

Clustering as Approximation Method to Optimize Hydrological Simulations

Elnaz Azmi¹(✉)^[0000-0002-0073-8940], Uwe Ehret²^[0000-0003-3454-8755], Jörg Meyer¹^[0000-0003-0861-8481], Rik van Pruijssen²^[0000-0002-9337-2246], Achim Streit¹^[0000-0002-5065-469X], and Marcus Strobl¹^[0000-0001-8265-227X]

¹ Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany

{elnaz.azmi,joerg.meyer2,achim.streit,marcus.strobl}@kit.edu

² Institute of Water and River Basin Management, Karlsruhe Institute of Technology, Karlsruhe, Germany
{uwe.ehret,rik.pruijssen}@kit.edu

Abstract. Accurate water-related predictions and decision-making require a simulation of hydrological systems in high spatio-temporal resolution. However, the simulation of such a large-scale dynamical system is compute-intensive. One approach to circumvent this issue, is to use landscape properties to reduce model redundancies and computation complexities. In this paper, we extend this approach by applying machine learning methods to cluster functionally similar model units and by running the model only on a small yet representative subset of each cluster. Our proposed approach consists of several steps, in particular the reduction of dimensionality of the hydrological time series, application of clustering methods, choice of a cluster representative, and study of the balance between the uncertainty of the simulation output of the representative model unit and the computational effort. For this purpose, three different clustering methods namely, *K*-Means, *K*-Medoids and DBSCAN are applied to the data set. For our test application, the *K*-means clustering achieved the best trade-off between decreasing computation time and increasing simulation uncertainty.

Keywords: Clustering · Time series analysis · *K*-Means · *K*-Medoids · DBSCAN · Simulation optimization.

1 Introduction

The simulation of hydrological systems and their interactions needs an advanced modeling of water-, energy- and mass cycles in high spatio-temporal resolution [20]. This kind of modeling is used to support water-related predictions and decision making. Such a high-resolution, distributed and physically based modeling demands high performance computing (HPC) and parallel processing of the model units to function fast and efficiently [10, 13, 14]. However, parallel running of such models is challenging for domain scientists, since the interactions among the model units are not strictly independent. Either one can run the processes

parallel e.g by using a Message Passing Interface (MPI) for communication and exchange of data between processes, or one can run processes of independent model units in parallel and the processes of dependent model units sequentially. Furthermore, development, testing, execution and update of such a model on HPC Clusters involve potentially a large configuration overhead and require advanced programming expertise of domain scientists. The main aim of this work is to reduce the computational effort of the model, and in addition, to discover underlying patterns of the hydrological systems [5]. The remainder of this paper is structured as follows: Section 2 provides further information about the study background, Section 3 is a survey of related work, the proposed approach is explained in Section 4. In Section 5, the processing results are presented, Section 6 is about the implementation environment and the conclusions are drawn in Section 7.

2 Background

2.1 Hydrological Model

In this paper we apply our methods on the CAOS (Catchment as Organized Systems) model proposed by Zehe et al.[20]. This model simulates water related dynamics in the lower mesoscale catchments (few tens to few hundreds of square kilometers). The CAOS model provides a high-resolution and distributed process based simulation of water- and energy fluxes in the near surface atmosphere, the earth's surface and subsurface. These simulations are generally applicable in the field of hydrological research, agricultural water demand estimation and erosion protection or flood forecasting. The landscape is represented by model elements organized in three major hierarchy levels (Fig. 1). The smallest model elements

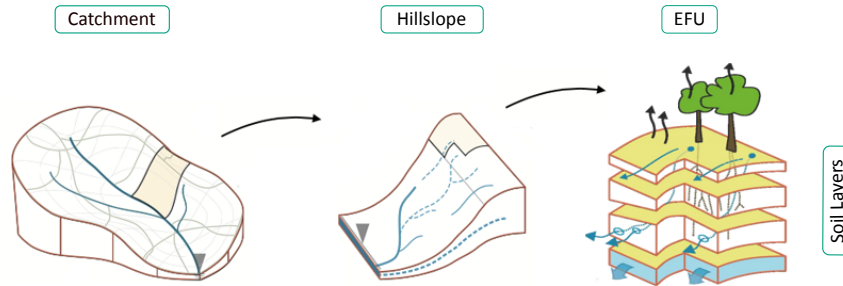


Fig. 1: Simplified hierarchy of the CAOS model units (modified after [20]).

