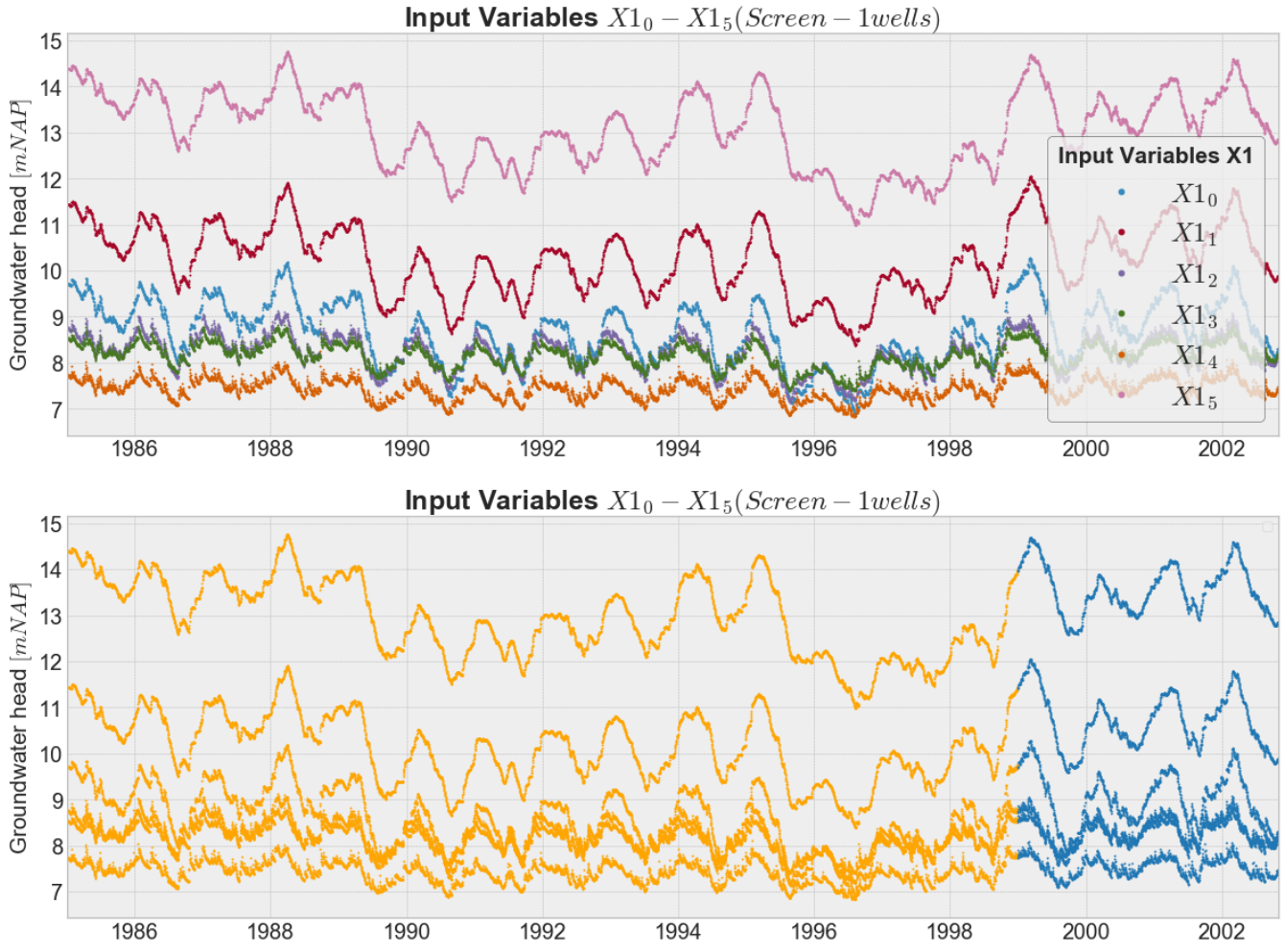


Figure D1. The time series of the input variables screen-1 wells $X1_0$, $X1_1$, $X1_2$, $X1_3$, $X1_4$ and $X1_5$ for model setup 2, divided into the training set (1985-1999) and the test set (1999-2003)

D1.2 Target for model setup 2

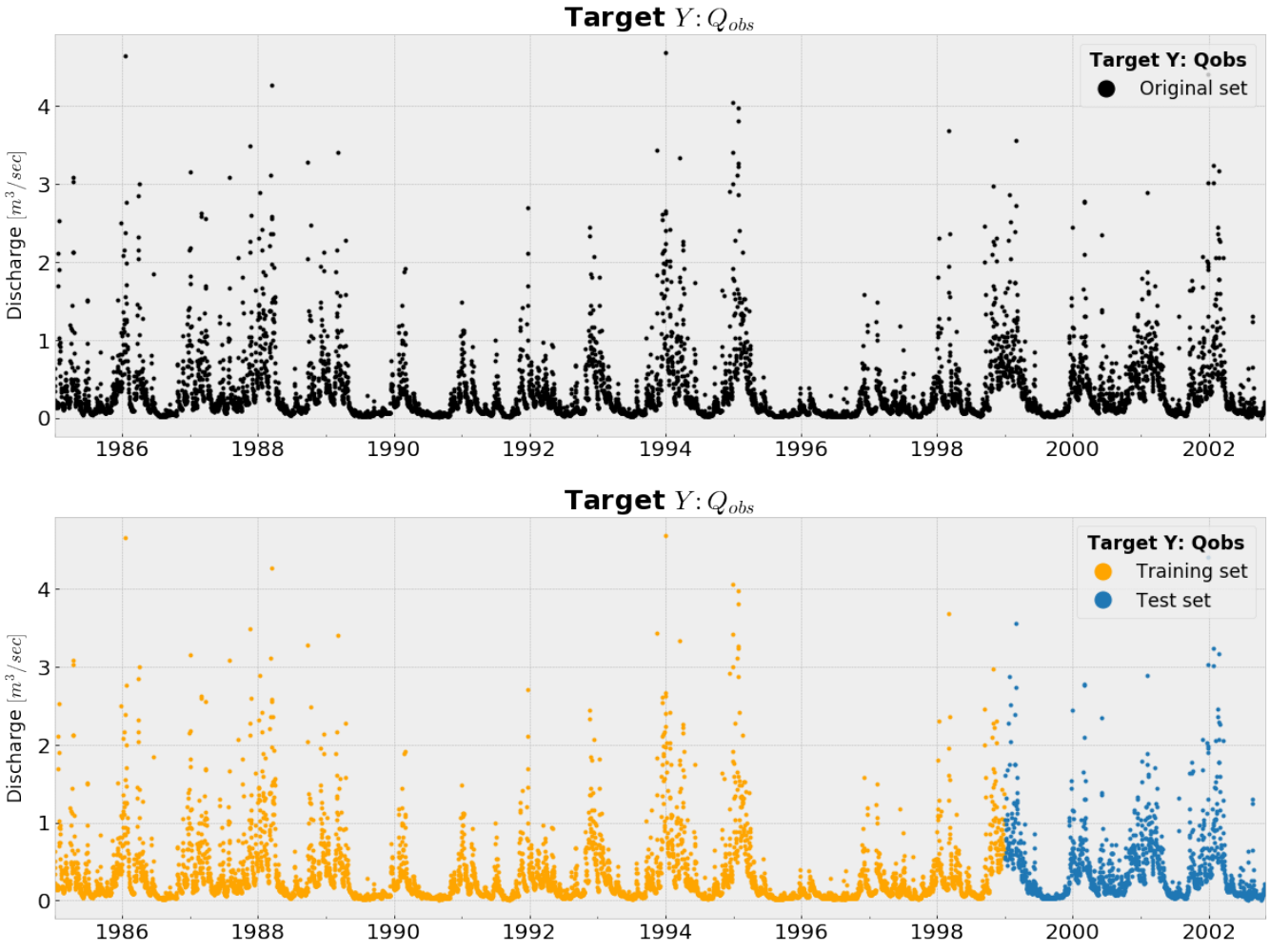


Figure D2. The time series of the target $Qobs$ for model setup 2, divided into the training set (1985-1999) and the test set (1999-2003)

D1.3 Correlation overview dataset for model setup 2

In this Appendix, a correlation heatmap is depicted of the dataset of model setup 2. This figure shows already to which input variable the target $Qobs$ is mostly correlated with.

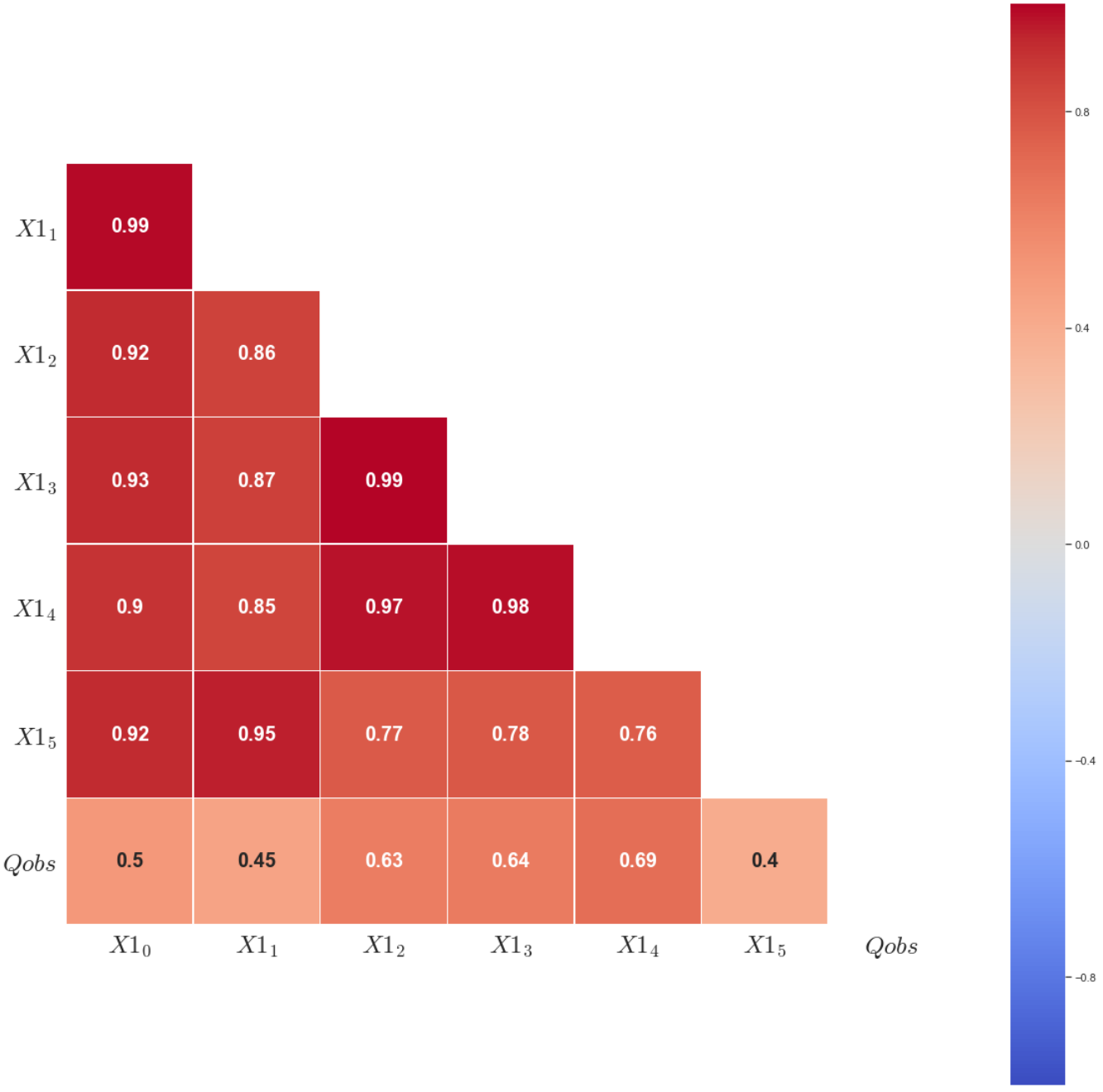


Figure D3. An overview of the correlations of the datatset for model setup 2, depicted in a heatmap

D2 Results - model setup 2

In this Appendix, the results of the different machine learning algorithms of model setup 2 are separately visualised. First, the $Qsim$ time series is plotted for the training and test, followed by a plot of zooming in on the test set. The last figure of each machine learning algorithm is a scatterplot of $Qobs$ against $Qsim$ to easlity detect over- or underfitting.

D2.1 Results DTR - model setup 2

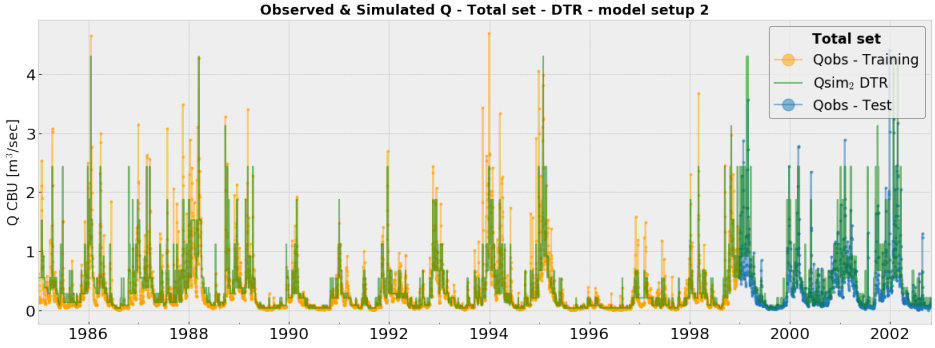


Figure D4. The time series of $Qsim$ and $Qobs$ for the training and test set, for DTR - model setup 2

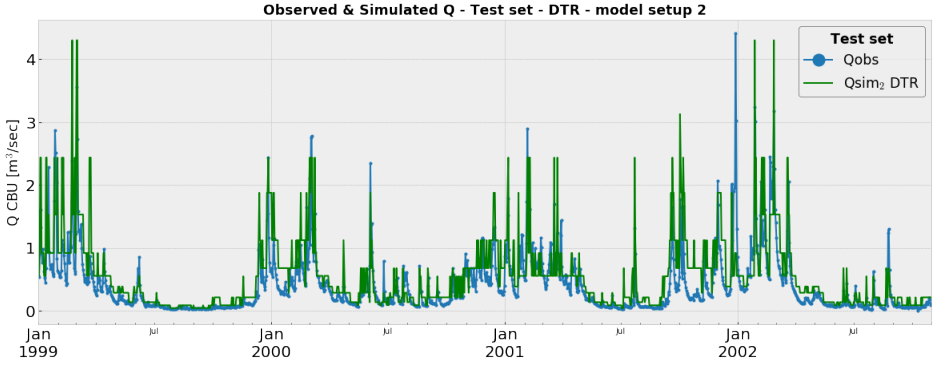


Figure D5. The time series of $Qsim$ and $Qobs$ for only the test set, for DTR - model setup 2

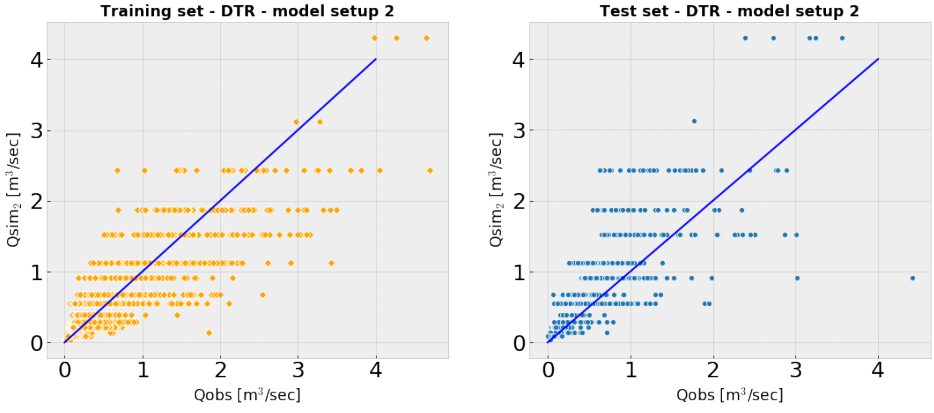


Figure D6. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for DTR - model setup 2. Underfitting is occuring when using the machine learning algorithm DTR for model setup 2.

D2.2 Results RFR - model setup 2

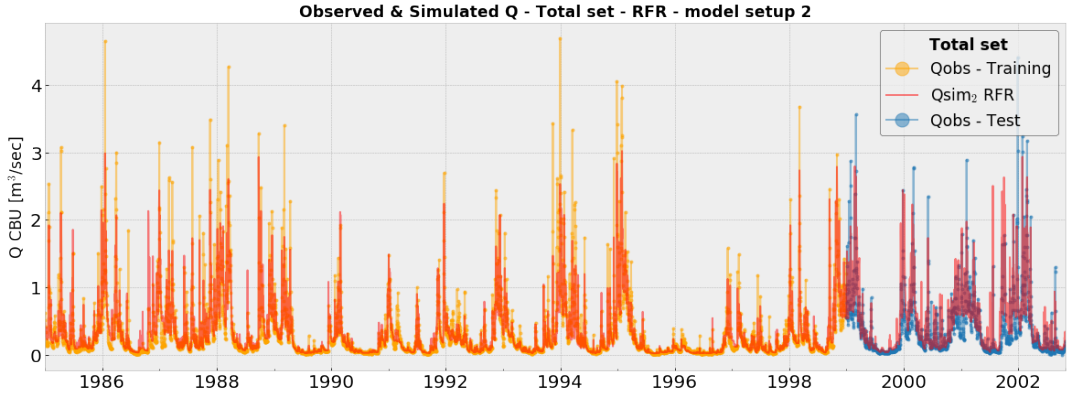


Figure D7. The time series of $Qsim$ and $Qobs$ for the training and test set, for RFR - model setup 2

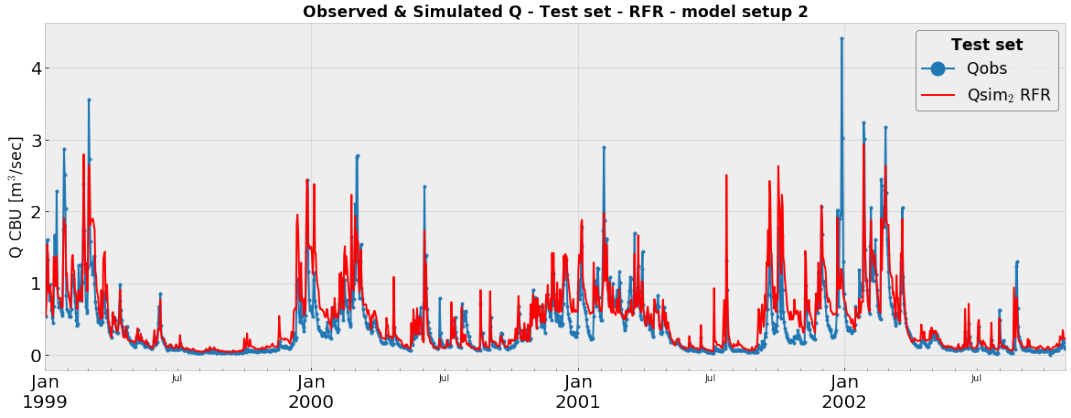


Figure D8. The time series of $Qsim$ and $Qobs$ for only the test set, for RFR - model setup 2

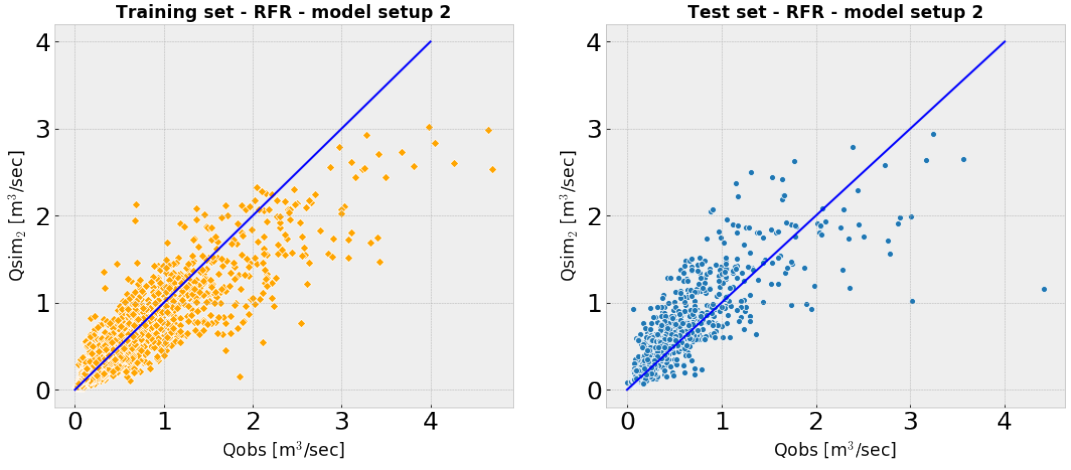


Figure D9. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for RFR - model setup 2. Slightly overfitting

D2.3 Results GBR - model setup 2

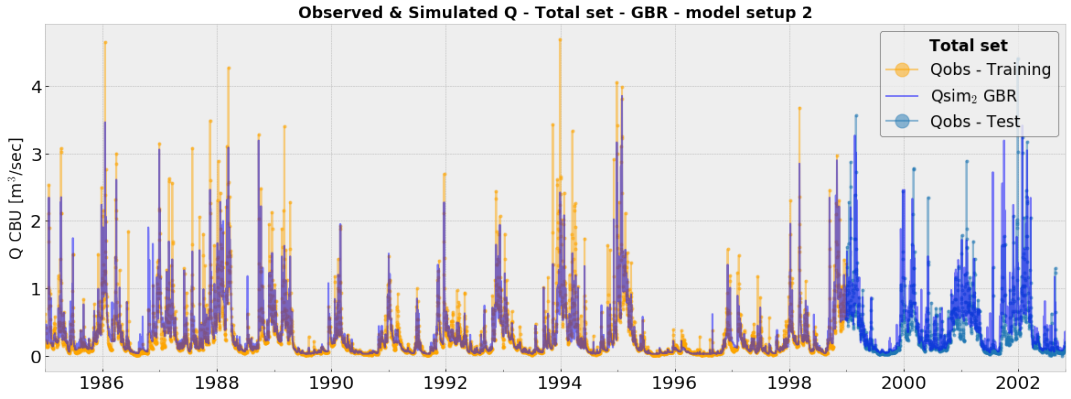


Figure D10. The time series of $Qsim$ and $Qobs$ for the training and test set, for GBR - model setup 2

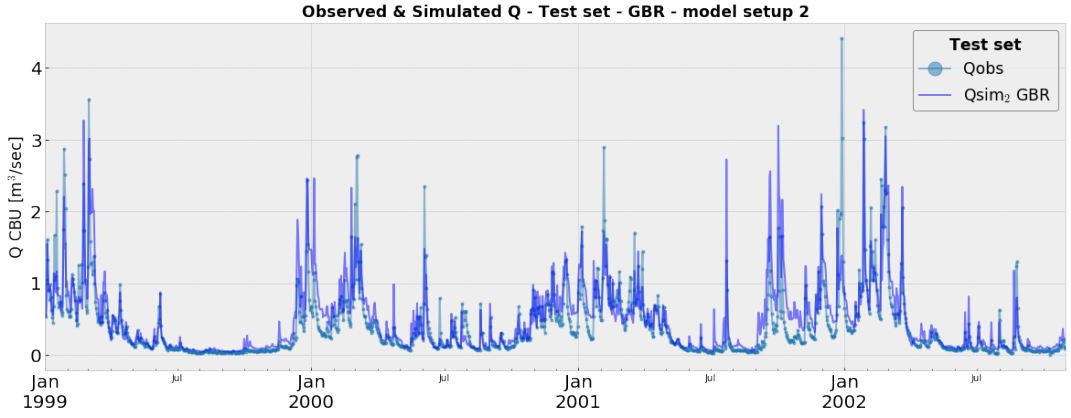


Figure D11. The time series of $Qsim$ and $Qobs$ for only the test set, for GBR - model setup 2

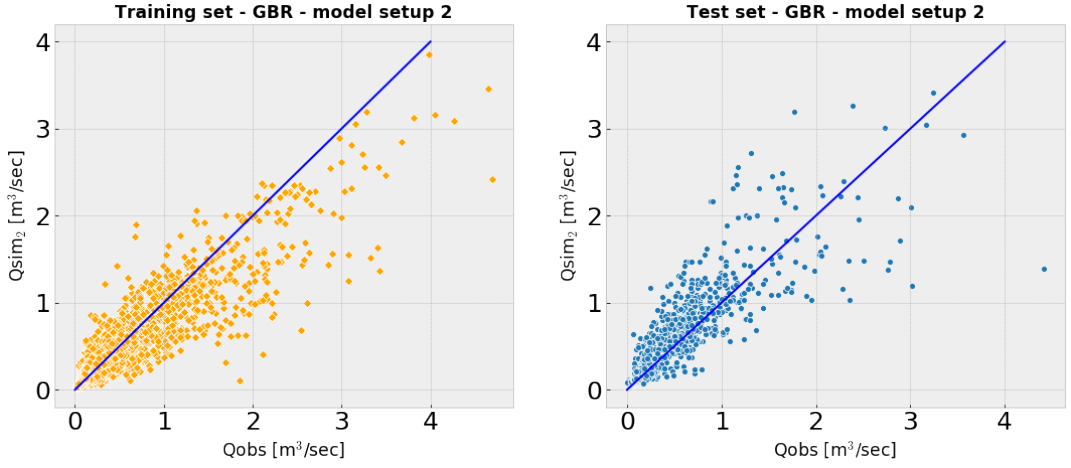


Figure D12. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for GBR - model setup 2. Slightly overfitting.