are soil columns referred to as Elementary Functional Units (EFUs). Each EFU is composed of Soilsurface, Soil layers, Macropores (vertical cracks) and Vegetation. In an EFU, all vertical water movements (infiltration, vertical soil water flow,

and evapotranspiration) are modelled. On the second hierarchy level, Hillslope model elements contain and connect all EFUs along the downhill path from a ridge line to a river. In a Hillslope, all lateral, downhill flow processes (surface flow and groundwater flow) are modelled in network-like flow structures called rills on the surface and pipes in the subsurface (blue lines in Fig. 1, middle and right sketch). A catchment model element finally contains all Hillslopes, i.e. the drainage area up to a point of interest at a river. In a catchment, all processes of lateral water transport in a river are modelled. EFUs within the same Hillslope may interact due to backwater effects. Hillslopes act completely independent of each other. Before executing the hydrologic simulation, the catchment is divided into Hillslopes based on the flow network derived from a Digital Elevation Model (DEM). Hillslopes are then subdivided in laterally connected EFUs (Fig. 1). The hierarchy of model elements can be abstracted into a network model [5] to profit the advantages of such a representation of objects and their relationships.

2.2 Study Case

The study area used to develop and test the hydrological model is the Attert catchment in the Grand Duchy of Luxembourg. Since the computation of the hydrological model is time consuming, a representative subset of the Attert catchment, the Wollefsbach catchment, is used for the initial development (Fig. 2). To give an insight into the required simulation time, we executed the CAOS model of the Wollefsbach catchment for January 2014 in 5-minute resolution on a single core system. The properties, main structure statistics, and execution time are presented in Table 1. The simulation execution time of the whole Attert catchment has not been determined yet.

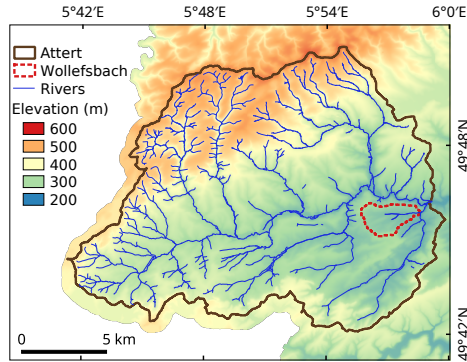


Table 1: Case study properties.

Catchment	Attert	Wollefsbach
Area	247 km ²	4.5 km ²
# Hillslopes	9716	232
Run Time	-	50.6 hours

Fig. 2: Digital Elevation Model of the Attert catchment (brown line) and the Wollefsbach catchment (current study case, red dashed line).

3 Related Work

Environmental scientists mostly use classification and clustering methods in order to detect patterns in data sets, make decisions and extract the required information by using similarity measurements [3, 15, 18, 19]. [19] studied K -means, Clara, HClust and Fuzzy clustering algorithms to analyze the uncertainty of weather situations. The proposed method reduced the RMSE of point forecasts by up to 10%. In order to predict the minimum and maximum of weather parameters like temperature and humidity, [17] compared the application of K -means and Hierarchical clustering using internal validation measures. [18] used spectral clustering to determine regions of coherent precipitation regime. They obtained spatial patterns of the precipitation regions that provide a new hydro-climatological insight to understand the hydrological systems. Furthermore, time series is one of the main input data types in environmental science [6] and dealing with these data requires additional preprocessing like dimensionality reduction and distance measurement [1].

4 Methodology

In order to speed up the simulation, we introduce a two-step approach. The first step is to apply an elementary parallelization on the independent model elements level (Hillslopes) and execute the simulation on multi-core processors [5]. The simulation has been run for the Wollefsbach catchment for the duration of one month (January, 2014). The model time resolution of the Hillslope outputs (flux drainage to River) is set to five minutes. Since the simulation code is being developed in MathWorks Matlab, for the elementary parallelization, we have used its *parfor* functionality. The average execution time of the parallel simulation with a 16-cores processor is 5.4 *hours*, which is a 9.4 times speedup in comparison to the sequential run. The second step of our method can be categorized into Model Order Reduction techniques [8] which is the focus of this paper. Such techniques aim to reduce the computational costs by dimensionality reduction and by computing an approximation of the original model. We exploit the hydrological similarity [9] to reduce the model complexity and computation efforts as a result. The underlying idea of our approach is that similar model units function similarly if departing from similar initial states and being exposed to similar forcing (rainfall or radiation). To realize that we apply clustering algorithms to cluster functionally similar model units. The studied model units are Hillslopes (the model units without exchange between individuals). Then we run the simulation only on the representative of each cluster and map the output to the other members of every cluster. The uncertainty of the approximation can be controlled by the number of clusters and the corresponding computation time.

5 Processing Results

In this work, the initial clustering which defines the initial state of the simulation model units is introduced. We use the time series of discharge from Hillslope

model elements obtained from a drainage test (Fig. 3). In the drainage test, the simulation model is executed for all of the Hillslopes initiated with full storage of water. In other words, at the beginning of the test, the Hillslopes are full of water up to their maximum capacity and drain over time. The test is applied on the Wollefsbach catchment and starts from an arbitrary time (in this case, January; Fig. 3) and lasts until the drainage of Hillslopes reaches a predefined boundary (Equilibrium). This time duration is called *Time to Equilibrium (TE)* of the Hillslopes. During the test no forcing factor is being applied. These time series are integral signatures of Hillslope size, slope, soil (Hillslope structure) and drainage properties, which we then express by two key features that are *TE* and *Active Storage (AS)*. The second feature, *AS* extracted from drainage test is the accumulated volume of water flowing out of a Hillslope at each time step normalized to the initial Total Storage of that Hillslope. The time series are used as the input data of our approach (Fig. 4).

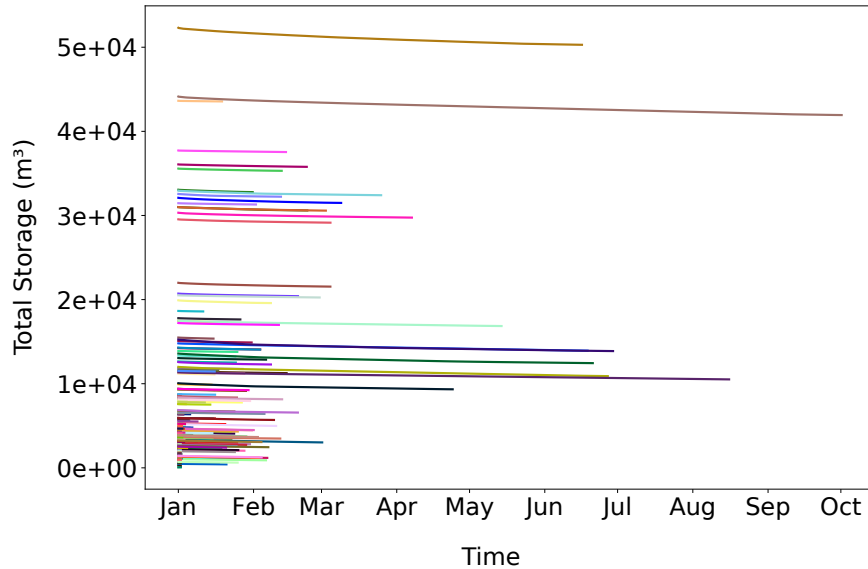


Fig. 3: Time series of Total Storage (total water volume) of Hillslopes at each time step; Each line represents a single Hillslope.

5.1 Dimensionality Reduction

Considering the input time series shown in Fig. 4, we extract the features describing their characteristics. The hydrologically meaningful features are *AS* of each Hillslope at the *TE* and the gradient of the first time step of the time series called *1st-Gradient*, because the speed of drainage especially at the first