D2.4 Results SVR - model setup 2

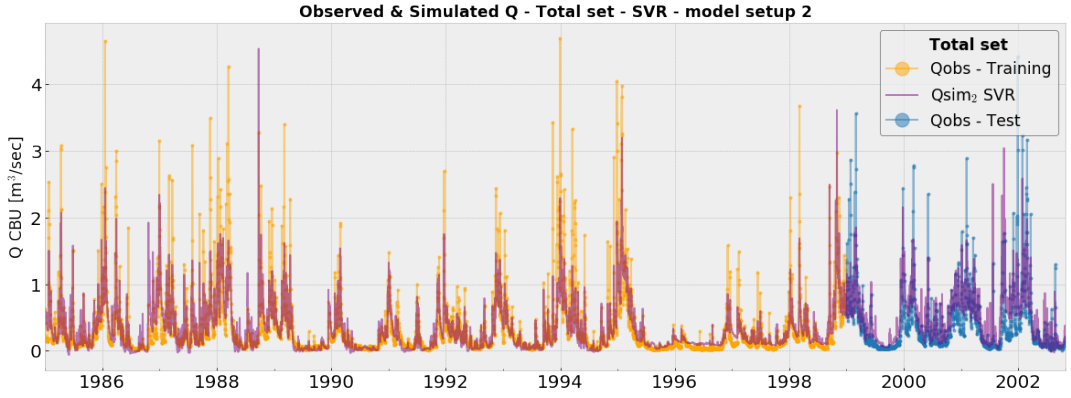


Figure D13. The time series of $Qsim$ and $Qobs$ for the training and test set, for SVR - model setup 2

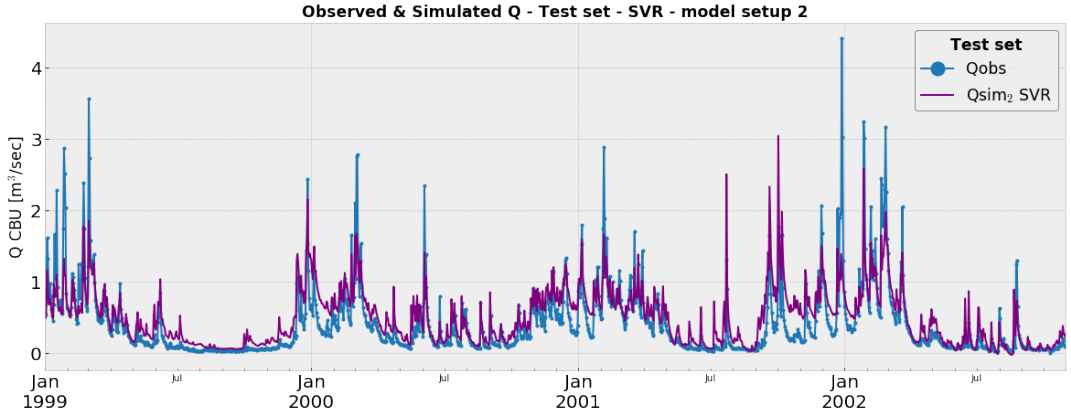


Figure D14. The time series of $Qsim$ and $Qobs$ for only the test set, for SVR - model setup 2

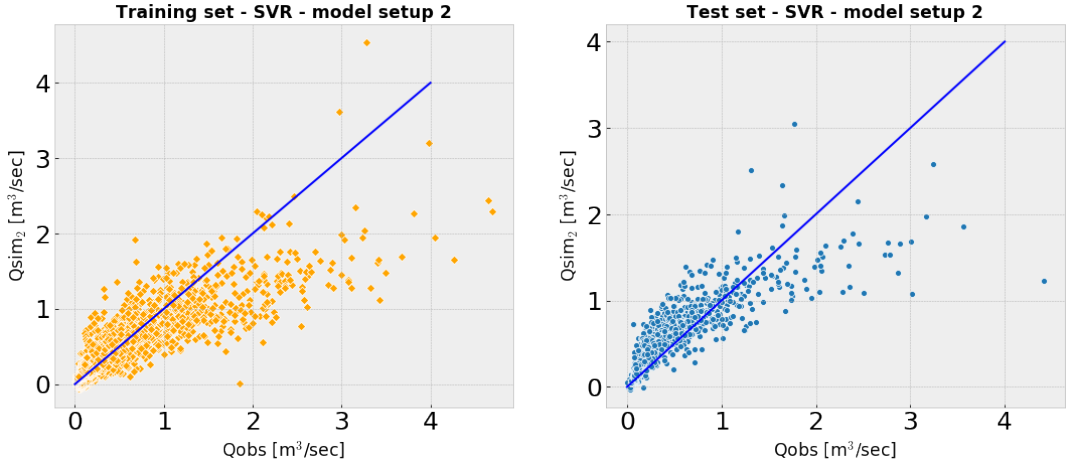


Figure D15. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for SVR - model setup 2. No under- or overfitting.

D3 Optimal hyperparameters - model setup 2

In this Appendix, the optimal hyperparameter set found with 5-folds grid search cross validation is depicted for each single machine learning algorithm in a Table. Moreover, the computation time for the hyperparameter tuning is given in the same table.

5-folds grid search cross validation		
Hyperparameters DTR - model setup 2	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MSE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	1
minimum samples to obtain a split	2, 5, 10	5
Computation time	*29.5 sec*	

Table D1. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 2 DTR

5-folds grid search cross validation		
Hyperparameters RFR - model setup 2	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	6
minimum samples in a leaf	1, 2, 4	2
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	250
Computation time	*169.9 min*	

Table D2. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 2 RFR

5-folds grid search cross validation		
Hyperparameters GBR - model setup 2	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	50
Computation time	*471.9 min*	

Table D3. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 2 GBR

5-folds grid search cross validation		
Hyperparameters SVR - model setup 2	**Grid**	**Optimal Hyperparameter**
gamma (kernel coefficient)	0.001, 0.01, 0.1, 1	0.1
C (penalty error parameter)	0.001, 0.01, 0.1, 1, 10	10
Computation time	*16.5 sec*	

Table D4. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 2 SVR

D4 Regression tree DTR - model setup 2

In this Appendix, the regression tree of the DTR of model setup 2 is visualised.

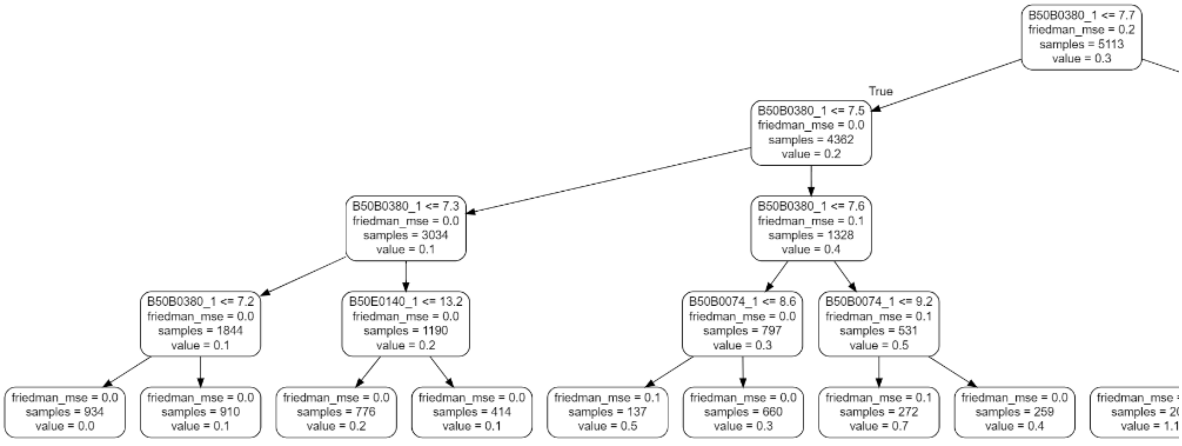


Figure D16. Left part of the regression tree of DTR model setup 2

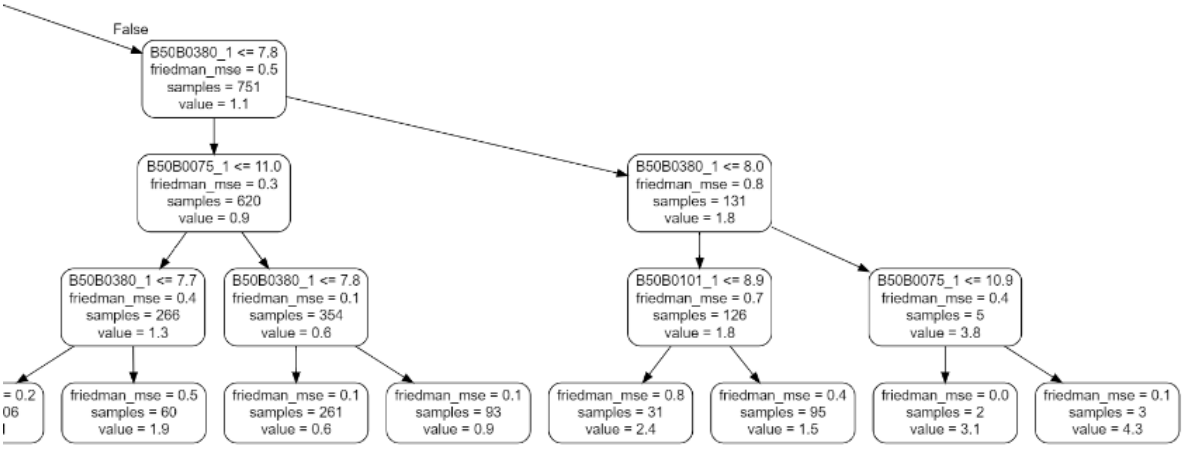


Figure D17. Right part of the regression tree of DTR model setup 2

Appendix E: model setup 3

E1 Dataset for model setup 3

In this Appendix, the time series of the input variables $X1_0$-$X1_5$, $X2_0$-$X2_5$ and the target $Qobs$ for model setup 3 are visualised. A division is made between the training set and the test set.

E1.1 Input variables for model setup 3

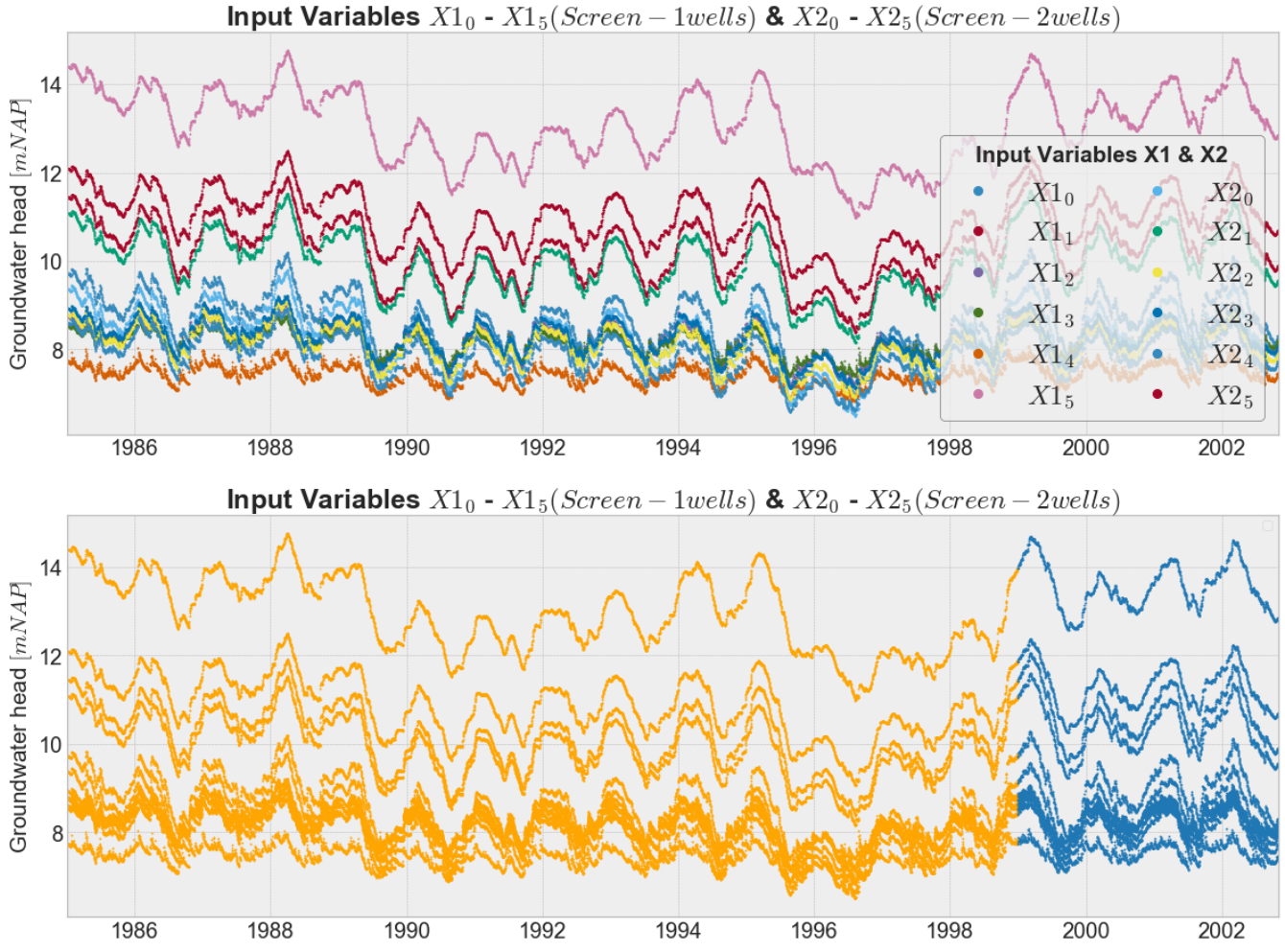


Figure E1. The time series of the input variables screen-1 & screen-2 wells $X1_0$, $X1_1$, $X1_2$, $X1_3$, $X1_4$, $X1_5$, $X2_0$, $X2_1$, $X2_2$, $X2_3$, $X2_4$ and $X2_5$ for model setup 3, divided into the training set (1985-1999) and the test set (1999-2003)

E1.2 Target for model setup 3

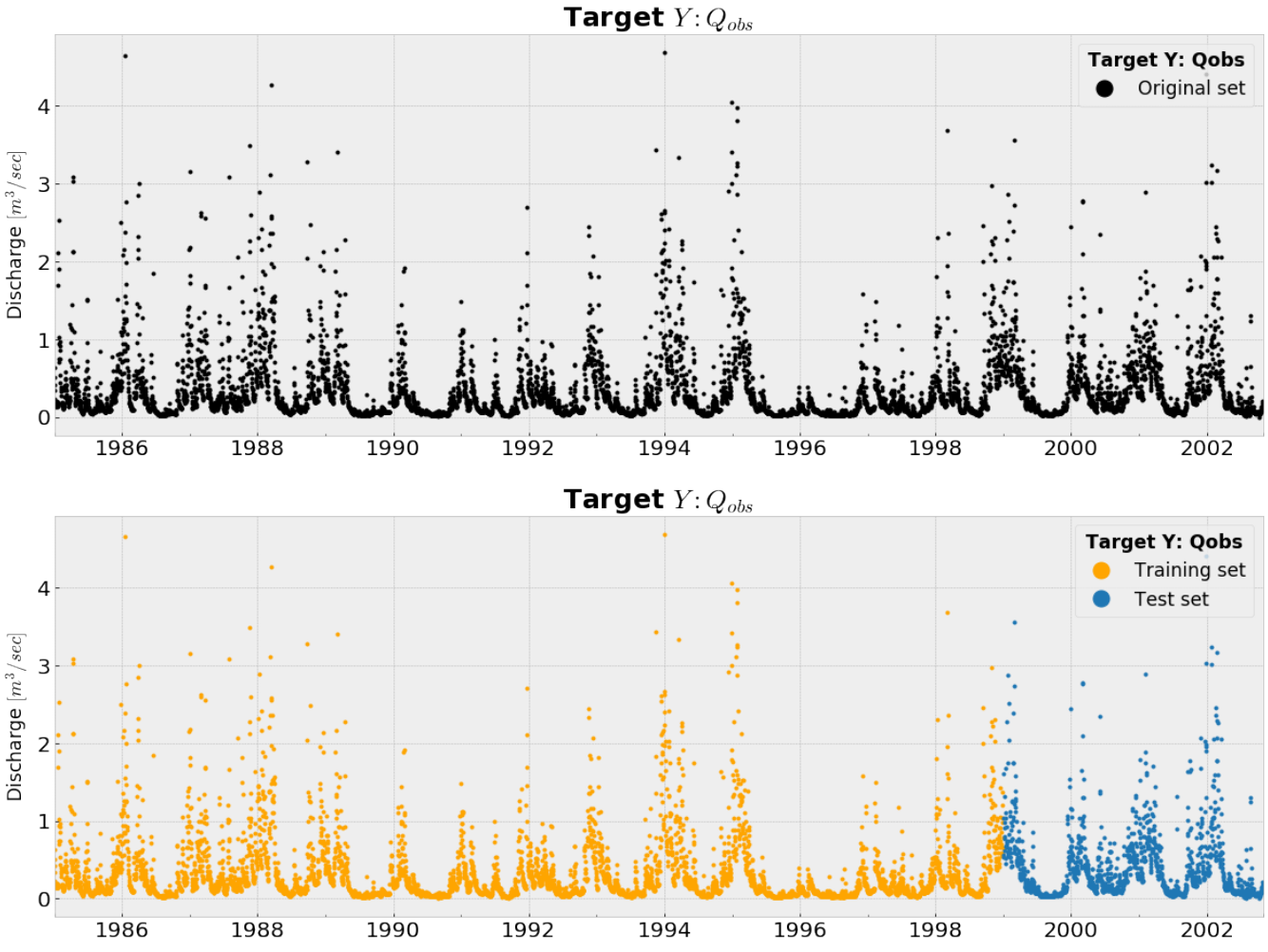


Figure E2. The time series of the target *Qobs* for model setup 3, divided into the training set (1985-1999) and the test set (1999-2003)

E1.3 Correlation overview dataset for model setup 3

In this Appendix, a correlation heatmap is depicted of the dataset of model setup 3. This figure shows already to which input variable the target $Qobs$ is mostly correlated with.

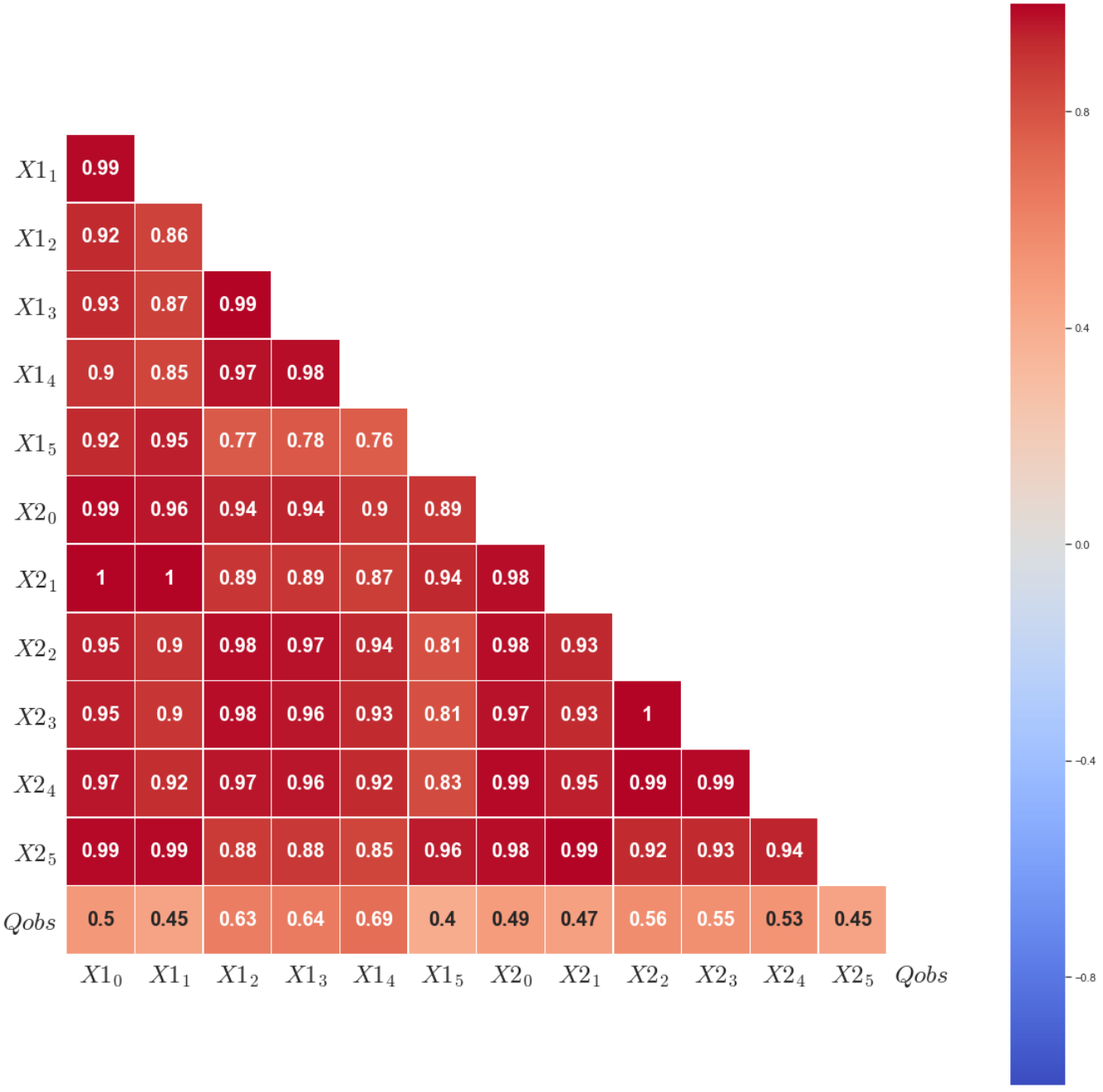


Figure E3. An overview of the correlations of the datatset for model setup 3, depicted in a heatmap

E2 Results - model setup 3

In this Appendix, the results of the different machine learning algorithms of model setup 3 are separately visualised. First, the $Qsim$ time series is plotted for the training and test, followed by a plot of zooming in on the test set. The last figure of each machine learning algorithm is a scatterplot of $Qobs$ against $Qsim$ to easlity detect over- or underfitting.

E2.1 Results DTR - model setup 3

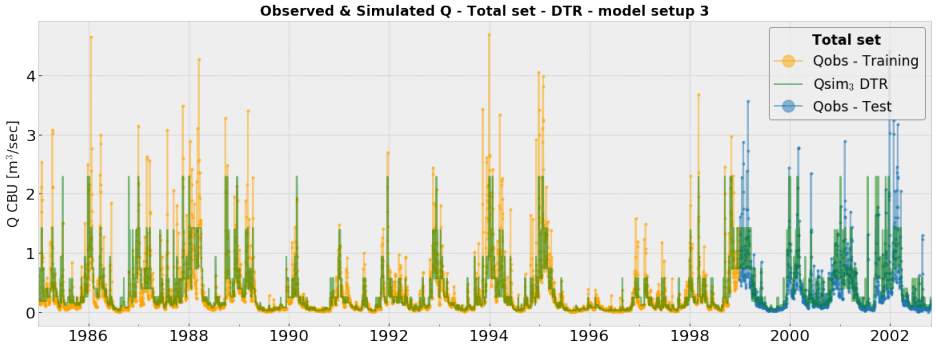


Figure E4. The time series of $Qsim$ and $Qobs$ for the training and test set, for DTR - model setup 3

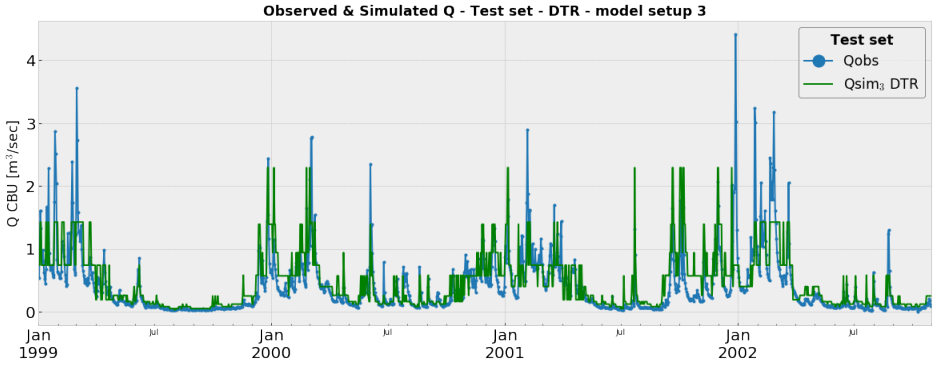


Figure E5. The time series of $Qsim$ and $Qobs$ for only the test set, for DTR - model setup 3

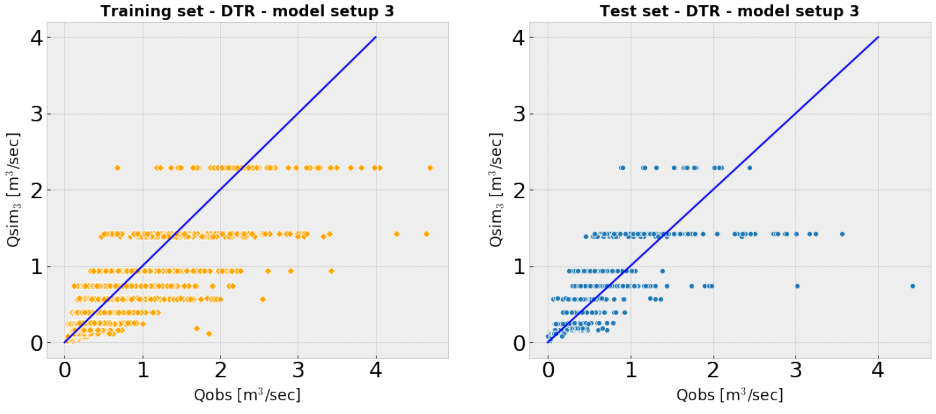


Figure E6. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for DTR - model setup 3

E2.2 Results RFR - model setup 3

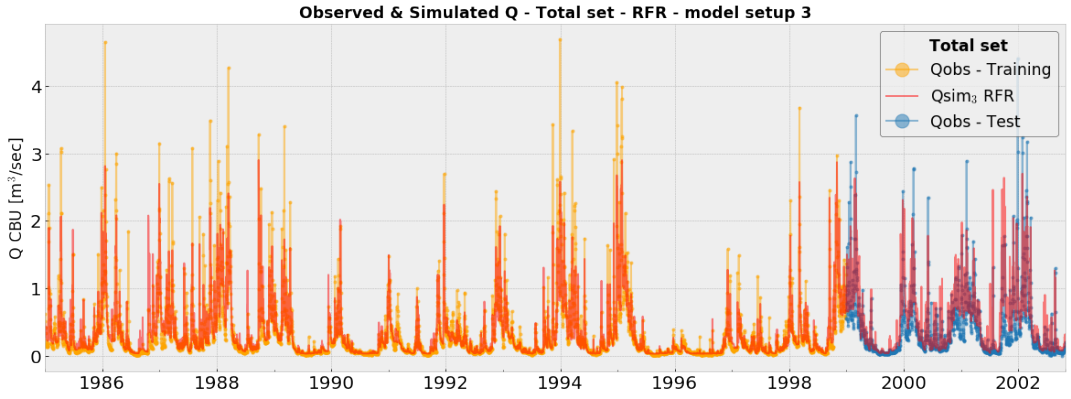


Figure E7. The time series of $Qsim$ and $Qobs$ for the training and test set, for RFR - model setup 3

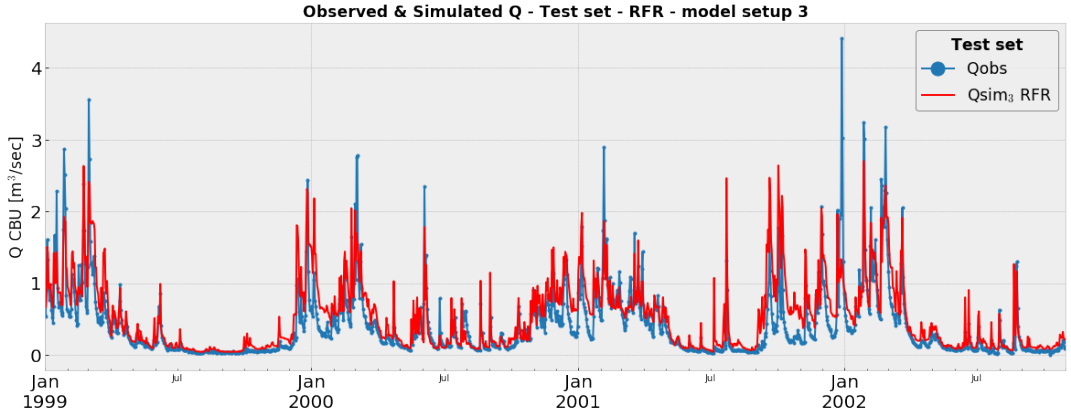


Figure E8. The time series of $Qsim$ and $Qobs$ for only the test set, for RFR - model setup 3

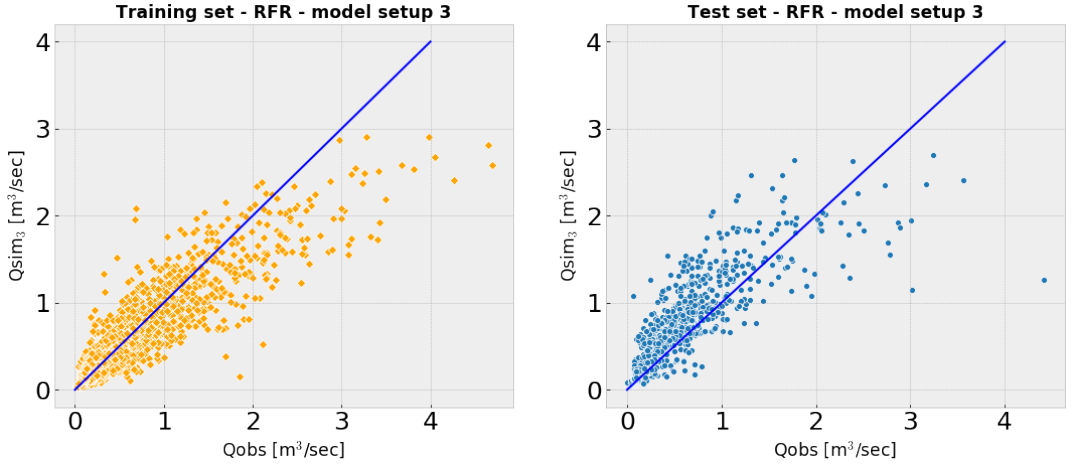


Figure E9. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for RFR - model setup 3

E2.3 Results GBR - model setup 3

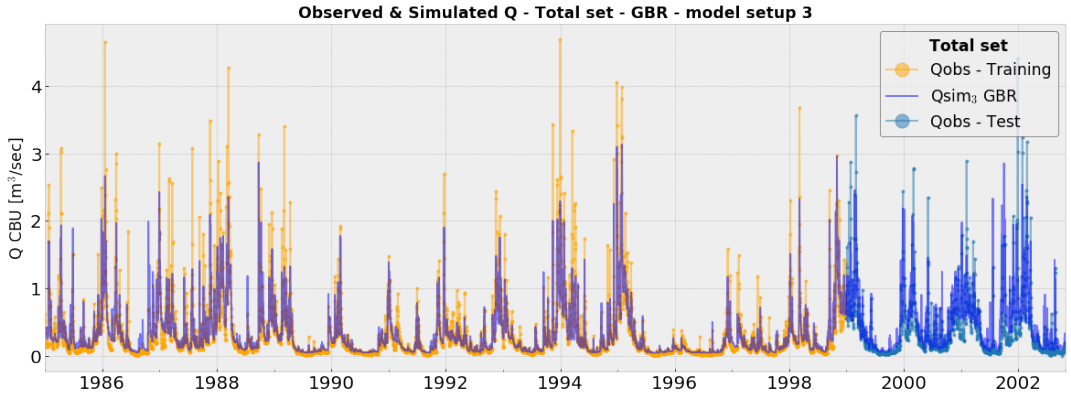


Figure E10. The time series of $Qsim$ and $Qobs$ for the training and test set, for GBR - model setup 3

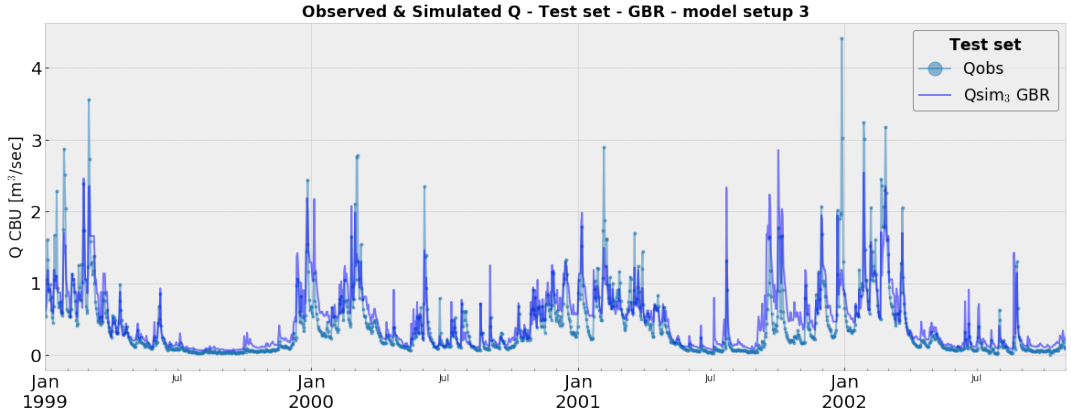


Figure E11. The time series of $Qsim$ and $Qobs$ for only the test set, for GBR - model setup 3

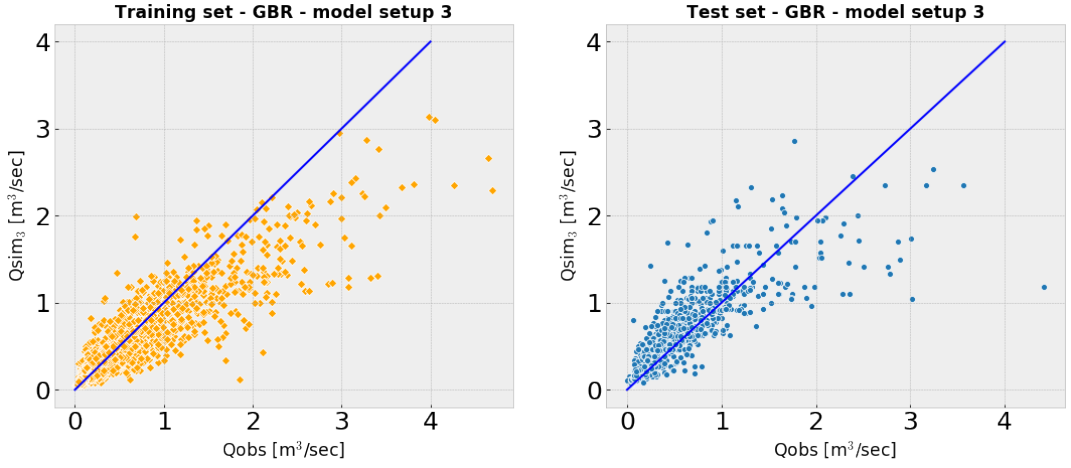


Figure E12. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for GBR - model setup 3

E2.4 Results SVR - model setup 3

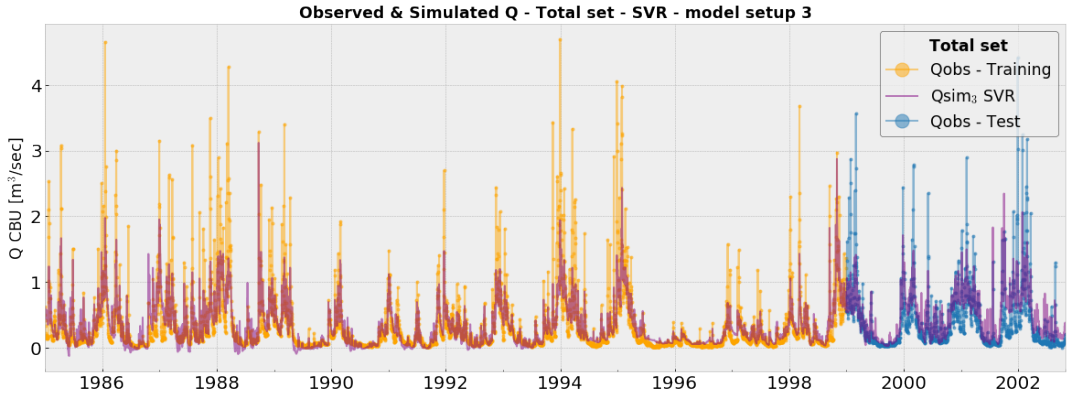


Figure E13. The time series of $Qsim$ and $Qobs$ for the training and test set, for SVR - model setup 3

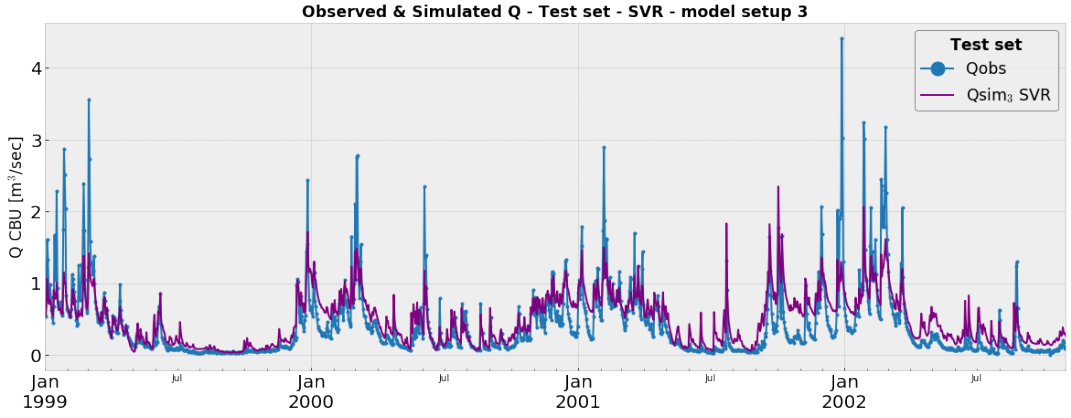


Figure E14. The time series of $Qsim$ and $Qobs$ for only the test set, for SVR - model setup 3

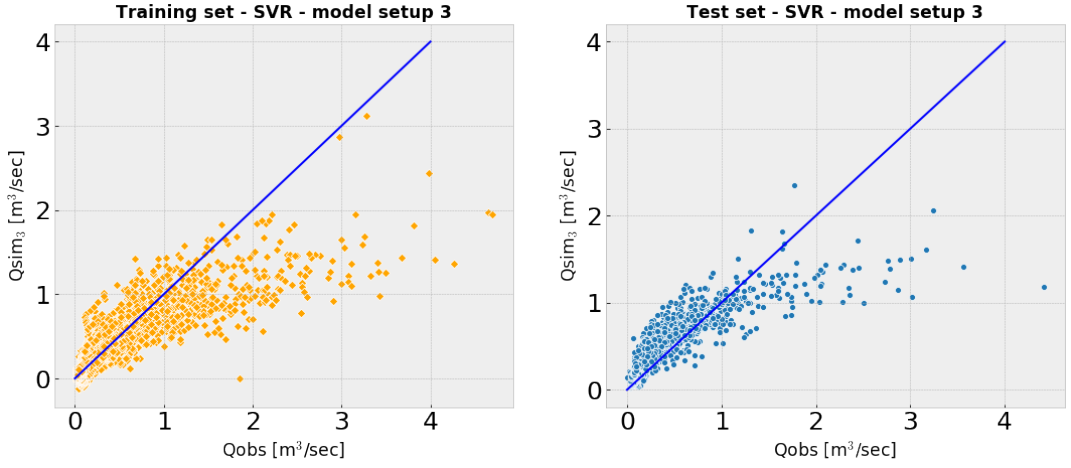


Figure E15. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for SVR - model setup 3

E3 Optimal hyperparameters - model setup 3

In this Appendix, the optimal hyperparameter set found with 5-folds grid search cross validation is depicted for each single machine learning algorithm in a Table. Moreover, the computation time for the hyperparameter tuning is given in the same table.

5-folds grid search cross validation		
Hyperparameters DTR - model setup 3	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
Computation time	*1.9 min*	

Table E1. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 3 DTR

5-folds grid search cross validation		
Hyperparameters RFR - model setup 3	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	6
minimum samples in a leaf	1, 2, 4	2
minimum samples to obtain a split	2, 5, 10	5
number of regression trees	10, 25, 50, 100, 250	250
Computation time	*270.4 min*	

Table E2. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 3 RFR

5-folds grid search cross validation		
Hyperparameters GBR - model setup 3	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	25
Computation time	*642.7 min*	

Table E3. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 3 GBR

5-folds grid search cross validation		
Hyperparameters SVR - model setup 3	**Grid**	**Optimal Hyperparameter**
gamma (kernel coefficient)	0.001, 0.01, 0.1, 1	0.1
C (penalty error parameter)	0.001, 0.01, 0.1, 1, 10	1
Computation time	*11.7 sec*	

Table E4. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 3 SVR

E4 Decision trees DTR - model setup 3

In this Appendix, the regression tree of the DTR of model setup 3 is visualised.

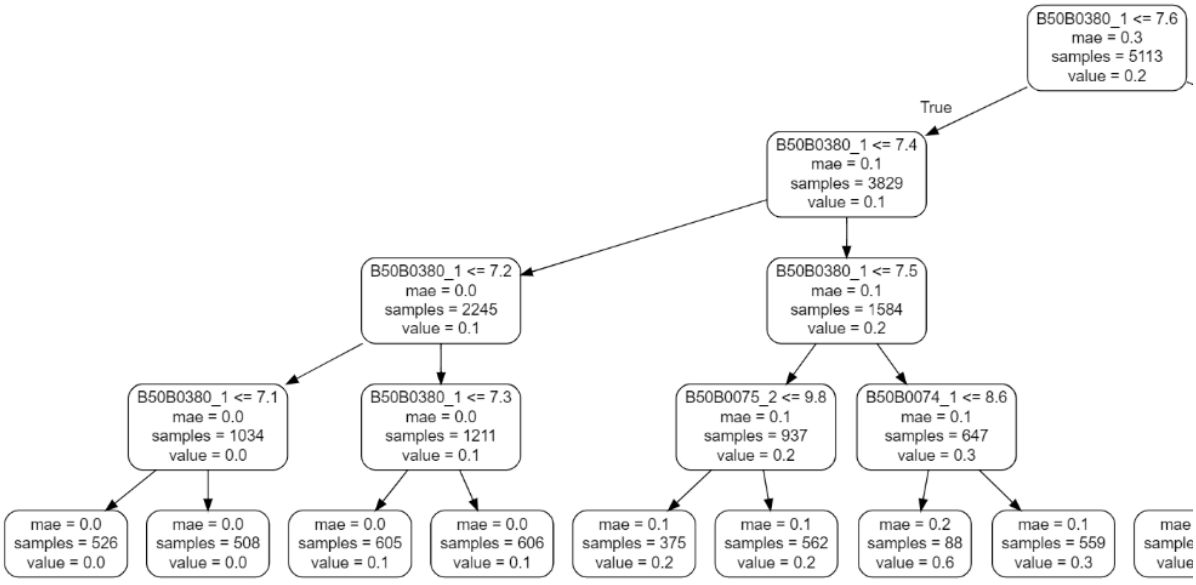


Figure E16. Left part of the regression tree of DTR model setup 3

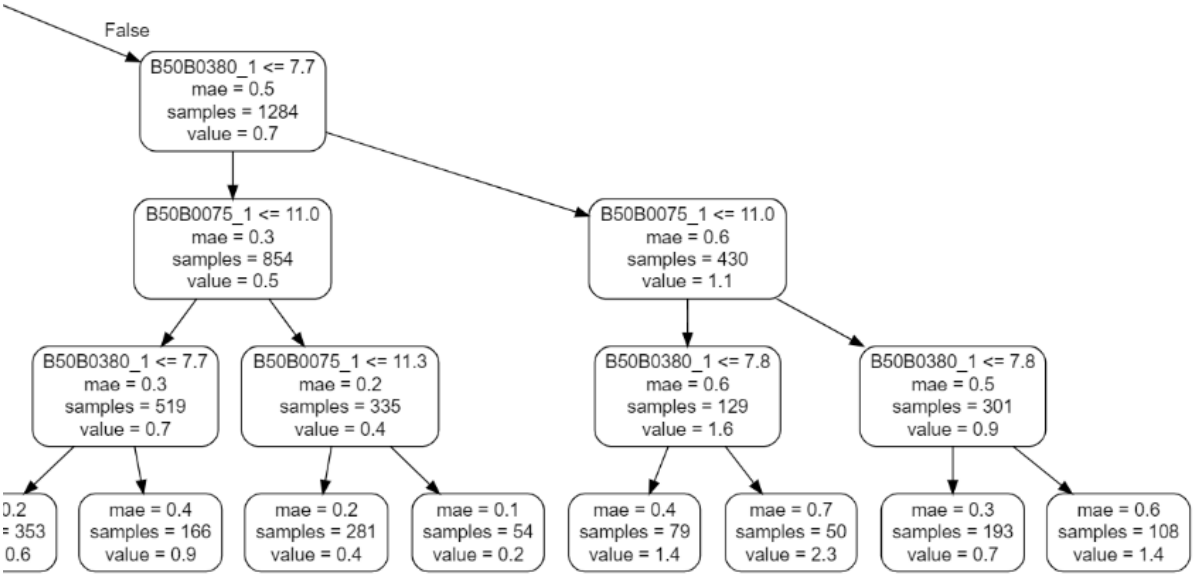


Figure E17. Right part of the regression tree of DTR model setup 3

Appendix F: model setup 4

F1 Dataset for model setup 4

In this Appendix, the time series of the input variables $X1_0$-$X1_5$, P, Ep and the target $Qobs$ for model setup 4 are visualised. A division is made between the training set and the test set.