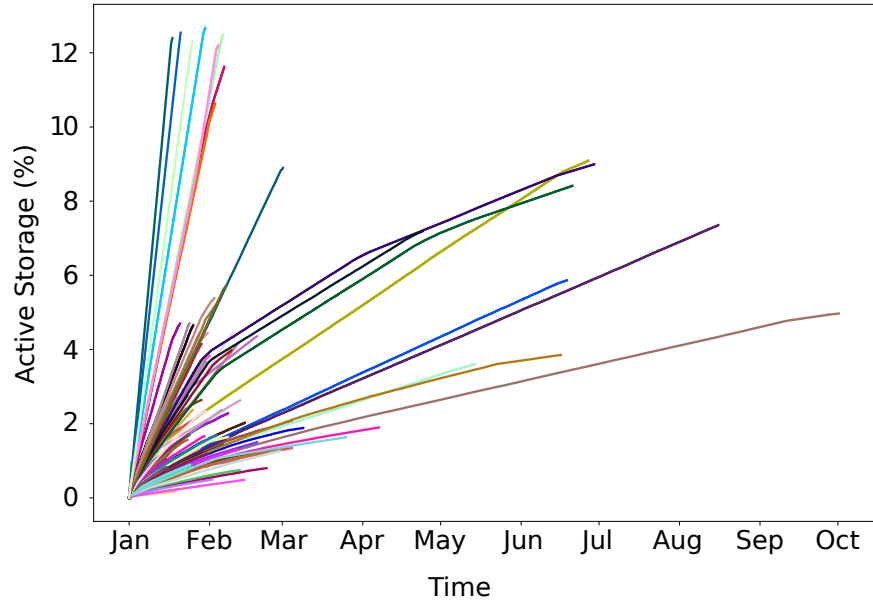


Fig. 4: Time series of AS ; Each line represents a single Hillslope.

steps of the test, characterizes the Hillslopes. Other features describing the time series are mathematical moments that express the shape of the distribution. We extracted the four moments *Mean*, *Variance*, *Skewness* and *Kurtosis*. Thus initially, we have overall a seven-dimensional feature set. Then, each feature set of all Hillslopes are normalized to standard deviation (σ) of that feature set. Dimensionality reduction is an important method to reduce computation complexity. For further dimensionality reduction, we filter highly correlated, i.e. redundant features. Principally, highly correlated features carry similar information so we can reduce them to only one feature. Here, the Pearson correlation coefficient was calculated for each pair of the extracted features (Fig. 5). The Pearson value ranges from -1 to 1 where 1 defines the total positive linear correlation, 0 is the no linear correlation and -1 is for total negative linear correlation [16]. Our extracted features are mostly non or positively correlated. Since AS and TE are our hydrological key features, we filter the features that are highly correlated with these two. Regarding Fig. 5, *Mean*, *Variance* and AS fit to our exclusion criteria. Therefore, we filter *Mean* and *Variance* from our feature set and keep AS . There is no pair of features with highly negative correlation. Finally, we have a five-dimensional feature set consisting of *Skewness*, *Kurtosis*, *1st-Gradient*, AS and TE .

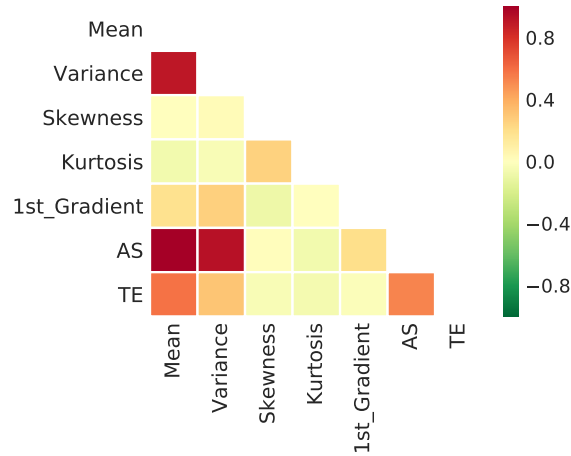


Fig. 5: Pearson correlation coefficient of the features from *AS* time series.

5.2 Clustering

Having a feature set as input data, we continue our approach with the application of popular conventional clustering methods namely *K*-means, *K*-medoid and DBSCAN and present their efficiency in our use case.