F1.1 Input variables for model setup 4

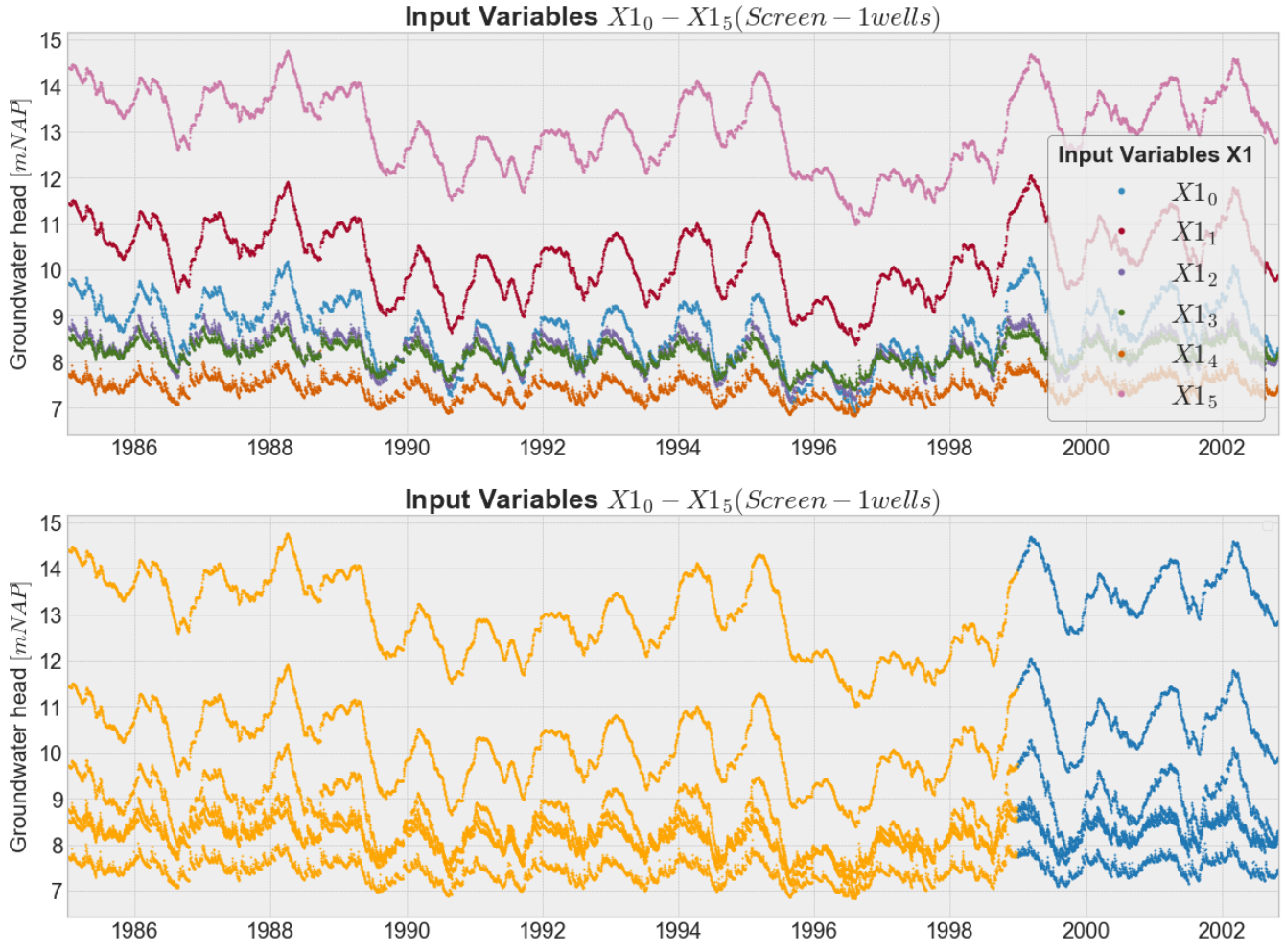


Figure F1. The time series of the input variables screen-1 wells $X1_0$, $X1_1$, $X1_2$, $X1_3$, $X1_4$ and $X1_5$ for model setup 4, divided into the training set (1985-1999) and the test set (1999-2003)

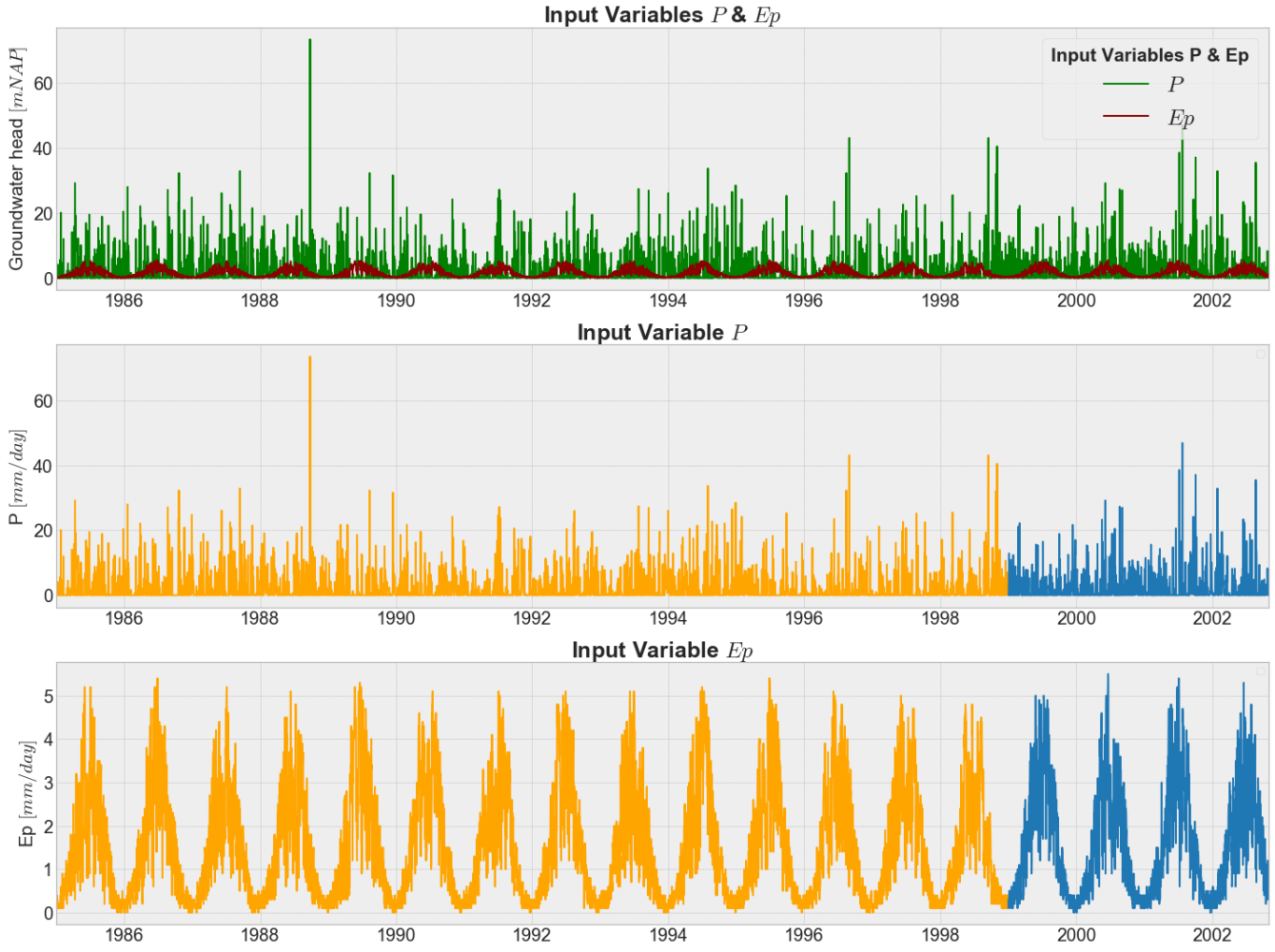


Figure F2. The time series of the input variables P & Ep for model setup 4, divided into the training set (1985-1999) and the test set (1999-2003)

F1.2 Target for model setup 4

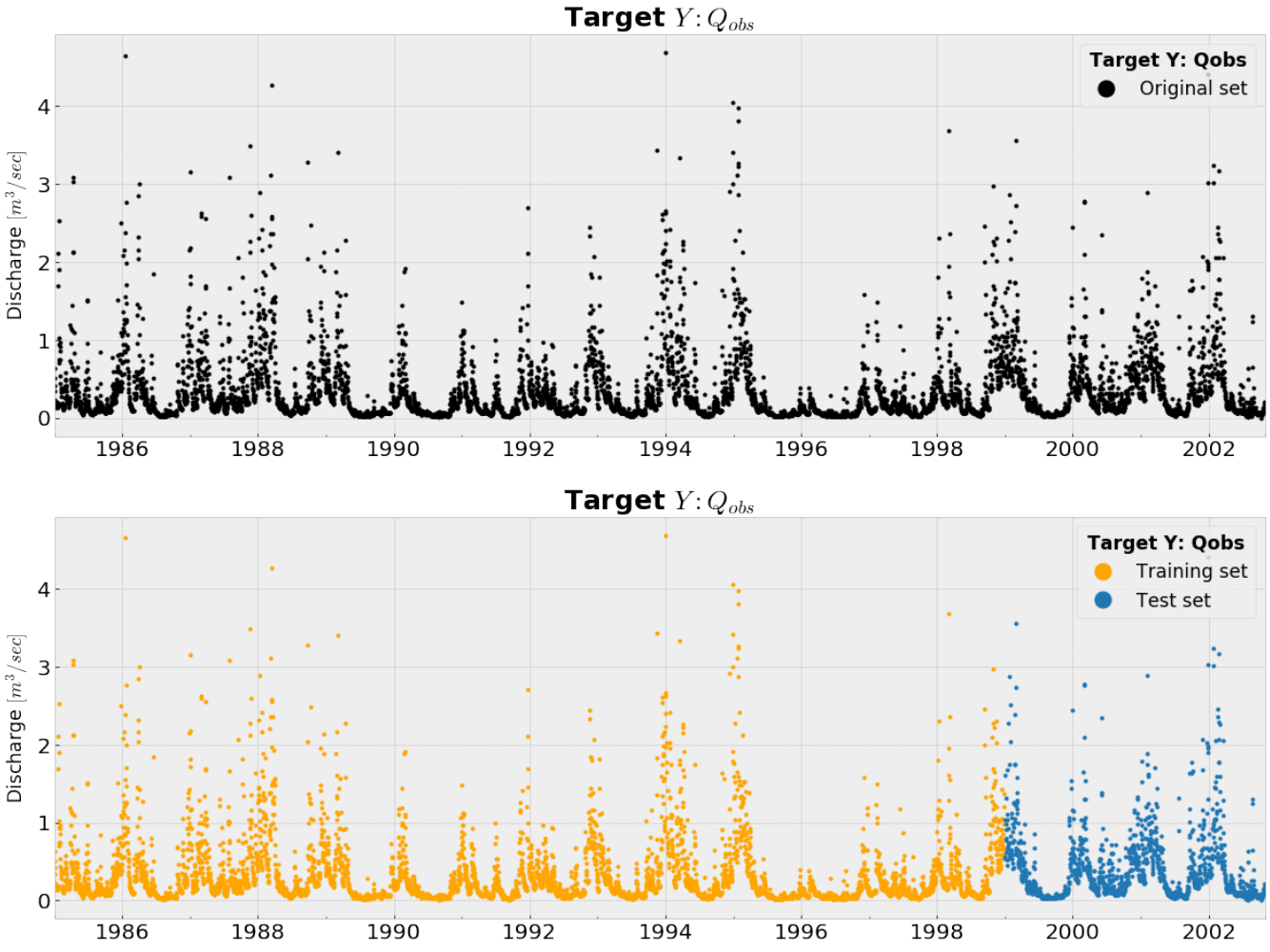


Figure F3. The time series of the target $Qobs$ for model setup 4, divided into the training set (1985-1999) and the test set (1999-2003)

F1.3 Correlation overview dataset for model setup 4

In this Appendix, a correlation heatmap is depicted of the dataset of model setup 4. This figure shows already to which input variable the target $Qobs$ is mostly correlated with.

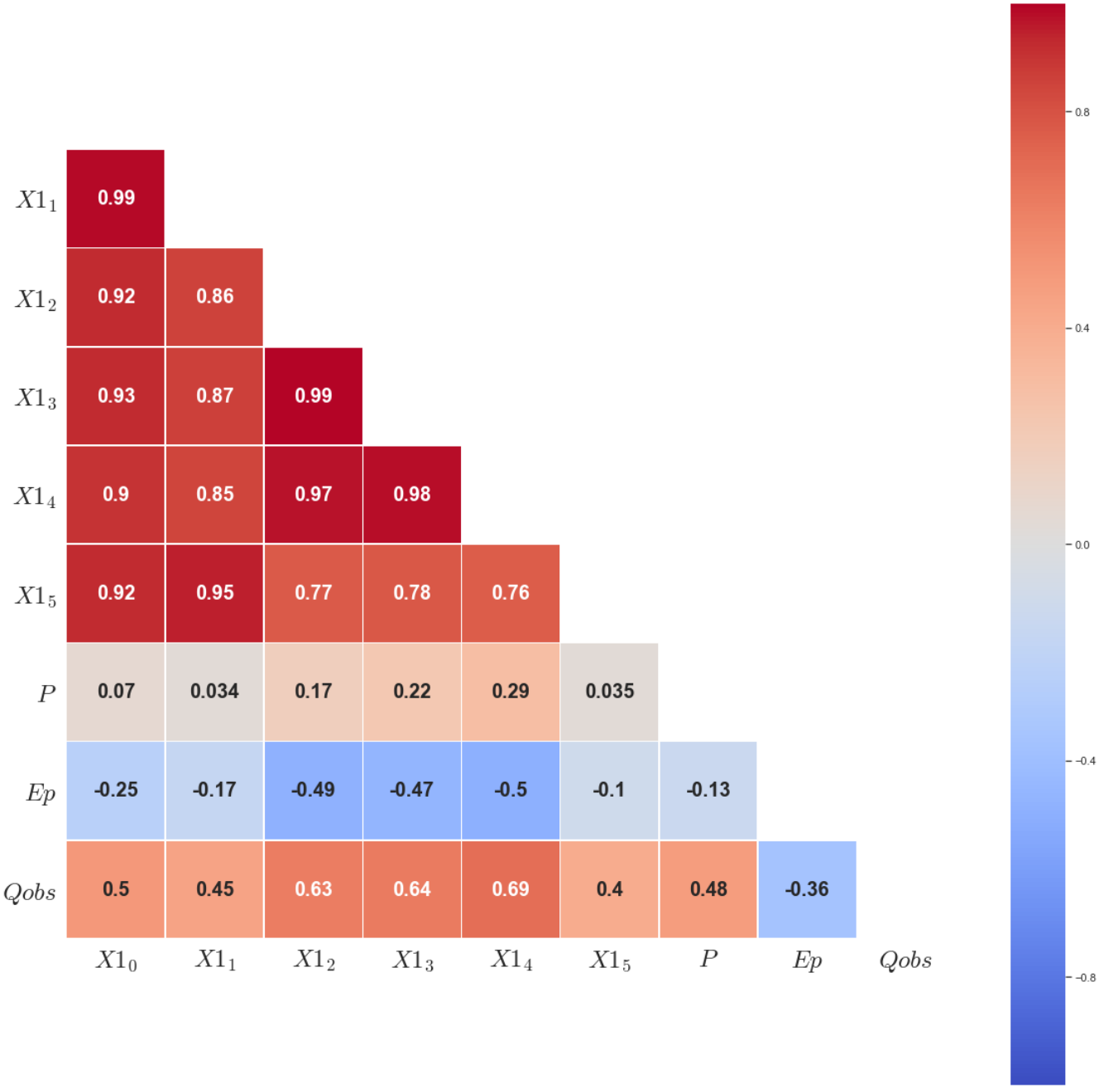


Figure F4. An overview of the correlations of the datatset for model setup 4, depicted in a heatmap

F2 Results - model setup 4

In this Appendix, the results of the different machine learning algorithms of model setup 4 are separately visualised. First, the $Qsim$ time series is plotted for the training and test, followed by a plot of zooming in on the test set. The last figure of each machine learning algorithm is a scatterplot of $Qobs$ against $Qsim$ to easlity detect over- or underfitting.

F2.1 Results DTR - model setup 4

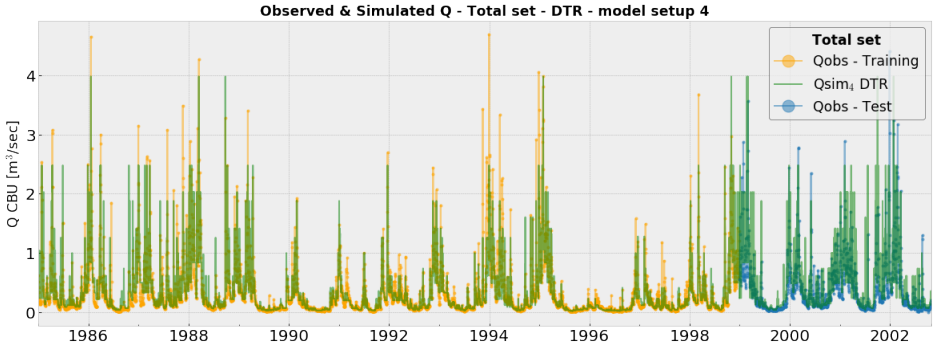


Figure F5. The time series of $Qsim$ and $Qobs$ for the training and test set, for DTR - model setup 4

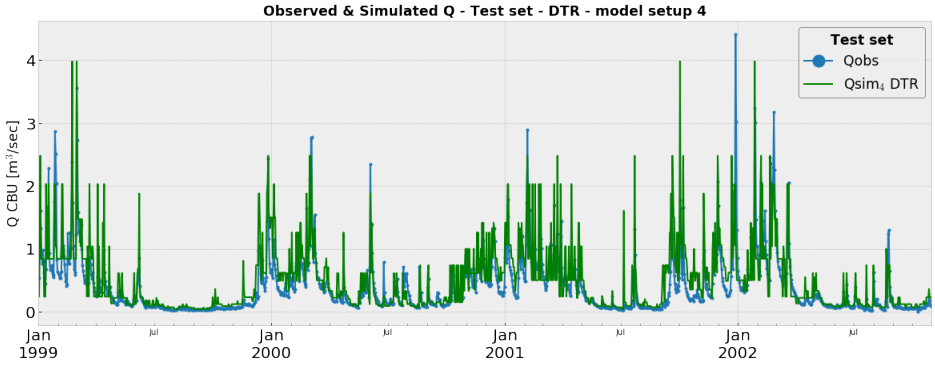


Figure F6. The time series of $Qsim$ and $Qobs$ for only the test set, for DTR - model setup 4

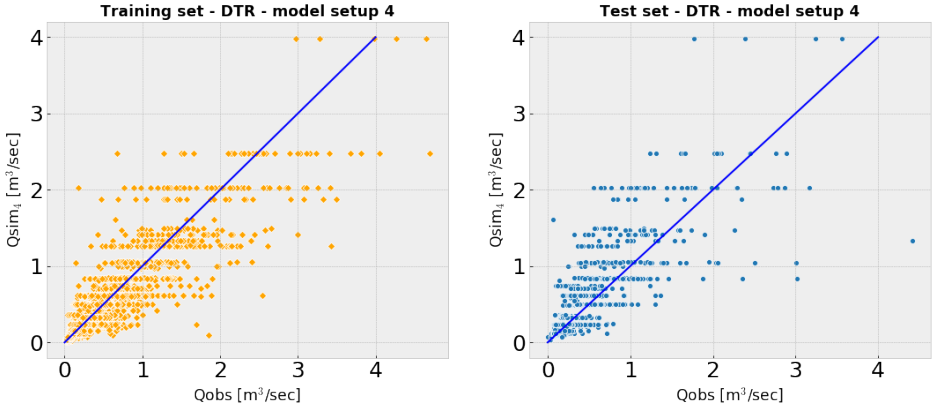


Figure F7. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for DTR - model setup 4

F2.2 Results RFR - model setup 4

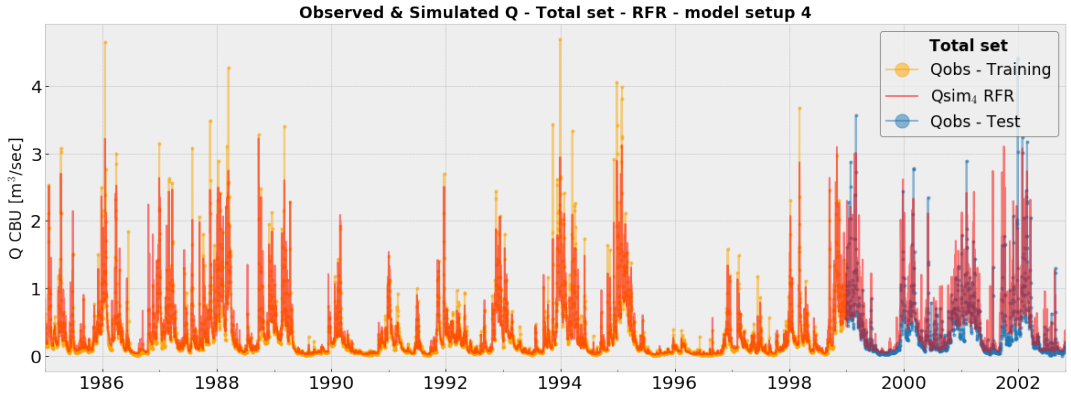


Figure F8. The time series of $Qsim$ and $Qobs$ for the training and test set, for RFR - model setup 4

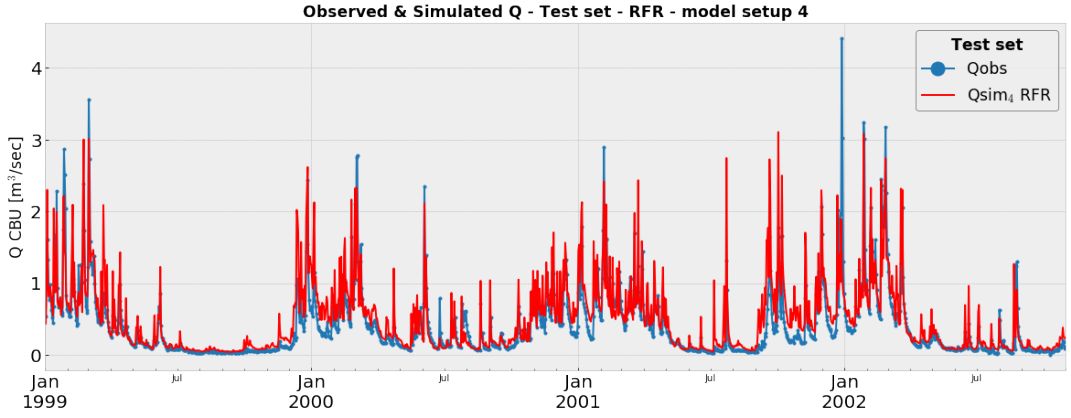


Figure F9. The time series of $Qsim$ and $Qobs$ for only the test set, for RFR - model setup 4

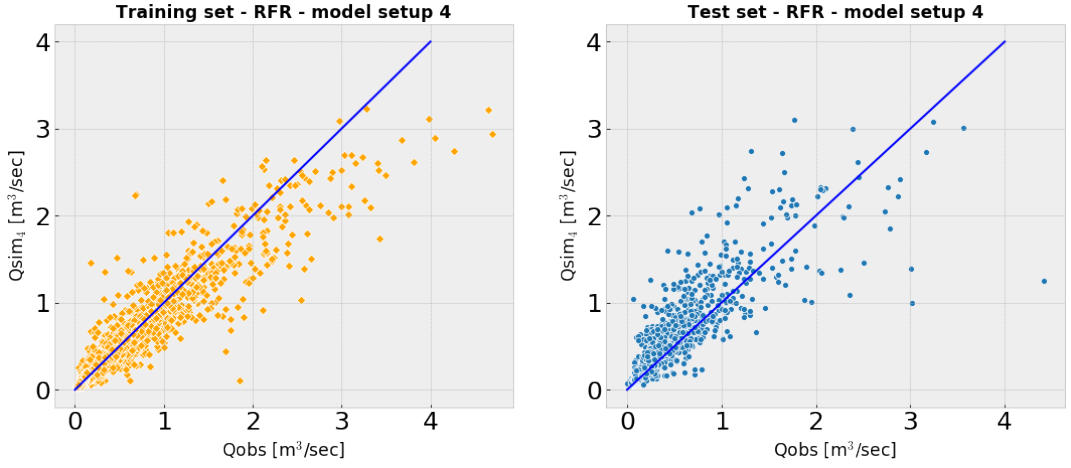


Figure F10. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for RFR - model setup 4

F2.3 Results GBR - model setup 4

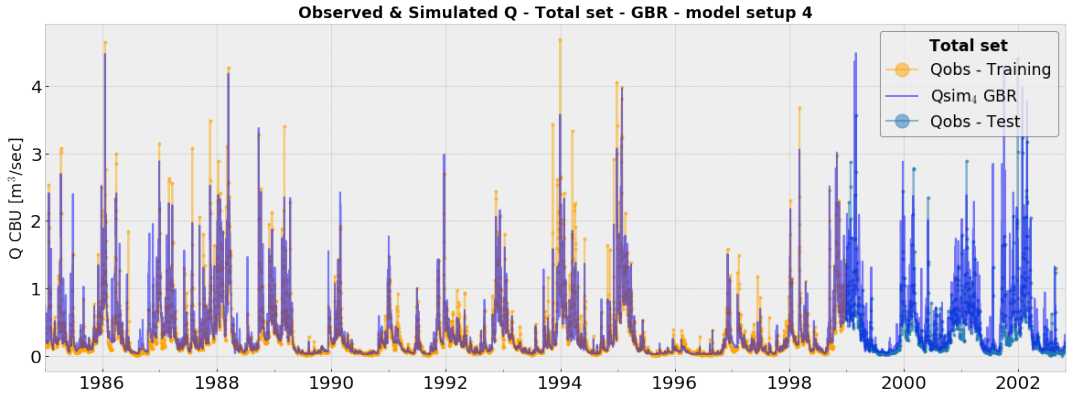


Figure F11. The time series of $Qsim$ and $Qobs$ for the training and test set, for GBR - model setup 4

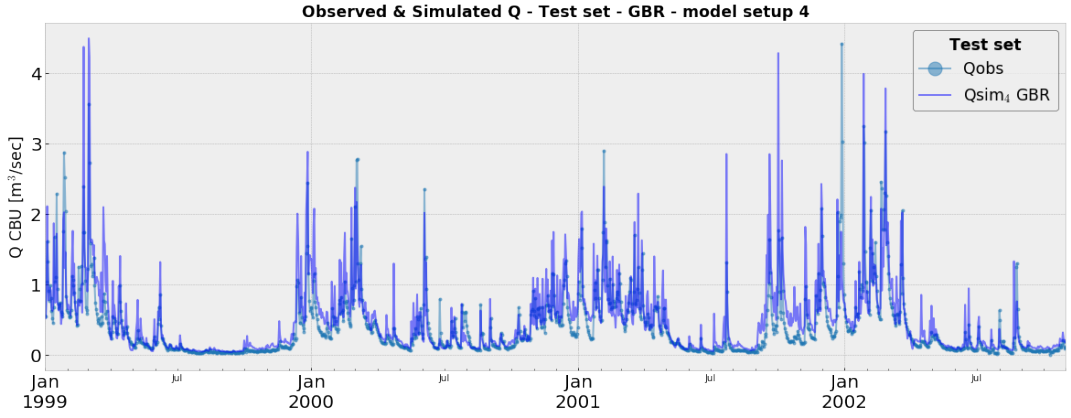


Figure F12. The time series of $Qsim$ and $Qobs$ for only the test set, for GBR - model setup 4

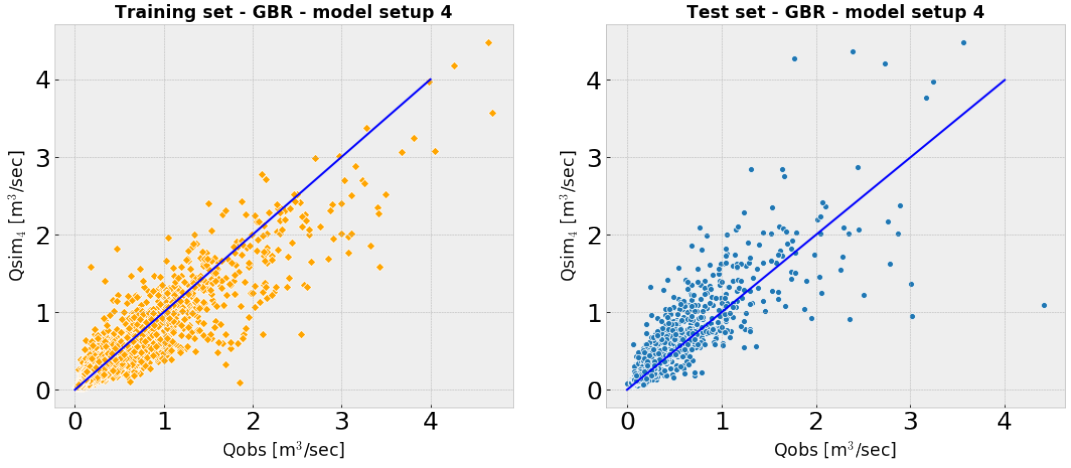


Figure F13. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for GBR - model setup 4

F2.4 Results SVR - model setup 4

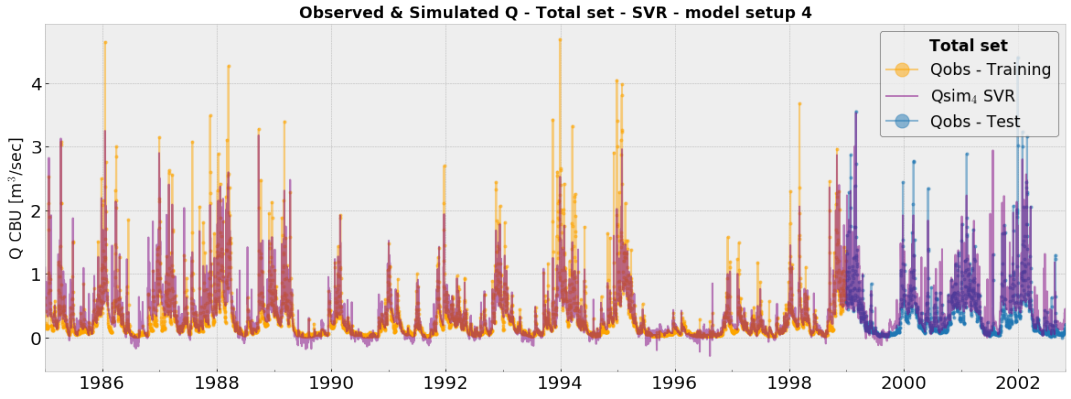


Figure F14. The time series of $Qsim$ and $Qobs$ for the training and test set, for SVR - model setup 4

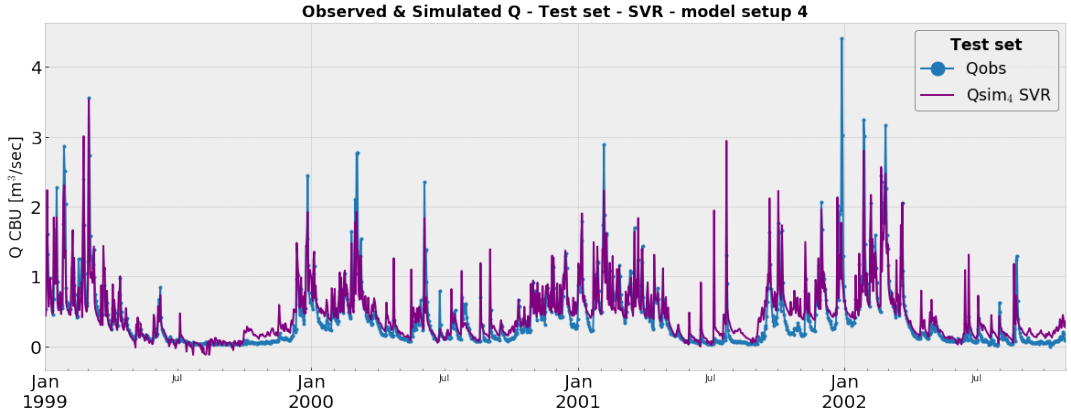


Figure F15. The time series of $Qsim$ and $Qobs$ for only the test set, for SVR - model setup 4

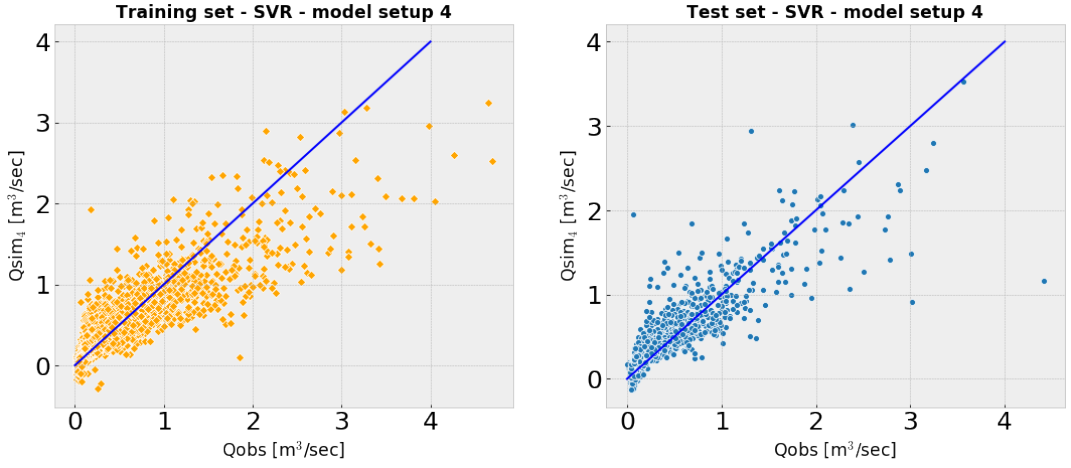


Figure F16. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for SVR - model setup 4

F3 Optimal hyperparameters - model setup 4

In this Appendix, the optimal hyperparameter set found with 5-folds grid search cross validation is depicted for each single machine learning algorithm in a Table. Moreover, the computation time for the hyperparameter tuning is given in the same table.

5-folds grid search cross validation		
Hyperparameters DTR - model setup 4	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	6
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
Computation time	*20.6 sec*	

Table F1. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 4 DTR

5-folds grid search cross validation		
Hyperparameters RFR - model setup 4	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	10
minimum samples in a leaf	1, 2, 4	4
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	250
Computation time	*62.6 min*	

Table F2. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 4 RFR

5-folds grid search cross validation		
Hyperparameters GBR - model setup 4	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	1
minimum samples to obtain a split	2, 5, 10	2
number of regression trees	10, 25, 50, 100, 250	100
Computation time	*360.4 min*	

Table F3. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 4 GBR

5-folds grid search cross validation		
Hyperparameters SVR - model setup 4	**Grid**	**Optimal Hyperparameter**
gamma (kernel coefficient)	0.001, 0.01, 0.1, 1	0.01
C (penalty error parameter)	0.001, 0.01, 0.1, 1, 10	10
Computation time	*18.0 sec*	

Table F4. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 4 SVR

F4 Decision trees DTR - model setup 4

In this Appendix, the regression tree of the DTR of model setup 4 is visualised. This tree has a depth of 6, but only the tree with depth 4 is visualised for simplicity reasons.

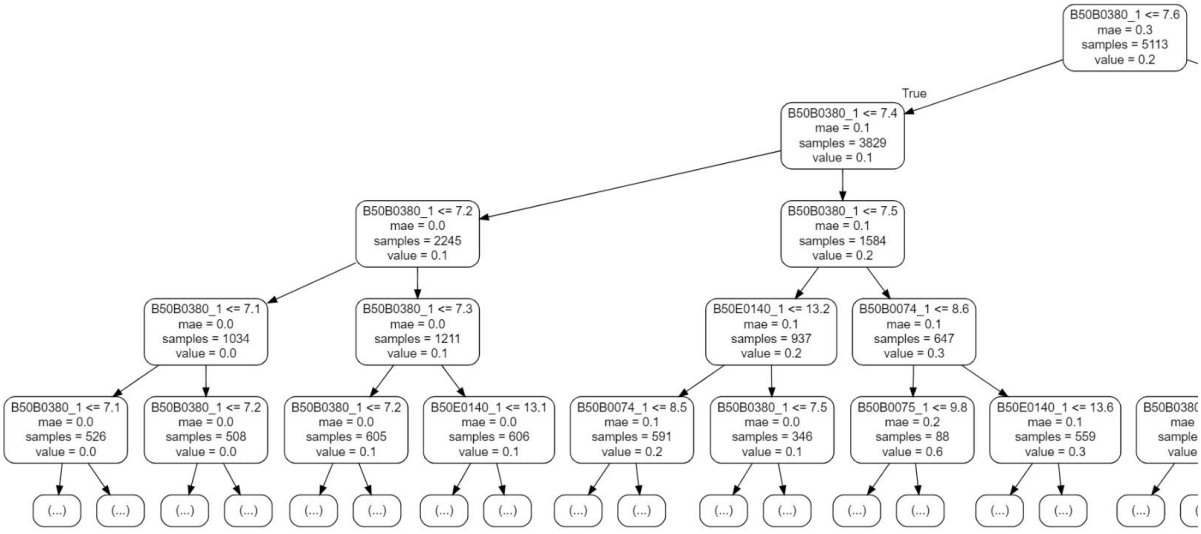


Figure F17. Left part of the regression tree of DTR model setup 4

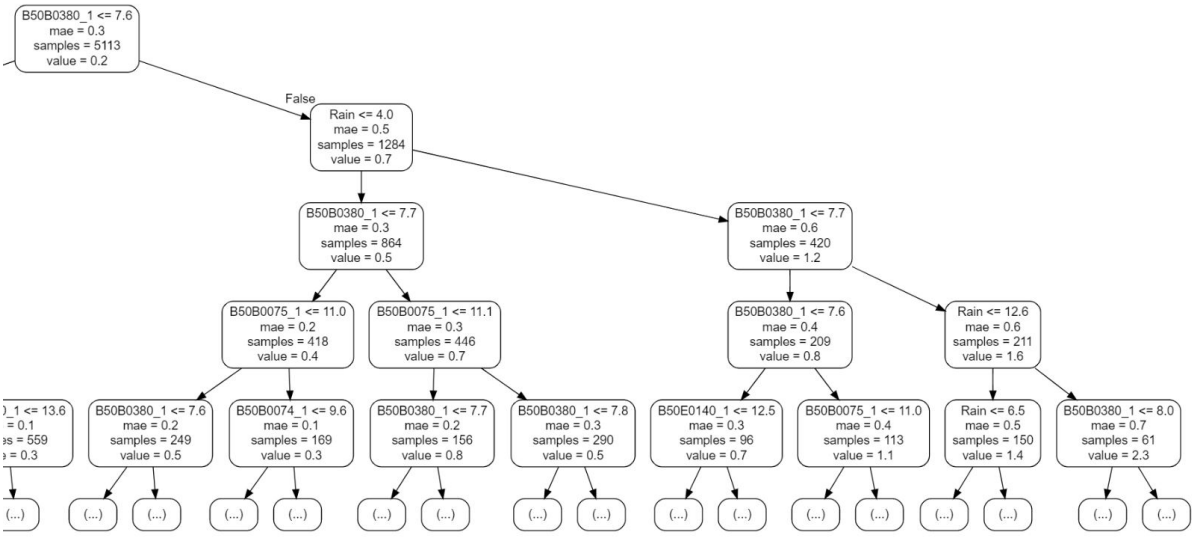


Figure F18. Right part of the regression tree of DTR model setup 4

Appendix G: model setup 5

G1 Dataset for model setup 5

In this Appendix, the time series of the input variables $X1_0$-$X1_5$, P, Ep including the rolling means (R3, R7, R14, R30, R120 days for $X1$ wells) and the shifts (S1, S2, S3, S4, S5, S6 days for P and Ep), and the target $Qobs$ for model setup 5 are visualised. A division is made between the training set and the test set.

G1.1 Input variables for model setup 5

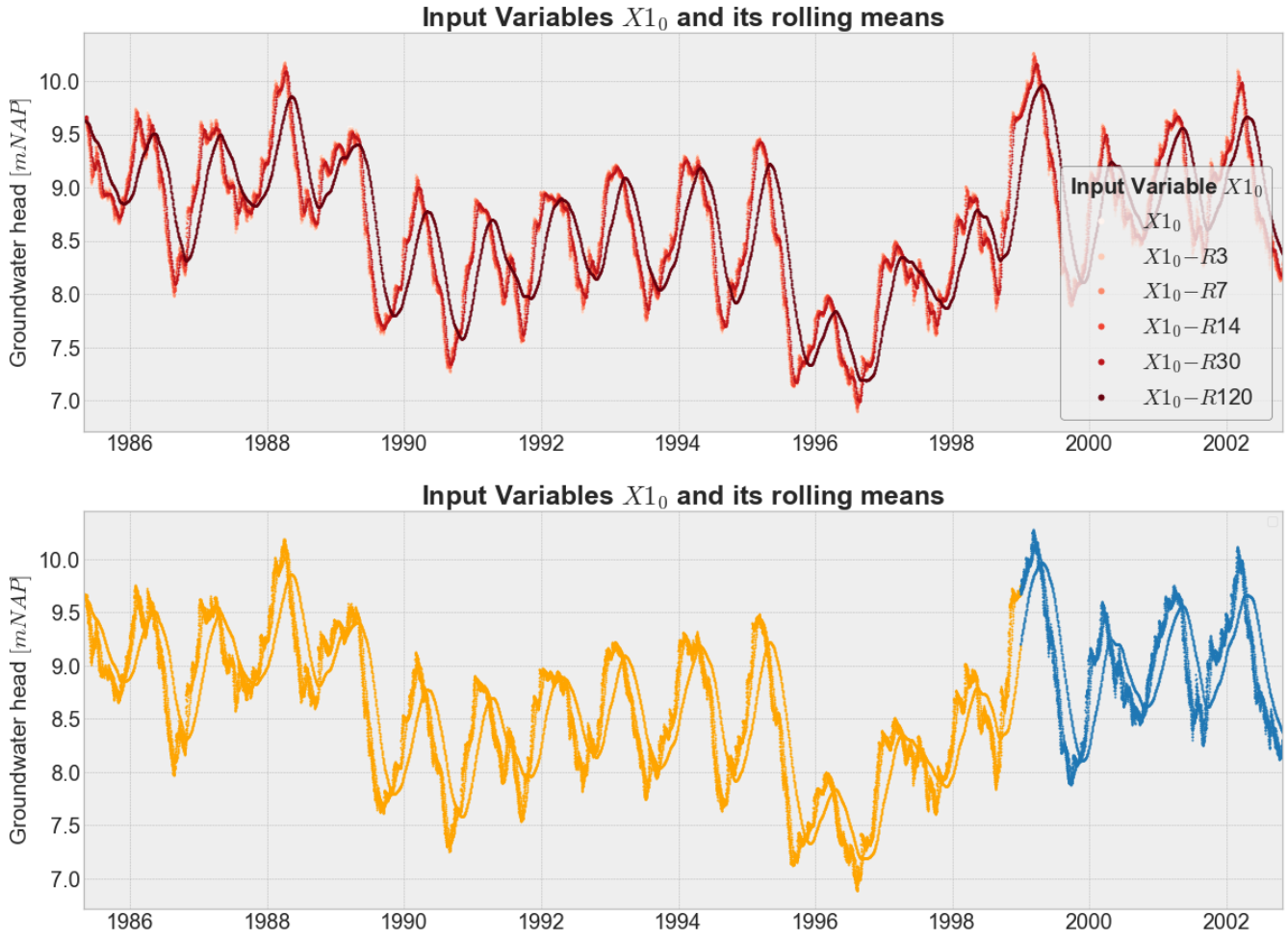


Figure G1. The time series of the input variables screen-1 well $X1_0$ and its rolling means $X1_0 - R3$, $X1_0 - R7$, $X1_0 - R14$, $X1_0 - R30$, $X1_0 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

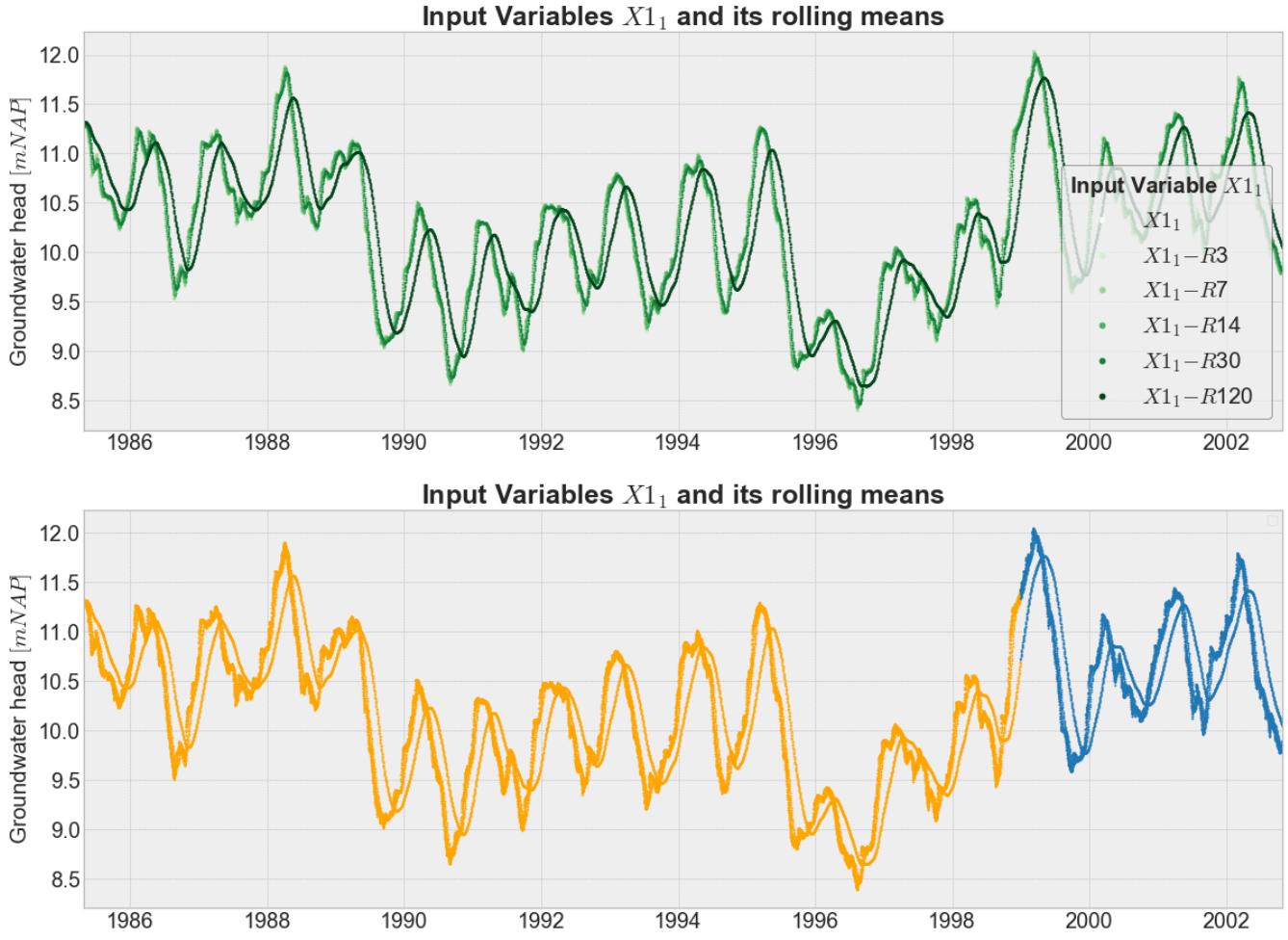


Figure G2. The time series of the input variables screen-1 well $X1_1$ and its rolling means $X1_1 - R3$, $X1_1 - R7$, $X1_1 - R14$, $X1_1 - R30$, $X1_1 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

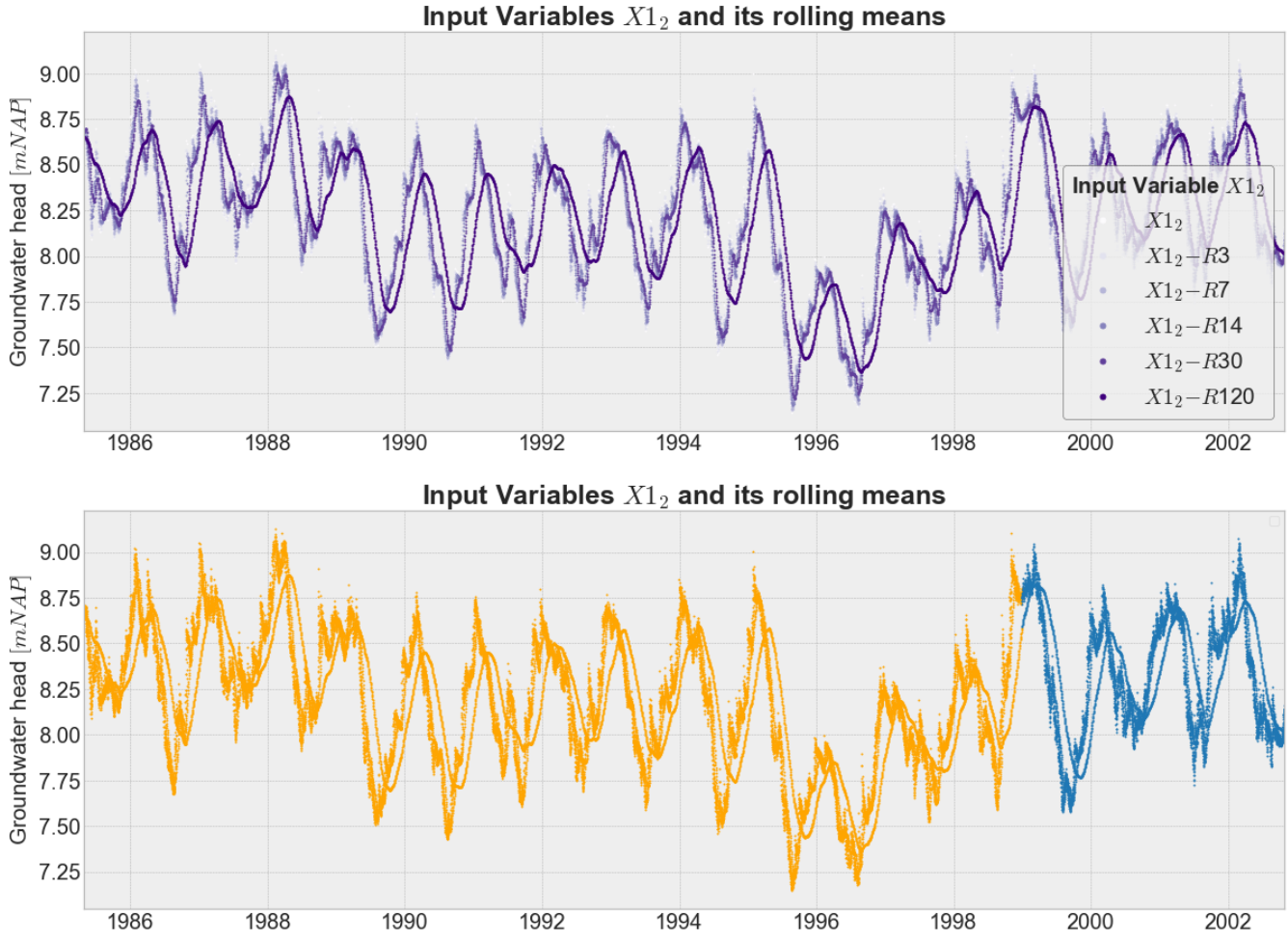


Figure G3. The time series of the input variables screen-1 well $X1_2$ and its rolling means $X1_2 - R3$, $X1_2 - R7$, $X1_2 - R14$, $X1_2 - R30$, $X1_2 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