Application of *K*-Means Clustering The only required parameter for the *K*-means algorithm is the number of clusters (*K*). In order to determine *K*, there exists the conventional elbow method to define the optimal number of *K* using the total within-cluster sum of squares (WSS) or the average distance to centroid [11]. This method is useful in cases where *K* should be determined only based on the location of the points to be clustered. However, there might be additional constraints suggesting *K*. In this work, we propose an approach that considers the uncertainty of the simulation introduced by the clustering approach. There is a balance between the number of clusters, *K*, and the hydrological model uncertainty, based on the RMSE and the simulation computation time of Hillslopes. We determine *K* with a small yet representative catchment (Wollefsbach) to apply it to the bigger catchment (Attert). Thus, the *K* parameter can be selected according to the criteria of the hydrologist. Initially, we apply the *K*-means clustering with varying number of *K*. Then the RMSE is calculated within each cluster between the cluster members and the representative of that cluster. We define the cluster representative as the Medoid data point whose average dissimilarity to other points in the cluster is minimal. Formally, the Medoid of x_1, x_2, \dots, x_n as members of each cluster is defined as [12]:

$$x_{\text{medoid}} = \operatorname{argmin}_{y \in \{x_1, x_2, \dots, x_n\}} \sum_{i=1}^n d(y, x_i), \quad (1)$$

where $d(y, x_i)$ is the distance function between y and the i st x . RMSE is the standard deviation of the prediction errors. Formally, RMSE is [7]:

$$\text{RMSE} = \sqrt{\sum_{i=1}^N (z_{f_i} - z_{o_i})^2 / N}, \quad (2)$$

where N is the sample size, z_{f_i} are the predicted values and z_{o_i} are the observed values. According to this, the RMSE measure was calculated between the AS time series of the cluster members and the representative of that cluster. Thus, there is one RMSE measurement per Hillslope for each K variation. Finally, the total RMSE measure of all Hillslope is calculated and plotted in Fig. 6 using the following equation:

$$\sigma_{\text{totalRMSE}} = \sqrt{\sum_{i=1}^P (\text{RMSE}_i)^2}, \quad (3)$$

where P is the number of data points in the feature set. In order to find the op-

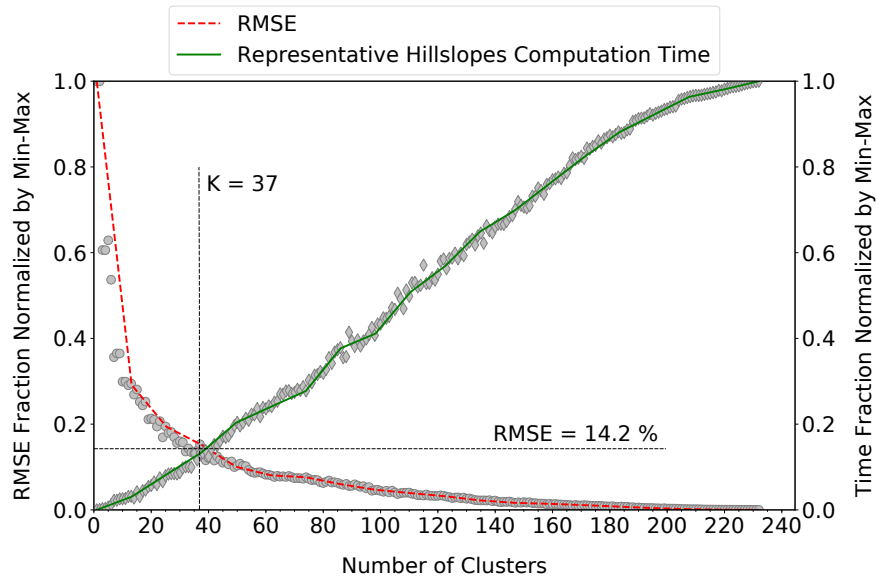


Fig. 6: RMSE and representative Hillslopes computation time for varying K s using K -means clustering. The gray markers show the original values and the curves in red and green represent their smoothed trend.

timal number of K , we use the trade-off between the RMSE measurement and sum of the computation time of representative Hillslopes of each cluster. According to our methodology, the simulation is applied only on the representative