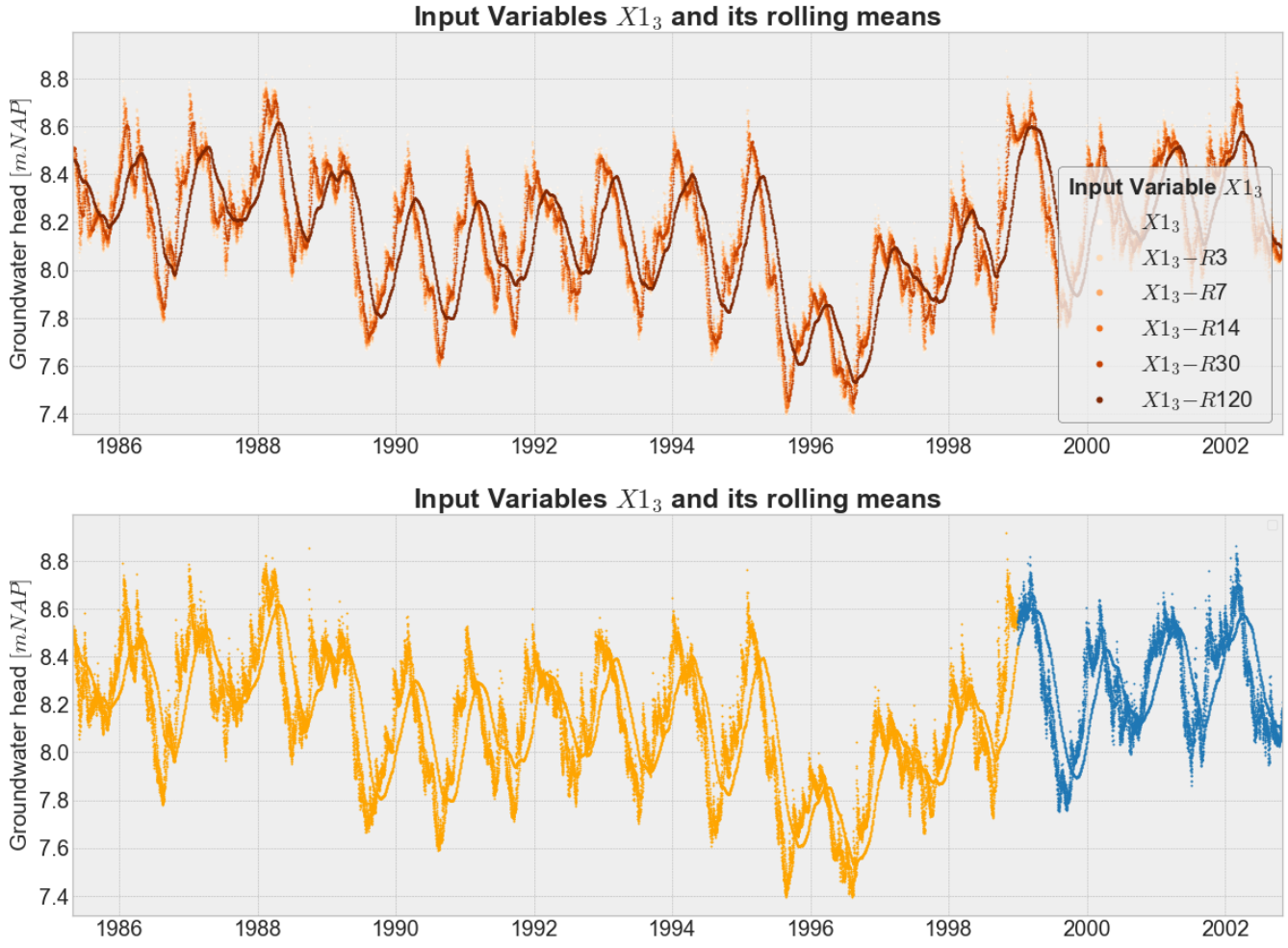


Figure G4. The time series of the input variables screen-1 well $X1_3$ and its rolling means $X1_3 - R3$, $X1_3 - R7$, $X1_3 - R14$, $X1_3 - R30$, $X1_3 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

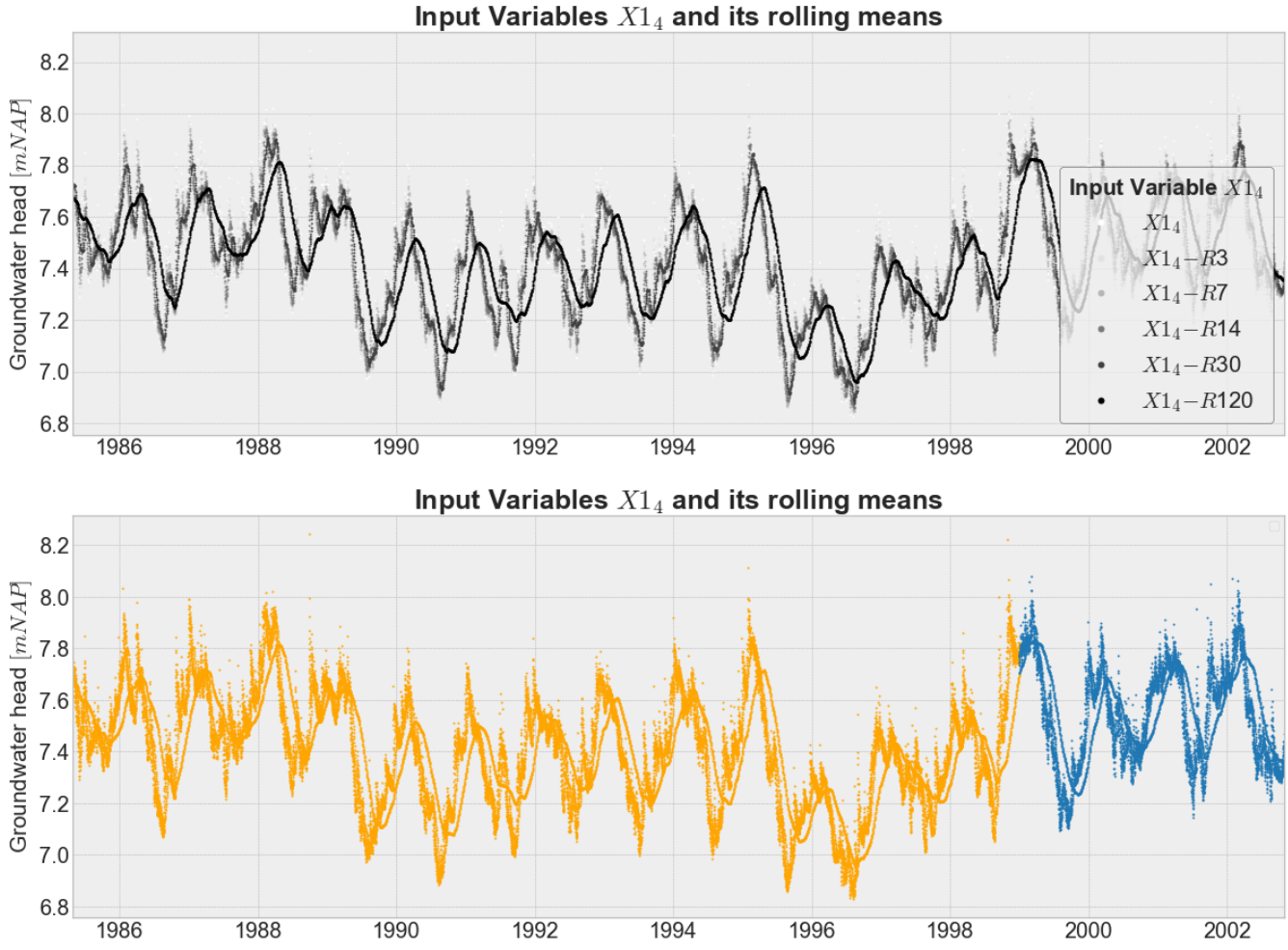


Figure G5. The time series of the input variables screen-1 well $X1_4$ and its rolling means $X1_4 - R3$, $X1_4 - R7$, $X1_4 - R14$, $X1_4 - R30$, $X1_4 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

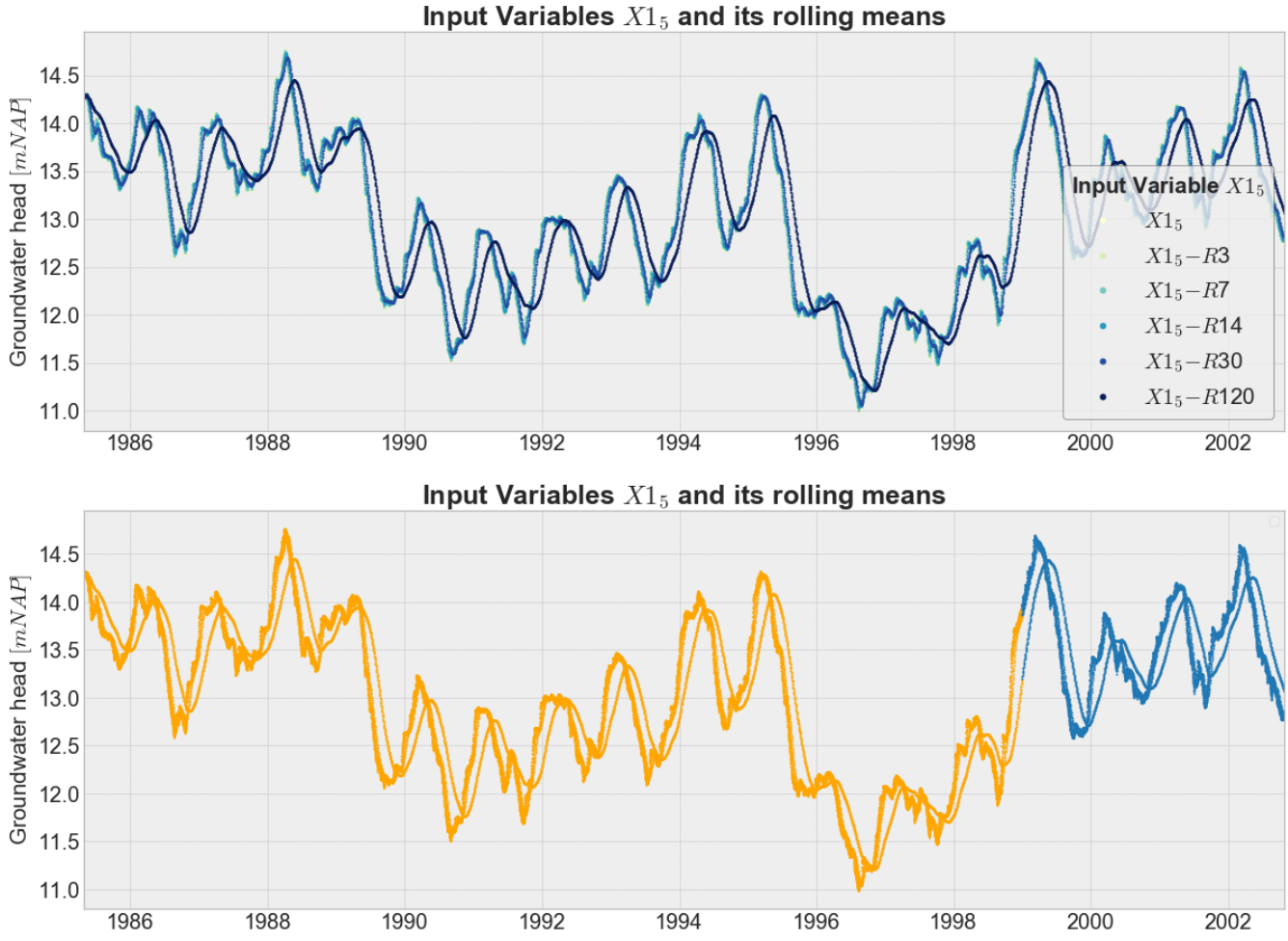


Figure G6. The time series of the input variables screen-1 well $X1_5$ and its rolling means $X1_5 - R3$, $X1_5 - R7$, $X1_5 - R14$, $X1_5 - R30$, $X1_5 - R120$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

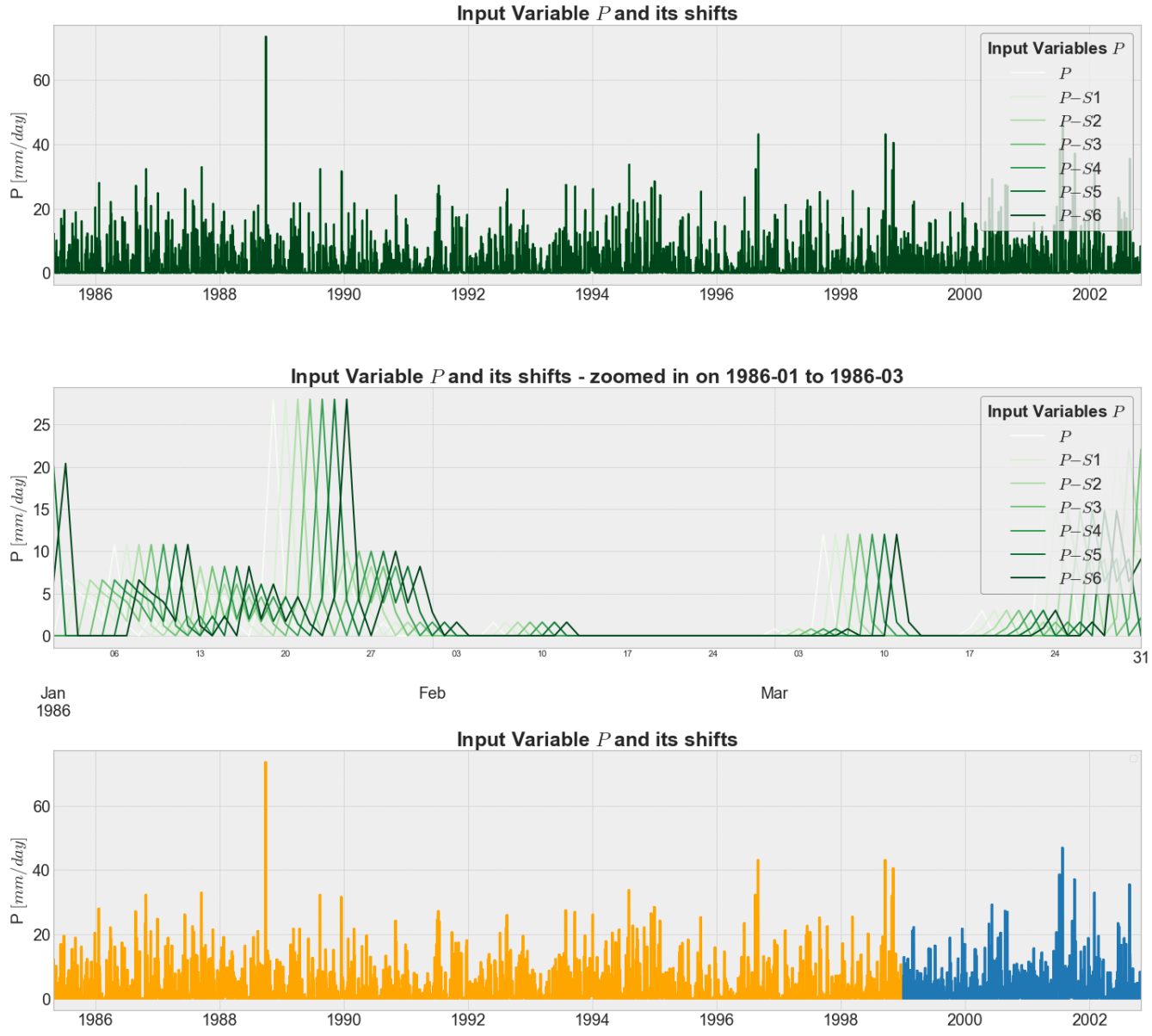


Figure G7. The time series of the input variables P and its shifts $P-S1$, $P-S2$, $P-S3$, $P-S4$, $P-S5$ and $P-S6$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

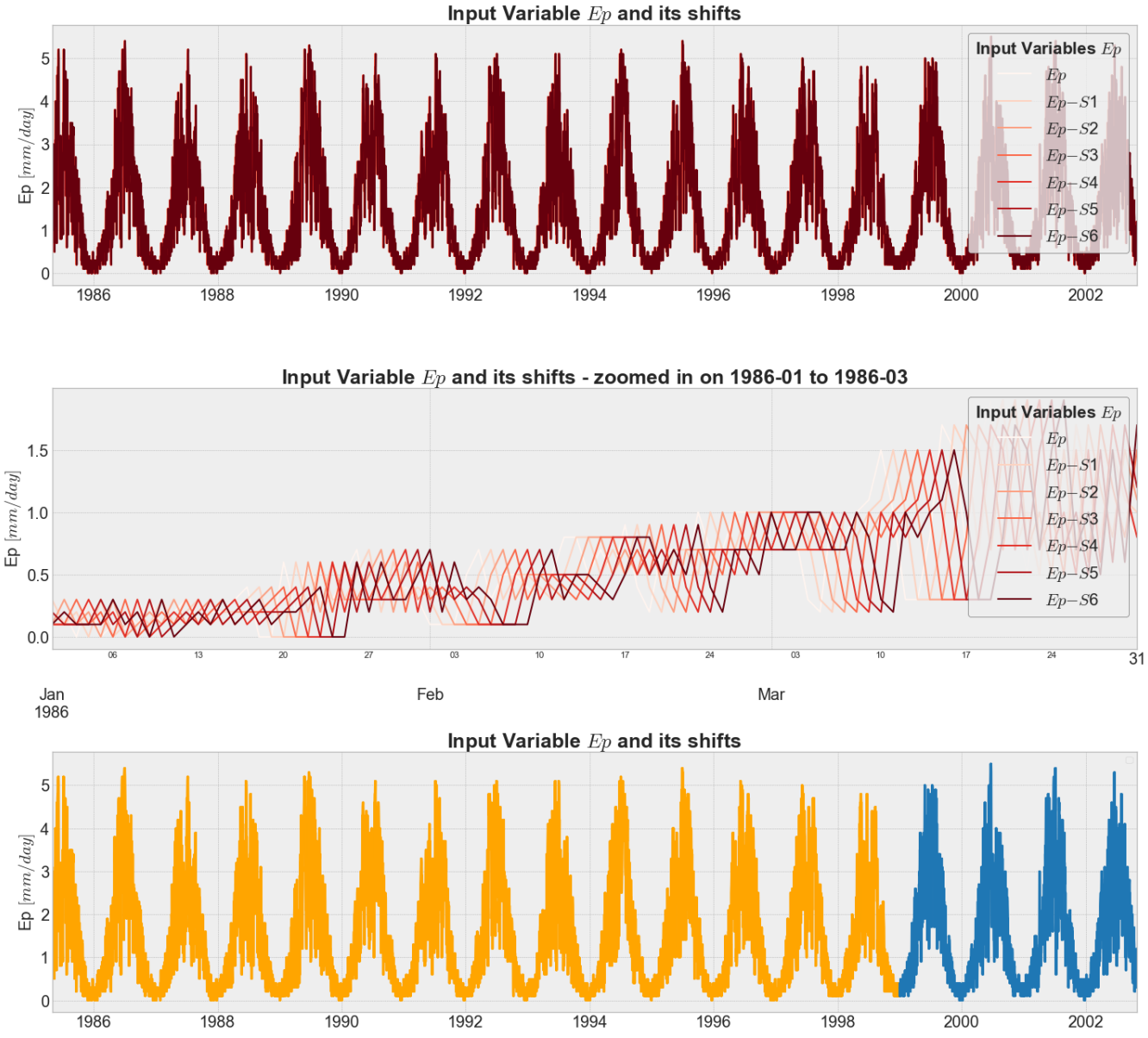


Figure G8. The time series of the input variables Ep and its shifts $Ep-S1$, $Ep-S2$, $Ep-S3$, $Ep-S4$, $Ep-S5$ and $Ep-S6$ for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

G1.2 Target for model setup 5

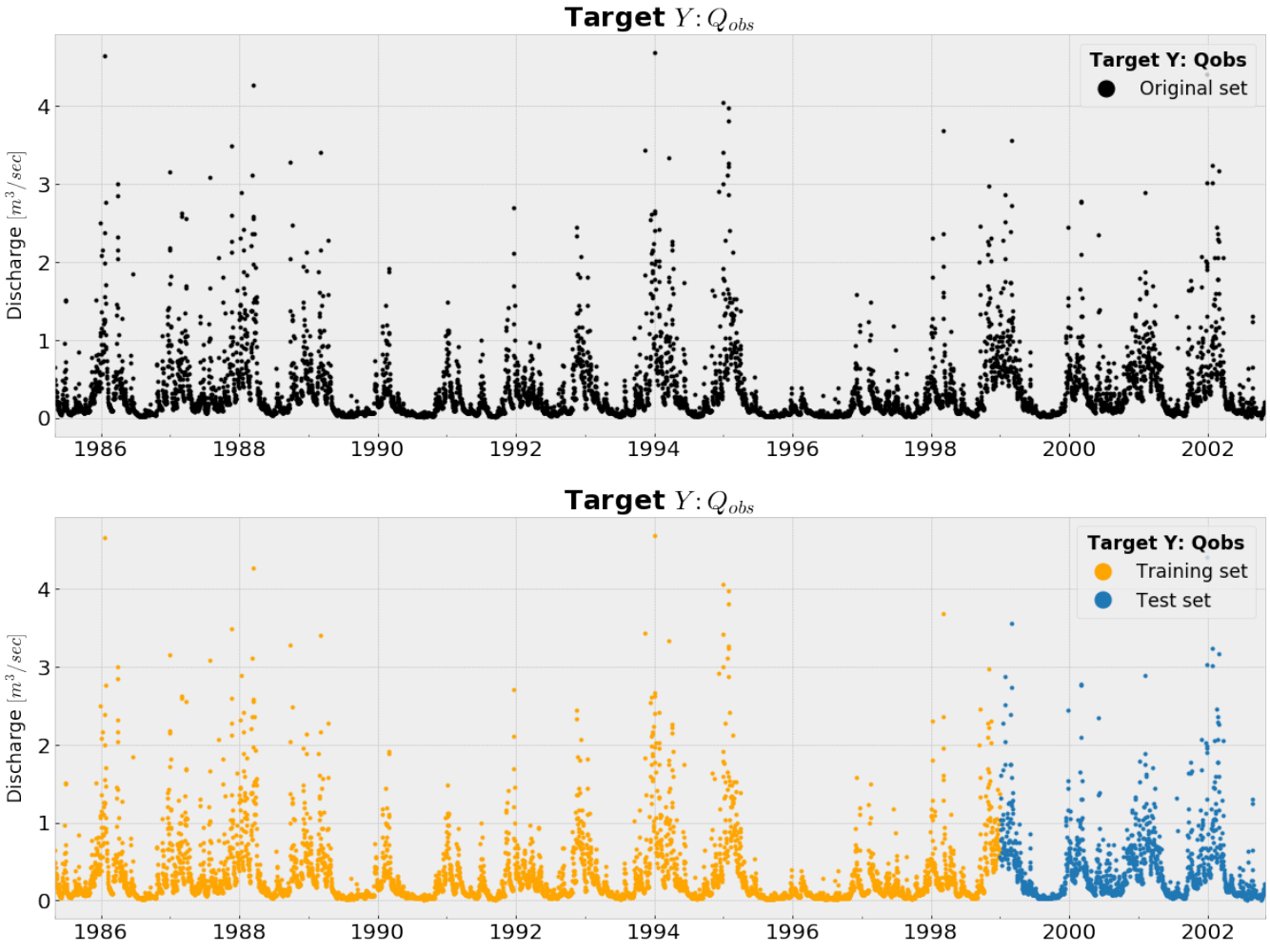


Figure G9. The time series of the target *Qobs* for model setup 5, divided into the training set (1985-1999) and the test set (1999-2003)

G1.3 Correlation overview dataset for model setup 5 - used for DTR and SVR

In this Appendix, a horizontal barplot is depicted of the correlation of all input variables of model setup 5 with the target $Qobs$. This figure shows already to which input variable the target $Qsobs$ is mostly correlated with. There are in total 50 input variables and 1 target. For DTR and SVR all these variables are taken into account, since these algorithms will not result in a very long computation time.