Hillslopes and sum of their computation time is calculated for each number of K . The results of this experiment are shown in Fig. 6, where the horizontal axis represents the number of clusters and the vertical axis, the RMSE measurement and sum of the computation time of representative Hillslopes of each cluster normalized by Min-Max normalization. Evidently, as the number of clusters raises, the corresponding RMSE decreases while the computation time increases (Fig. 6). The main goal of our approach is to achieve the best trade-off between computation time and simulation uncertainty. In Fig. 6, a range of the intended compromise between RMSE and computation time is recognizable where the curves intersect. As K -means places the initial centroids randomly, the output of its executions with the same number of K differs slightly. Thus, the intended compromise occurs where $32 < K < 42$, $11.8\% < RMSE < 14.2\%$ of the maximum $RMSE = 39.2$ and the computation time ranges from 10.3% to 16.2% of total computation time (31.8 days). As an example, the spatial distribution of the K -means clustering at $K = 37$ which corresponds to the best compromise between RMSE and computation time in Wollefsbach catchment is shown in Fig. 7. Each color indicates a cluster and the number of its members can be found in the legend of map. All the single member clusters are shown in blue, which are single Hillslopes that do not fit into the other clusters. The map shows a valid Hillslopes clustering, considering the hydrological parameters like the structure, size and location of the Hillslopes. Generally, the overhead of running such a clustering during the simulation is negligible.

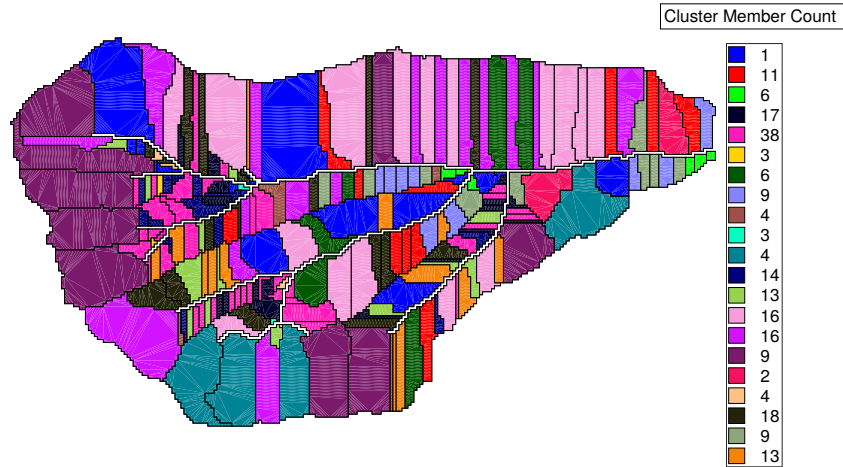


Fig. 7: Spatial distribution of K -means clusters at $K = 37$ applied on Wollefsbach catchment. All single member clusters are shown in dark blue.

Application of K -Medoids Clustering Another variant of K -means is the K -medoids algorithm that uses the actual data points as cluster centers. It receives the number of clusters (K) and the distance matrix of points as input parameters. We have used the K -medoid source code available at [2]. The algorithm was run for variable number of K and the results are shown in Fig. 8. The plot indicates that the intended compromise range between RMSE and computation time occurs where $58 < K < 78$, $16.8\% < RMSE < 34.7\%$ of the maximum $RMSE = 31.8$ and the related computation time is between 22.7% and 33.8% of the maximum computation time (31.8 days).

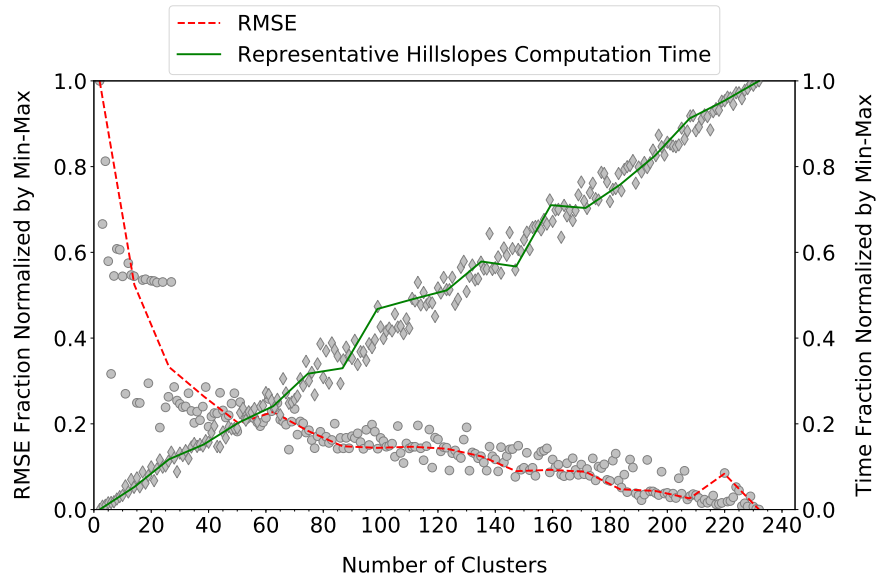


Fig. 8: RMSE and representative Hillslopes computation time for varying K 's using K -medoids clustering. The gray markers show the original values and the curves in red and green represent the smoothed trend.

Application of DBSCAN Clustering DBSCAN clustering requires two main parameters as input, namely Eps and $MinPts$. In order to find a set of optimal parameters, DBSCAN clustering is applied on a different range of Eps and $MinPts$. The same method of determining and visualizing RMSE with the computation time described in Section 5.2 is used with DBSCAN clustering. For each set of parameters, the number of clusters is calculated. Noise clusters are considered as one cluster in the whole number of clusters. The results shown in Fig. 9 indicate that the intended compromise range between RMSE and computation time is achieved where the number of clusters ranges between 51 – 62,

$0.3 < Eps < 0.7$, $1 < MinPts < 21$, the RMSE is between 14.5% and 31.4% of maximum RMSE (38.6) and the computation time is in range of 17.9% and 23% of the maximum computation time (31.8 days). The direct comparison of

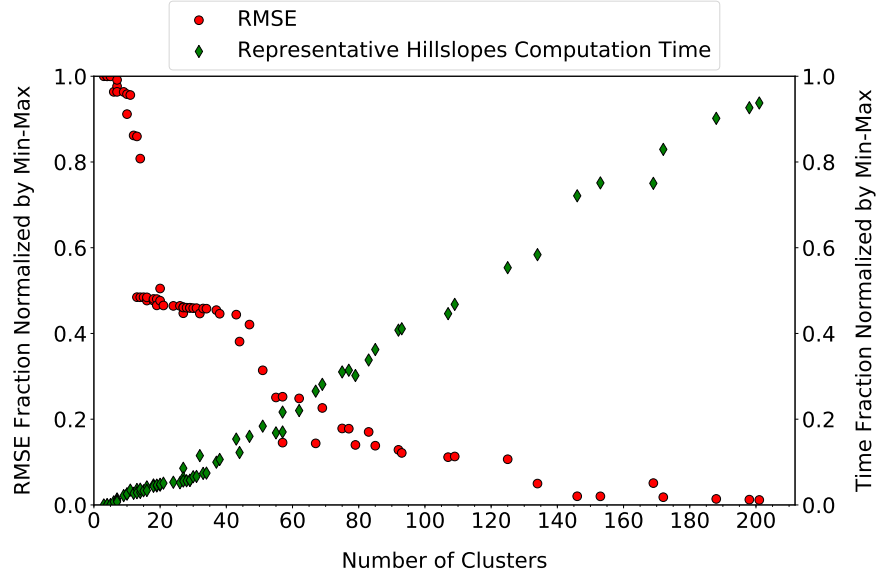


Fig. 9: RMSE and representative Hillslopes computation time for variable Eps and $MinPts$ using DBSCAN clustering. Some of the DBSCAN parameters' combination generate the same number of clusters.

the three applied methods is illustrated in Fig 10, which clearly shows that the K -means clustering performs better for the studied case and features the lowest RMSE for up to 18 days of computation. A summary of all results are available in Table 2.

6 Implementation Environment

All the analysis methods are implemented in Python and executed on a computer with Ubuntu 16.04.4 LTS operating system running the Linux kernel 4.4.0-127-generic and a four-core 64-bit Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz processor. The benchmarking of simulation model parallelization has been done on a computer with Red Hat Enterprise Linux Server release 7.4 running the linux kernel 3.10.0-693.11.6.el7.x86_64 and a 16-core Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz processor. All scripts, data files and requirements of the analyses are available as a gitlab repository named “hyda” [4].