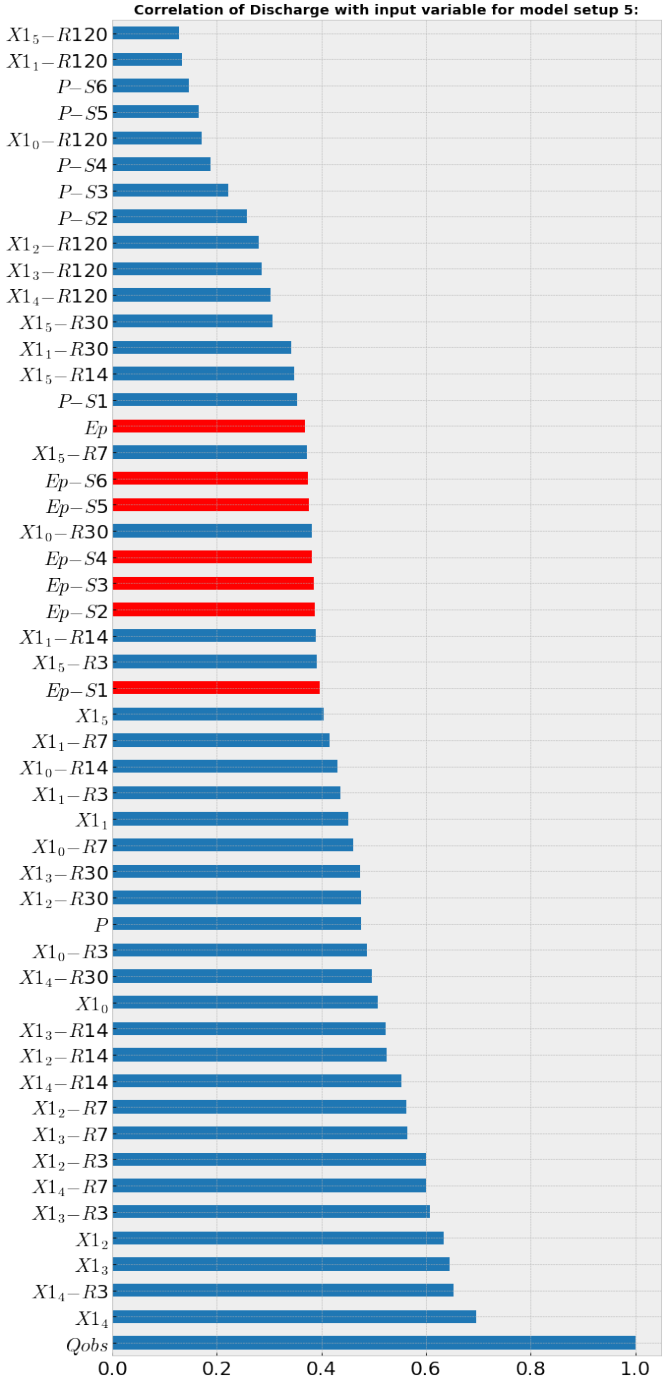


Figure G10. An overview of the correlations of the target $Qobs$ with the input variables of the datatset for model setup 5 used for DTR and SVR, depicted in a horizontal barplot. Note that red means a negative correlation with $Qobs$ and blue a positive correlation with $Qobs$.

G1.4 Correlation overview dataset for model setup 5 - used for RFR and GBR

Taking all input variables into account for RFR and GBR, will lead to a really long computation time. Therefore, for RFR and GBR only the 10 variables with the highest relative importance are taken into account, which are from high to low relative importance:$X1_4$, P, $P-S1$, $X1_0-R14$, $P-S2$, $X1_0$, $X1_1$, $X1_2$, $X1_3$ and $X1_5$. A correlation heatmap of these variables (including the target $Qobs$) is depicted in the Figure below.

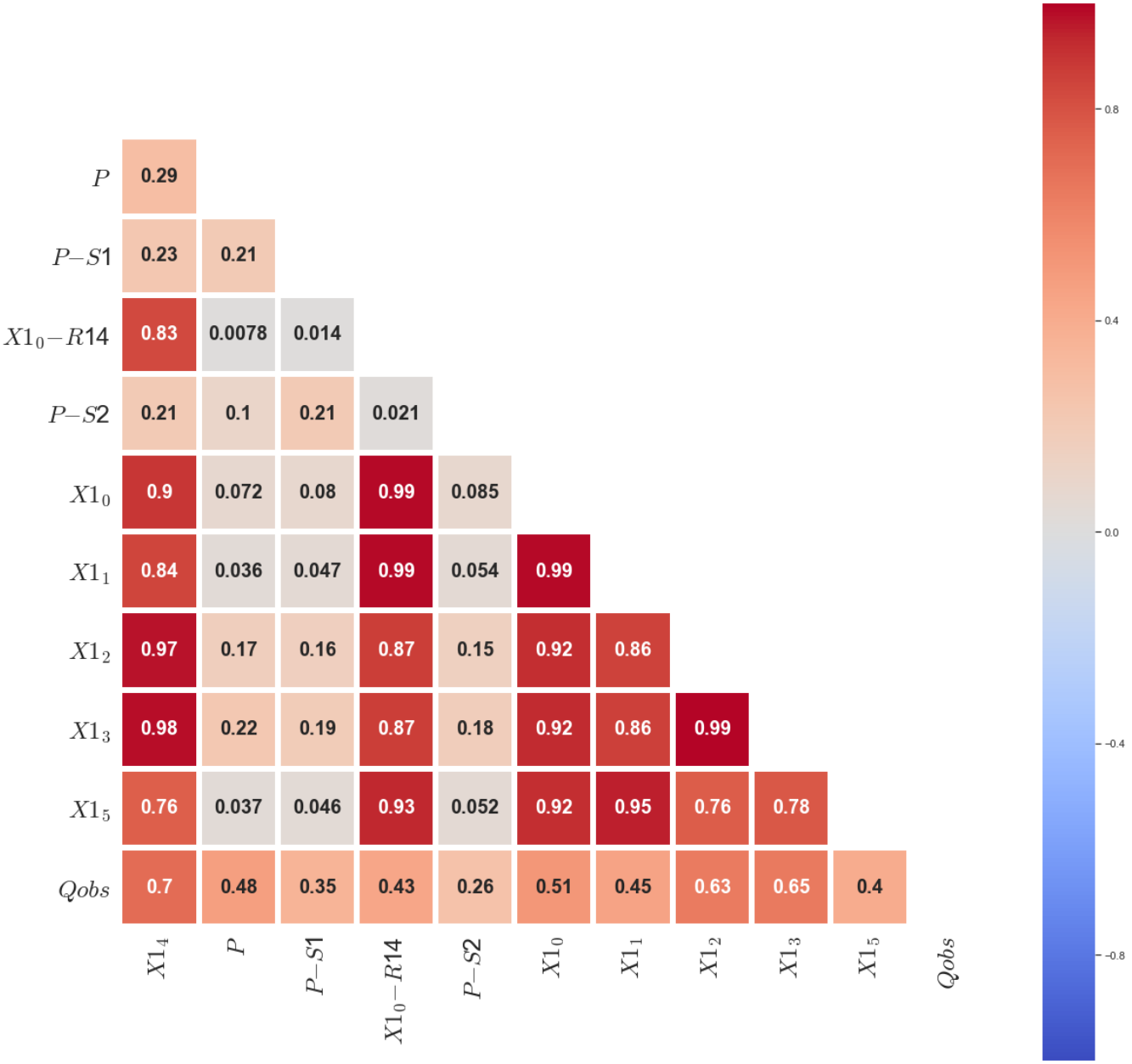


Figure G11. An overview of the correlations of the datatset for model setup 5 used for RFR and GBR, depicted in a heatmap.

G2 Results - model setup 5

In this Appendix, the results of the different machine learning algorithms of model setup 5 are separately visualised. First, the $Qsim$ time series is plotted for the training and test, followed by a plot of zooming in on the test set. The last figure of each machine learning algorithm is a scatterplot of $Qobs$ against $Qsim$ to easlity detect over- or underfitting.

G2.1 Results DTR - model setup 5

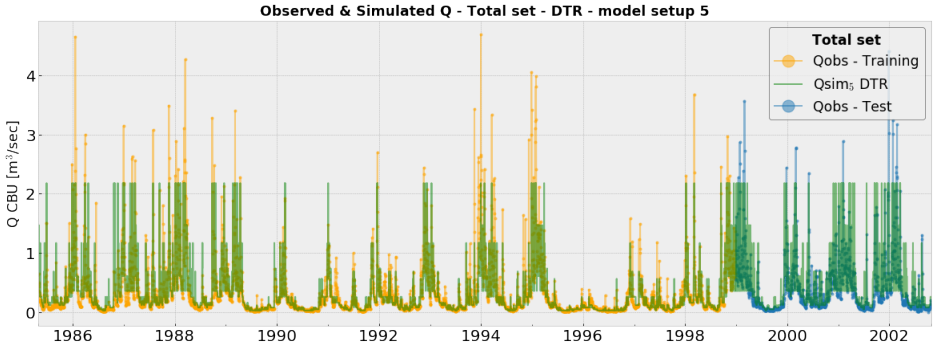


Figure G12. The time series of $Qsim$ and $Qobs$ for the training and test set, for DTR - model setup 5

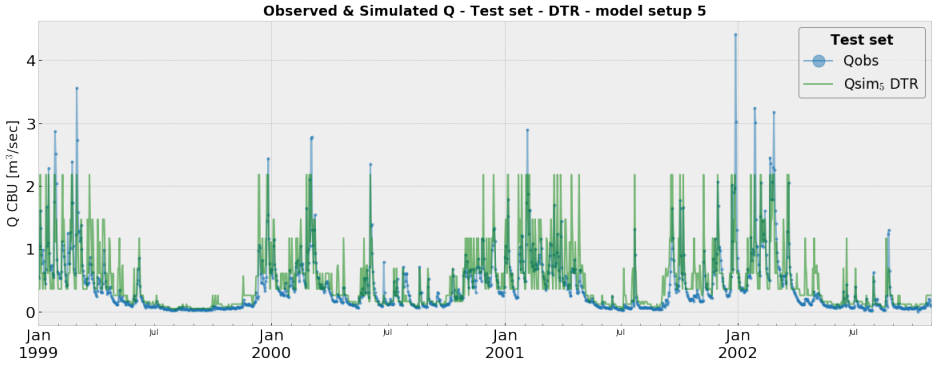


Figure G13. The time series of $Qsim$ and $Qobs$ for only the test set, for DTR - model setup 5

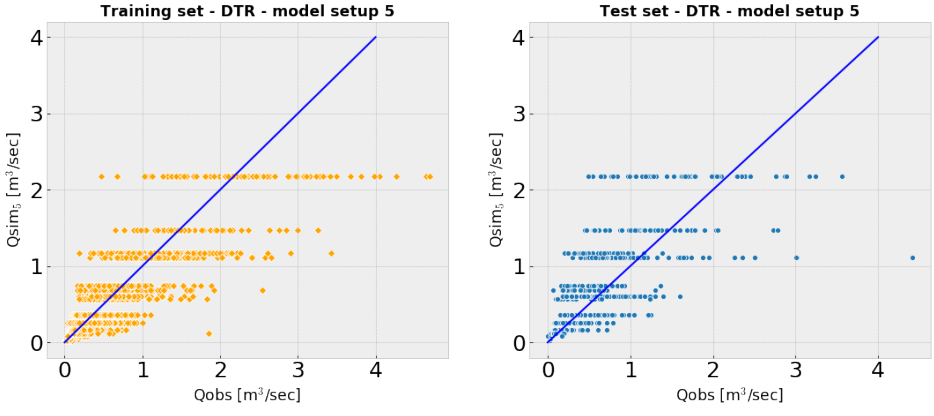


Figure G14. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for DTR - model setup 5

G2.2 Results RFR - model setup 5

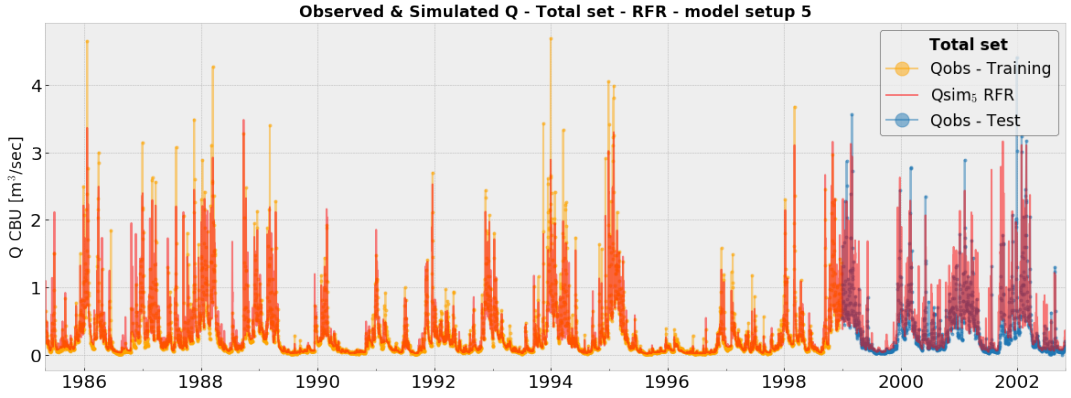


Figure G15. The time series of $Qsim$ and $Qobs$ for the training and test set, for RFR - model setup 5

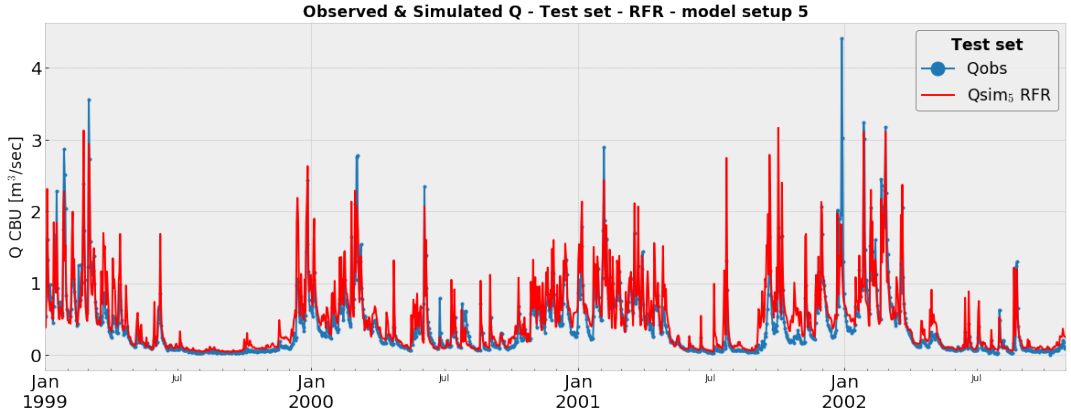


Figure G16. The time series of $Qsim$ and $Qobs$ for only the test set, for RFR - model setup 5

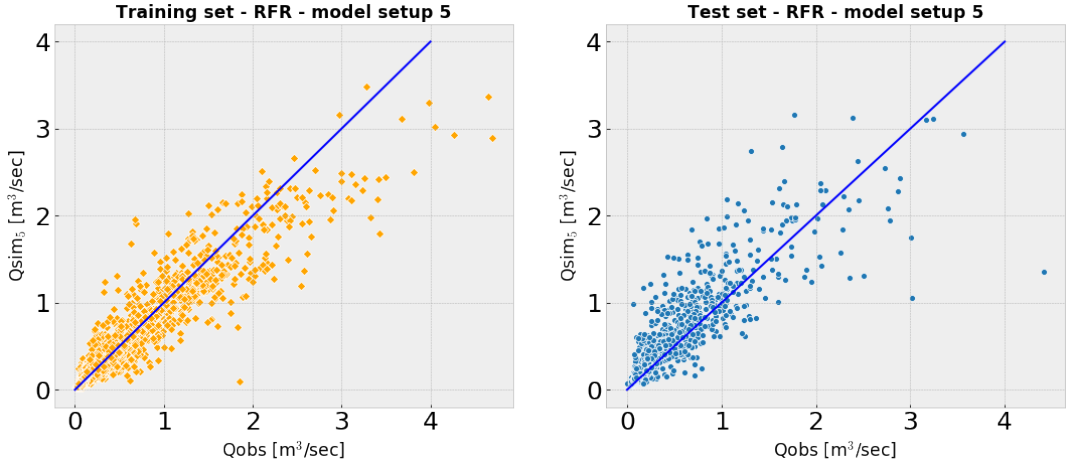


Figure G17. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for RFR - model setup 5

G2.3 Results GBR - model setup 5

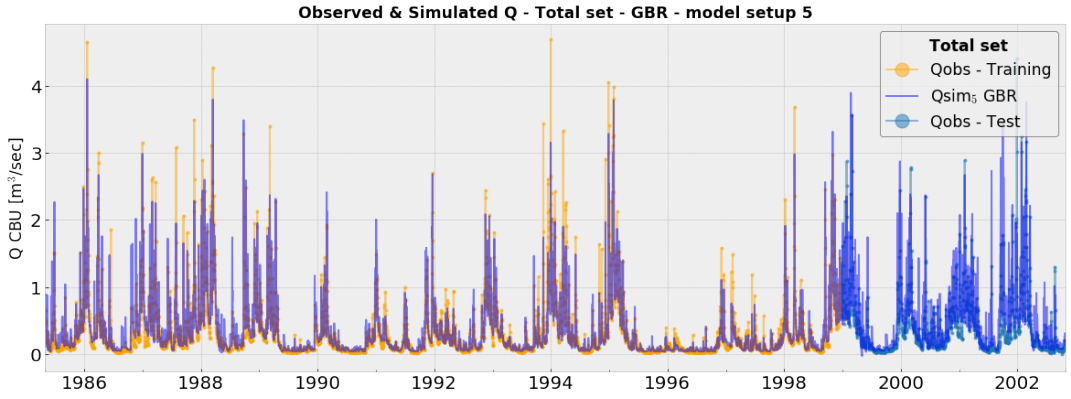


Figure G18. The time series of $Qsim$ and $Qobs$ for the training and test set, for GBR - model setup 5

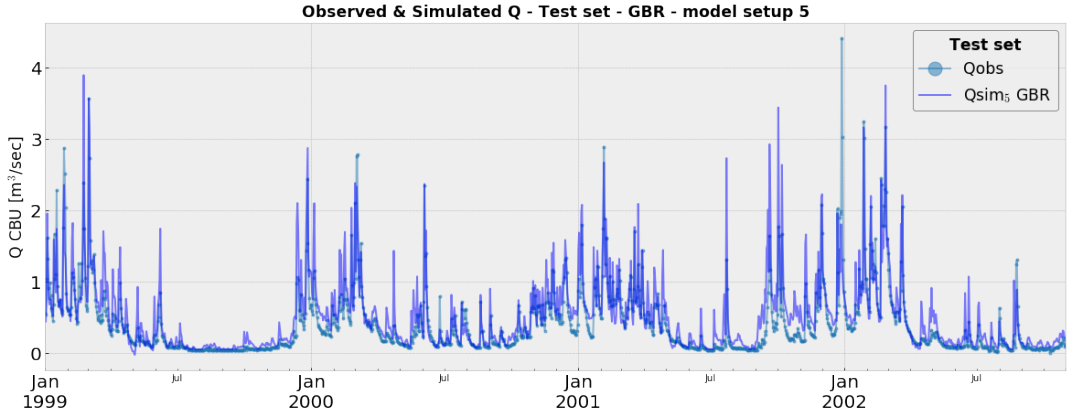


Figure G19. The time series of $Qsim$ and $Qobs$ for only the test set, for GBR - model setup 5

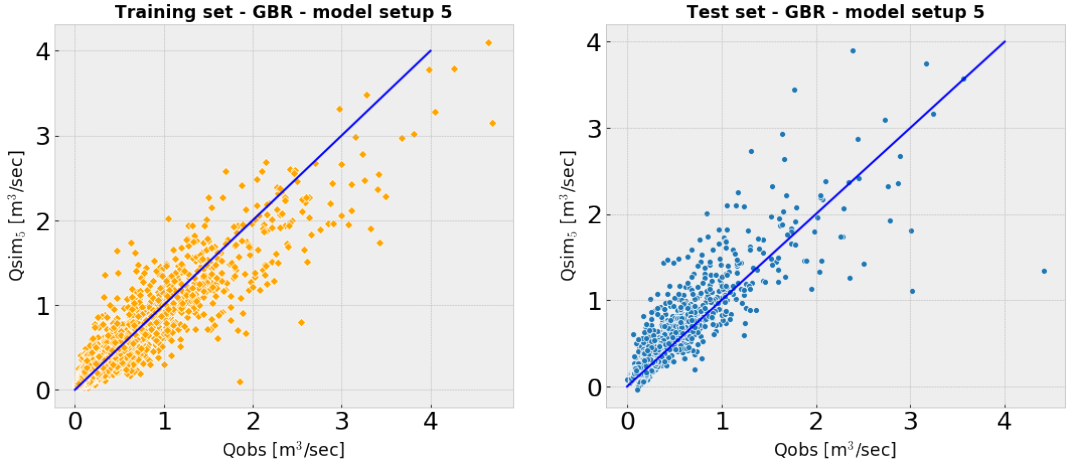


Figure G20. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for GBR - model setup 5

G2.4 Results SVR - model setup 5

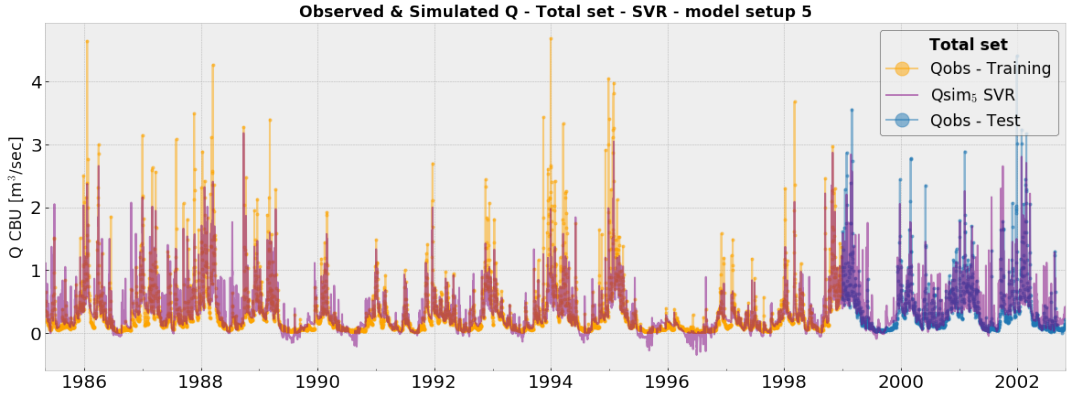


Figure G21. The time series of $Qsim$ and $Qobs$ for the training and test set, for SVR - model setup 5

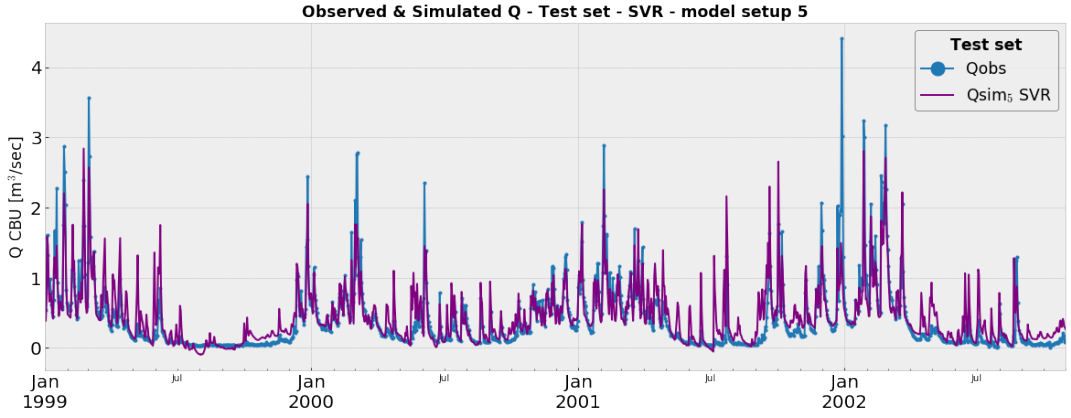


Figure G22. The time series of $Qsim$ and $Qobs$ for only the test set, for SVR - model setup 5

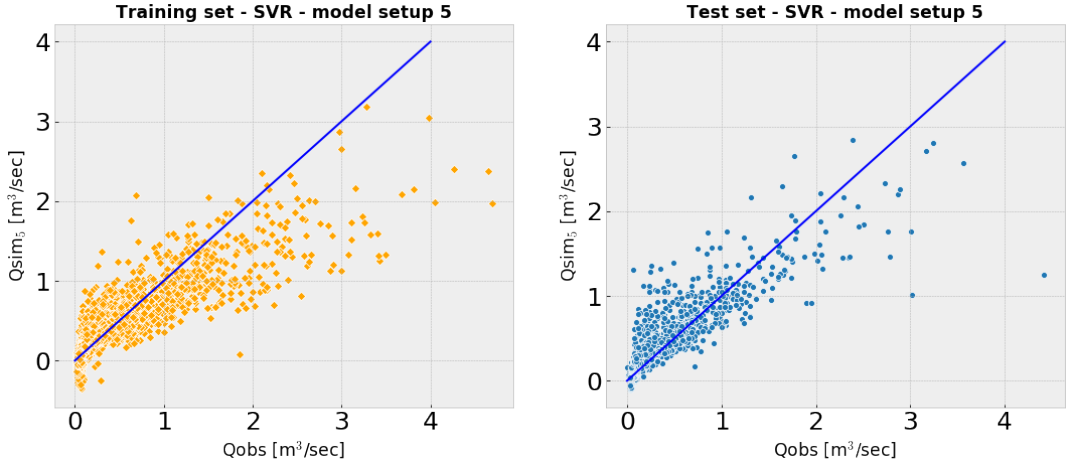


Figure G23. Scatterplots of $Qobs$ versus $Qsim$ for the training and test set, for SVR - model setup 5

G3 Optimal hyperparameters - model setup 5

In this Appendix, the optimal hyperparameter set found with 5-folds grid search cross validation is depicted for each single machine learning algorithm in a Table. Moreover, the computation time for the hyperparameter tuning is given in the same table.

	5-folds grid search cross validation	
Hyperparameters DTR - model setup 5	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	4
minimum samples in a leaf	1, 2, 4	1
minimum samples to obtain a split	2, 5, 10	2
Computation time	*3.9 min*	

Table G1. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 5 DTR

	5-folds grid search cross validation	
Hyperparameters RFR - model setup 5	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MAE
maximum tree depth	2, 4, 6, 8, 10	8
minimum samples in a leaf	1, 2, 4	2
minimum samples to obtain a split	2, 5, 10	5
number of regression trees	10, 25, 50, 100, 250	25
Computation time	*141.5 min*	

Table G2. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 5 RFR

	5-folds grid search cross validation	
Hyperparameters GBR - model setup 5	**Grid**	**Optimal Hyperparameter**
partition criteria	MSE, MAE	MSE
maximum tree depth	2, 4, 6, 8, 10	2
minimum samples in a leaf	1, 2, 4	1
minimum samples to obtain a split	2, 5, 10	10
number of regression trees	10, 25, 50, 100, 250	100
Computation time	*328.3 min*	

Table G3. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 5 GBR

	5-folds grid search cross validation	
Hyperparameters SVR - model setup 5	**Grid**	**Optimal Hyperparameter**
gamma (kernel coefficient)	0.001, 0.01, 0.1, 1	0.001
C (penalty error parameter)	0.001, 0.01, 0.1, 1, 10	10
Computation time	*9.8 sec*	

Table G4. Optimal hyperparameters found with 5-folds grid search cross validation - model setup 5 SVR

G4 Decision trees DTR - model setup 5

In this Appendix, the regression tree of the DTR of model setup 5 is visualised. This tree has a depth of 4.

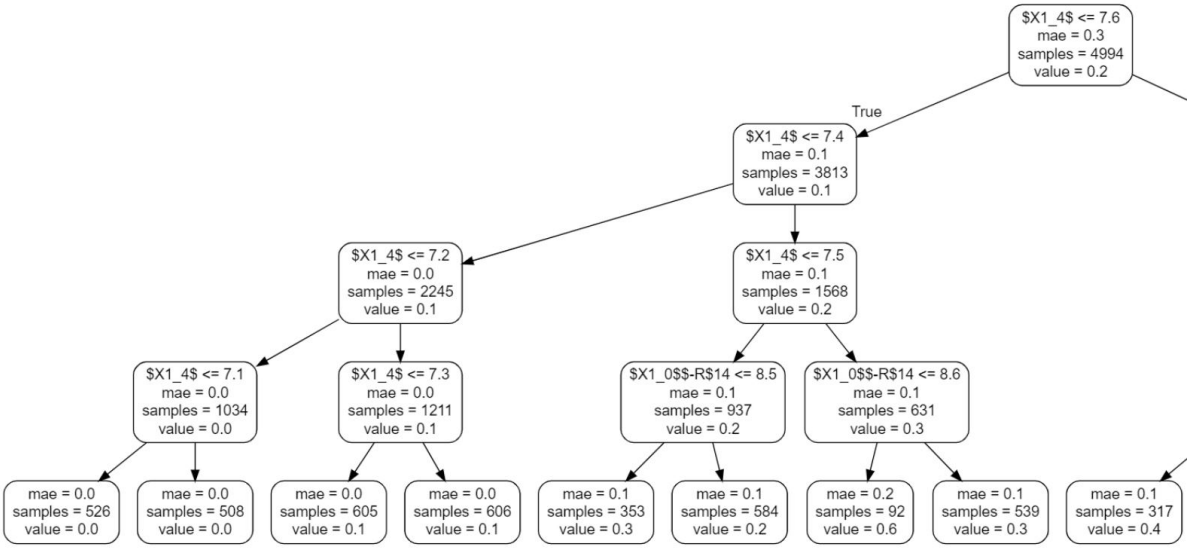


Figure G24. Left part of the regression tree of DTR model setup 5

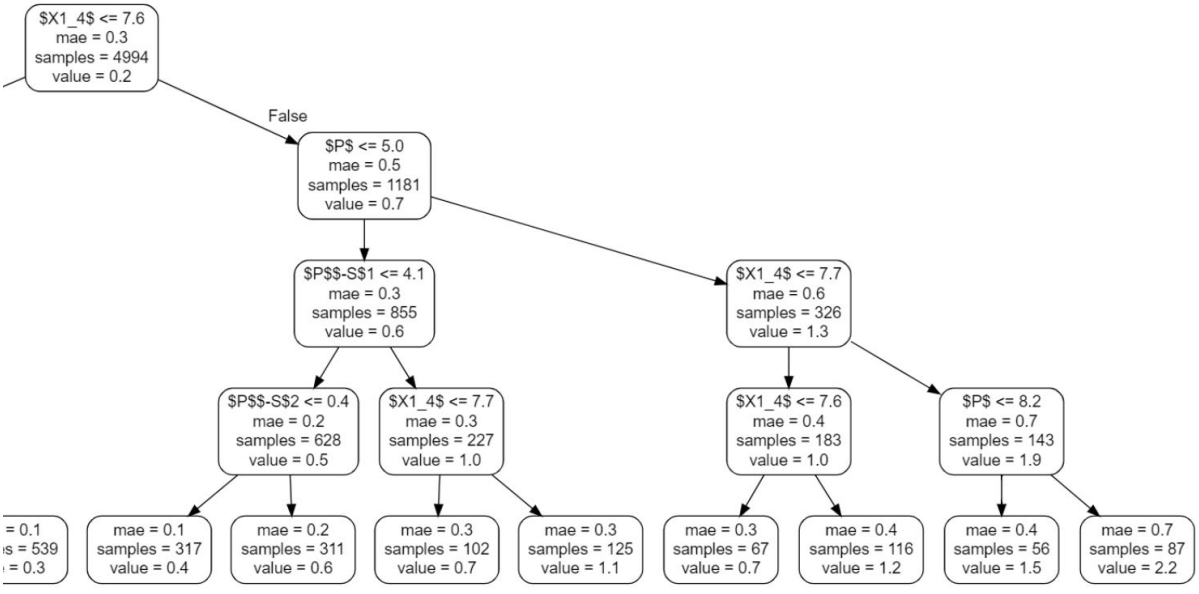


Figure G25. Right part of the regression tree of DTR model setup 5

Appendix H: Conceptual model GR4J

H1 Dataset for GR4J model

In this Appendix, the time series of the input variables and the target for this GR4J model are visualised. A division is made between the training set and the test set. Note that for hydrological models the training set is defined as calibration period, and the test set as validation period. The timeline of the calibration and validation set is similar to the timeline of the training and test set for the machine learning models.

H1.1 Input variables for GR4J model

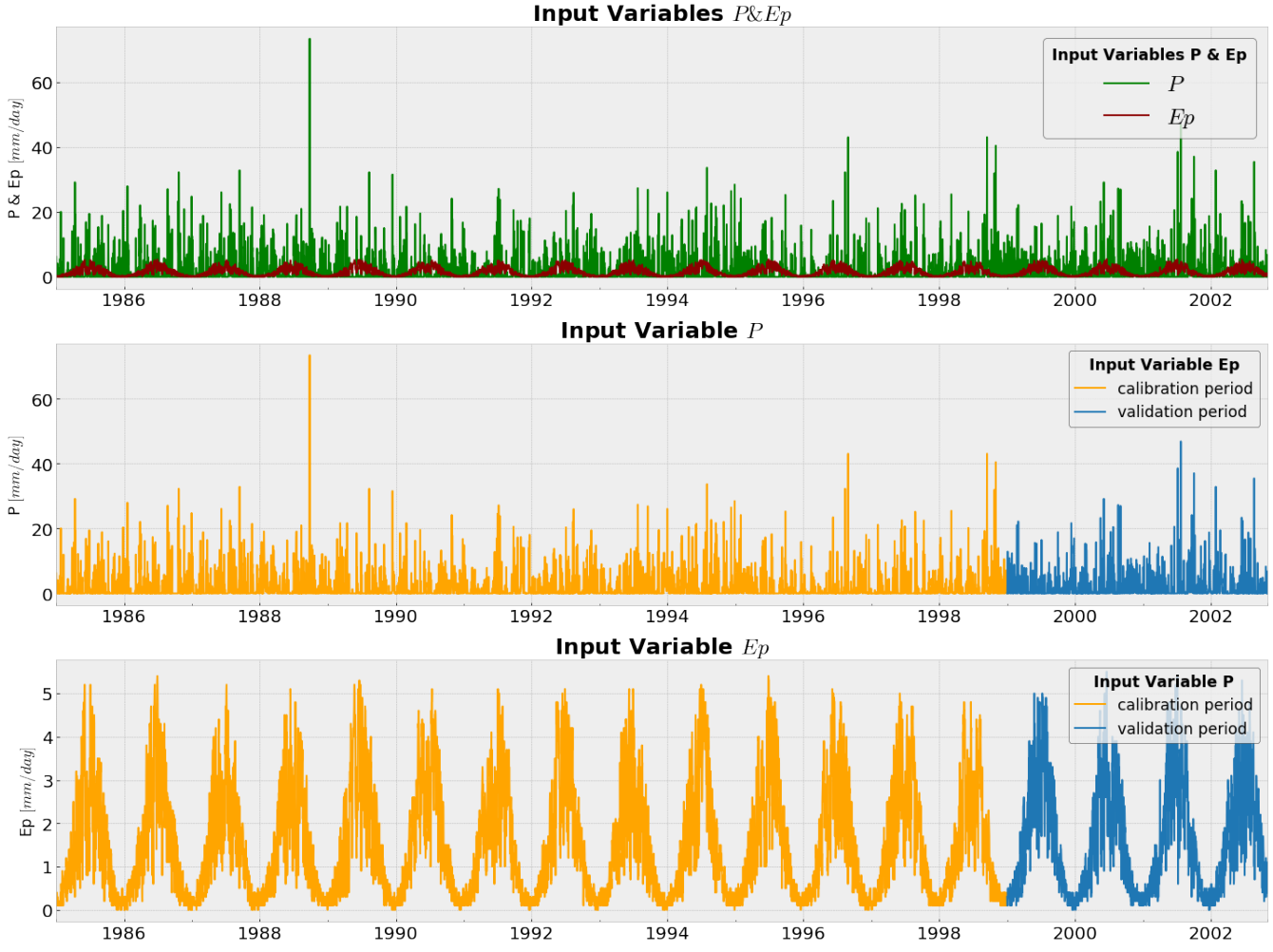


Figure H1. The time series of the input variables P and Ep for the GR4J model, divided into the calibration set (1985-1999) and the validation set (1999-2003)

H1.2 Target for GR4J model

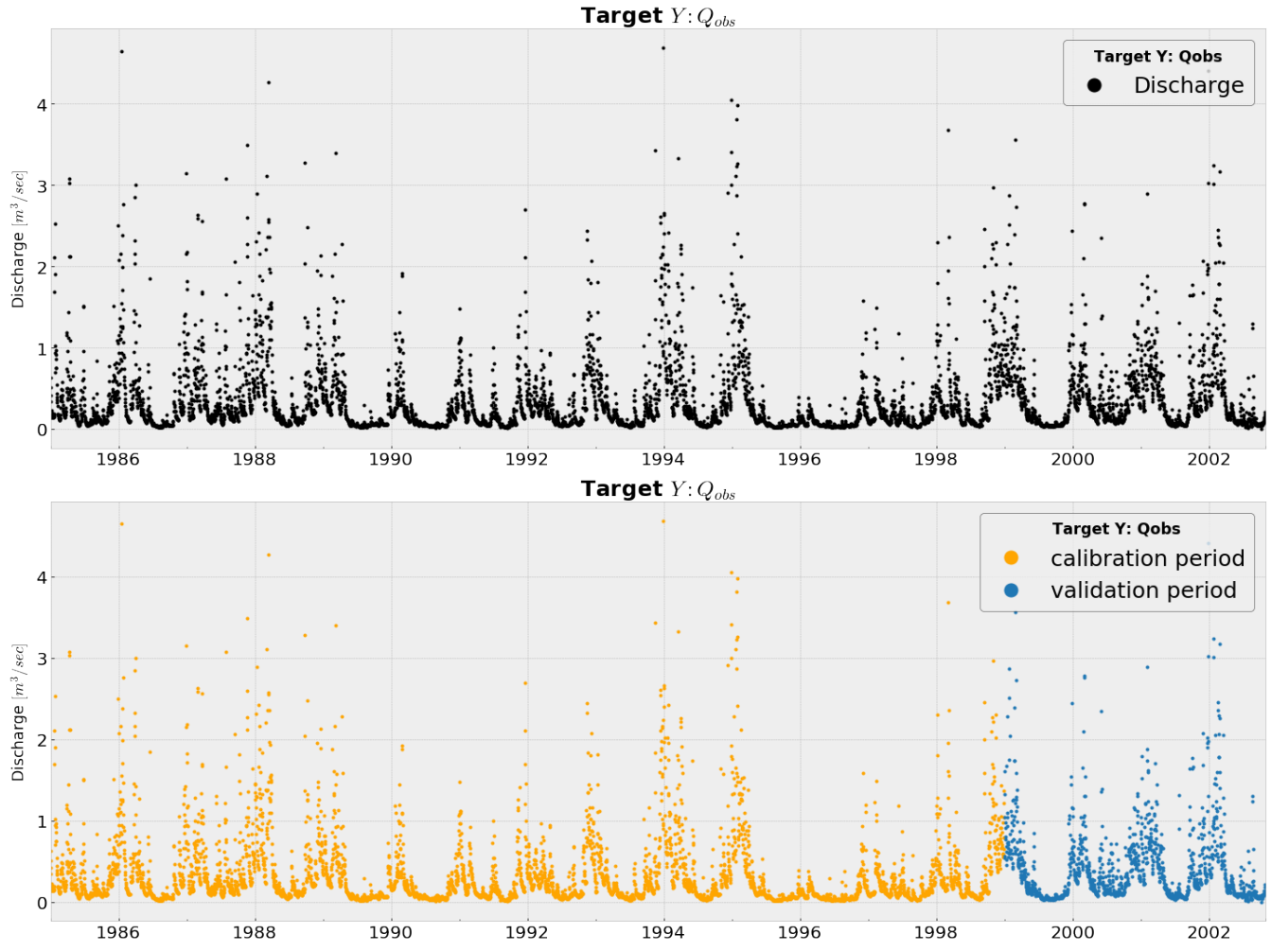


Figure H2. The time series of the target $Qobs$ for the GR4J model, divided into the calibration set (1985-1999) and the validation set (1999-2003)

H2 Results - GR4J model calibrated with different objective functions

In this Appendix, the results of the GR4J models calibrated with the objective functions NSE and MAE are separately visualised. First, the $Qsim$ time series is plotted for the calibration and validation set, followed by a plot of zooming in on only the validation set. The last figure of each GR4J model is a scatterplot of $Qobs$ against $Qsim$ to easily detect over- or underfitting.

H2.1 Results GR4J model - calibrated with objective function NSE

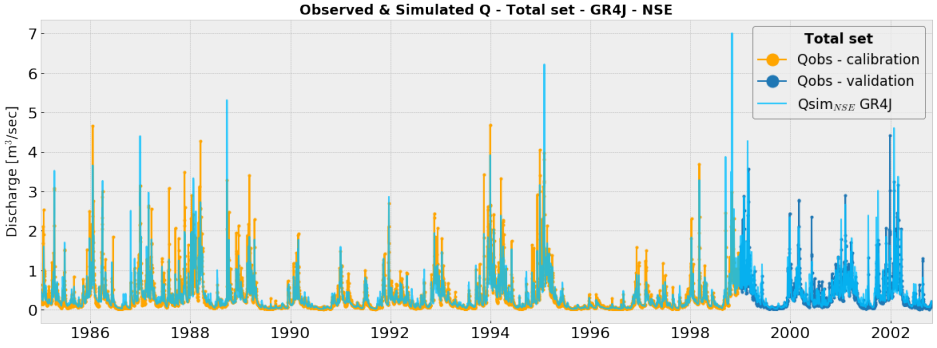


Figure H3. The time series of $Qsim$ and $Qobs$ for the calibration and validation set, for GR4J model - calibrated with objective function NSE

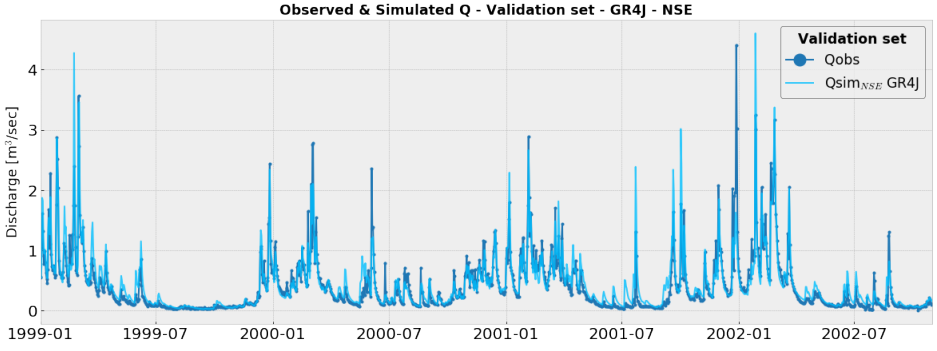


Figure H4. The time series of $Qsim$ and $Qobs$ for only the validation set, for GR4J model - calibrated with objective function NSE

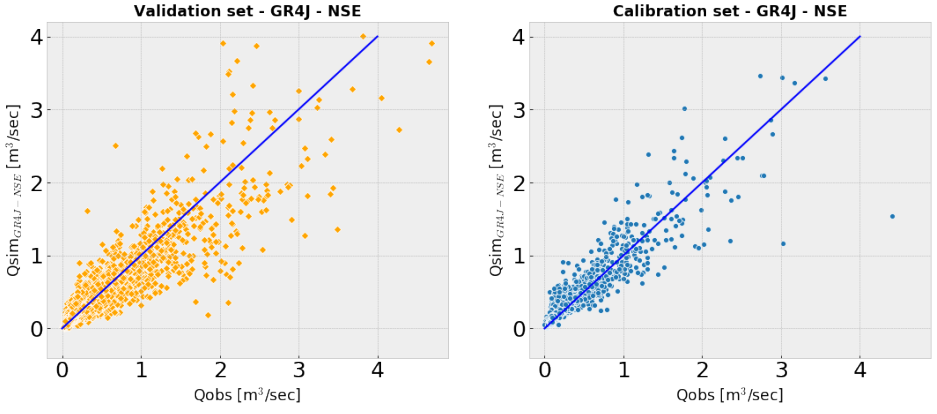


Figure H5. Scatterplots of $Qobs$ versus $Qsim$ for the calibration and validation set, for GR4J model - calibrated with objective function NSE

H2.2 Results GR4J model - calibrated with objective function MAE

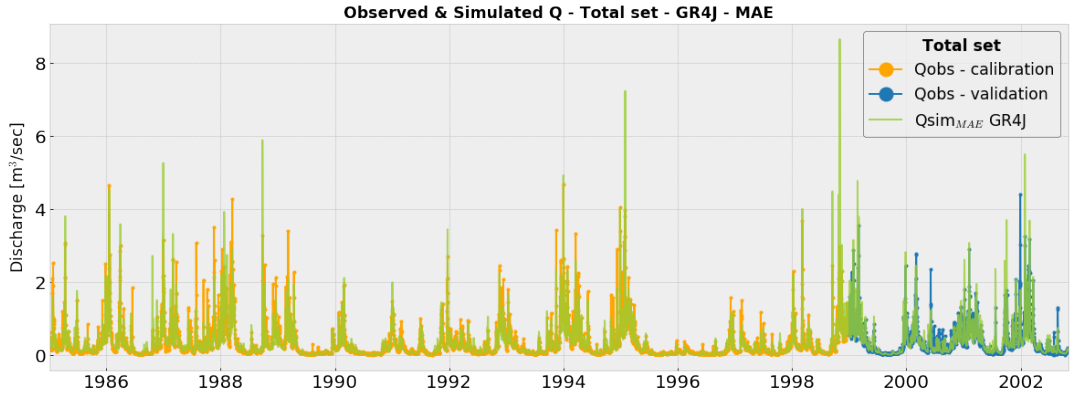


Figure H6. The time series of $Qsim$ and $Qobs$ for the calibration and validation set, for GR4J model - calibrated with objective function MAE

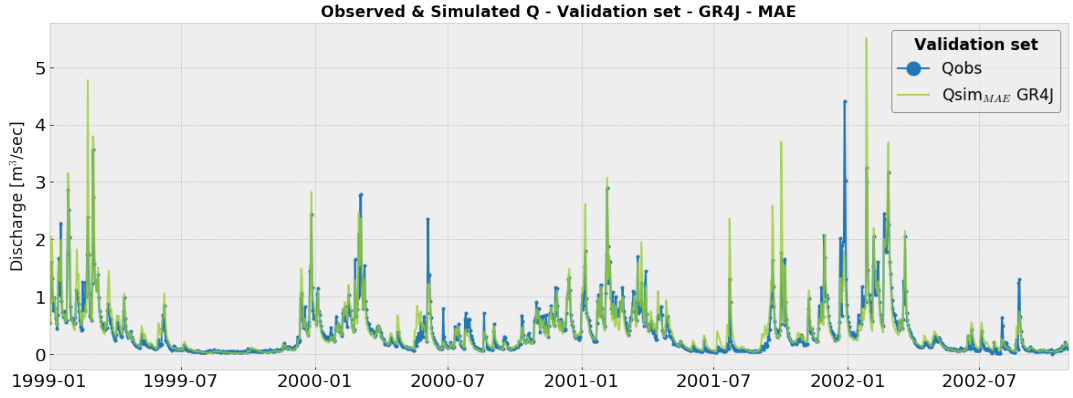


Figure H7. The time series of $Qsim$ and $Qobs$ for only the validation set, for GR4J model - calibrated with objective function MAE

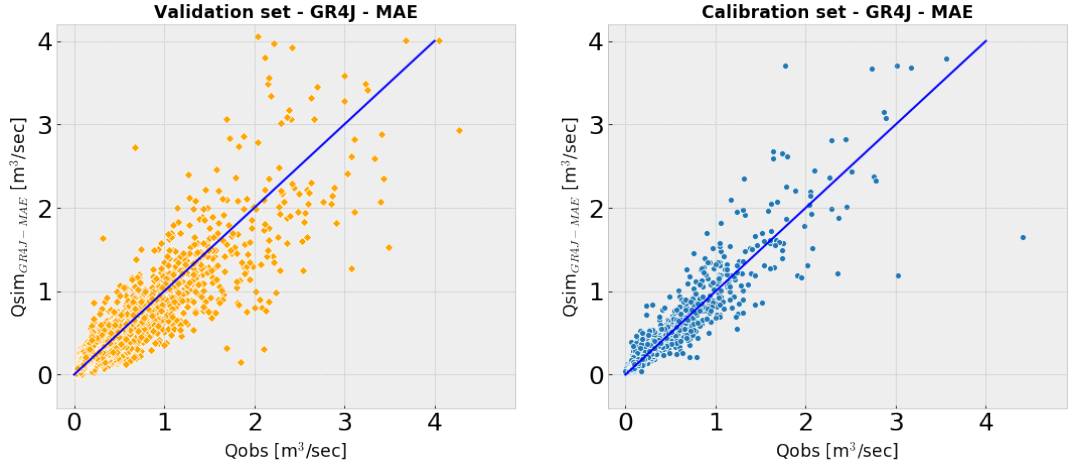


Figure H8. Scatterplots of $Qobs$ versus $Qsim$ for the calibration and validation set, for GR4J model - calibrated with objective function MAE