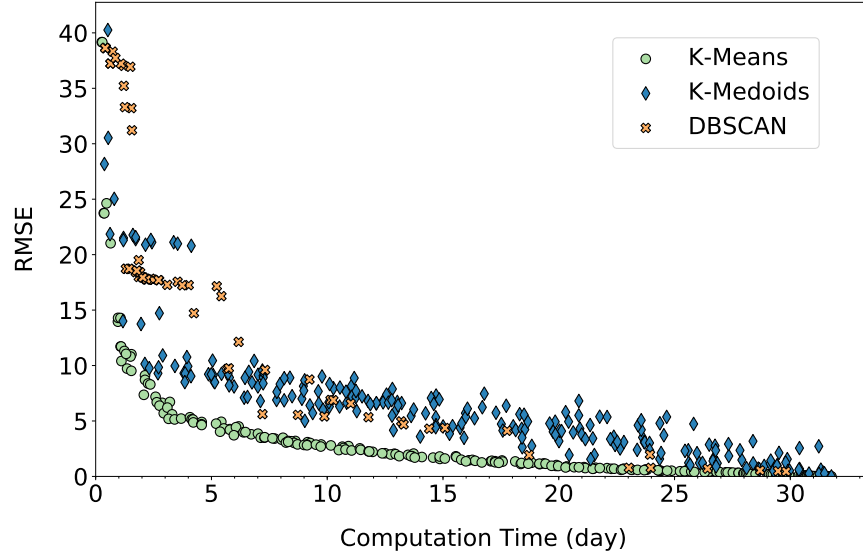


Fig. 10: Comparison of the RMSE and computation time of all analyses.

Table 2: Parameters and achievements of different clustering methods

Parameters	K-Means	K-Medoids	DBSCAN
K (# clusters)	32 - 42	58 - 78	51 - 62
Eps	-	-	0.3 - 0.7
MinPts	-	-	1 - 21
RMSE (%)	11.8 - 14.2	16.8 - 34.7	14.5 - 31.4
Max RMSE	39.2	31.8	38.6
Computation Time (%)	10.3 - 16.2	22.7 - 33.8	17.9 - 23
Max Computation Time (d)	31.8	31.8	31.8

7 Conclusions and Future Work

In this work we introduced an approach to make use of landscape properties to reduce computational redundancies in hydrological model simulations. We applied three different clustering methods namely, *K*-Means, *K*-Medoids and

DBSCAN on the time series data from a study case in hydrology. According to the results, the K -means clustering functions better than the other applied clustering methods. It achieves the intended compromise between RMSE and Hillslopes computation time in a range of $11.8\% < RMSE < 14.2\%$ and $10.3\% < computation\ time < 16.2\%$. The K -means clustering requires a smaller number of clusters and consequently lower representative Hillslopes computation time in comparison to the other studied clustering methods. Considering the $16.8\% < RMSE < 34.7\%$ and $22.7\% < computation\ time < 33.8\%$, K -medoids clustering shows worse performance than the other two methods. DBSCAN clustering has promising results also not pleasing as the K -means method. The main challenge of applying DBSCAN is to find an intended balance of both Eps and $MinPts$ parameters. As a future work, the methods will be applied on the whole Attert catchment simulations and as a forward step the clustering approach will be extended to consider also forcing in the simulation model.

References

1. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering—a decade review. *Inf. Syst.* pp. 16–38 (2015). <https://doi.org/10.1016/j.is.2015.04.007>
2. Alspaugh, S.: k -medoids clustering (May 2018), https://github.com/salspaugh/machine_learning/blob/master/clustering/kmedoids.py
3. Arroyo, Á., Tricio, V., Corchado, E., Herrero, Á.: A comparison of clustering techniques for meteorological analysis. In: 10th International Conference on Soft Computing Models in Industrial and Environmental Applications. pp. 117–130 (2015)
4. Azmi, E.: Hydrological data analysis (August 2018), <https://gitlab.com/elnazazmi/hyda>
5. Azmi, E.: On using clustering for the optimization of hydrological simulations. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 1495–1496 (2018). <https://doi.org/10.1109/ICDMW.2018.00215>
6. Bărbulescu, A.: Studies on Time Series Applications in Environmental Sciences, vol. 103 (2016)
7. Barnston, A.G.: Correspondence among the correlation, rmse, and heidke forecast verification measures; refinement of the heidke score. *Weather Forecast.* pp. 699–709 (1992)
8. Benner, P., Faßbender, H.: Model order reduction: Techniques and tools. *Encyclopedia of Systems and Control* pp. 1–10 (2013)
9. Ehret, U., Zehe, E., Scherer, U., Westhoff, M.: Dynamical grouping and representative computation: a new approach to reduce computational efforts in distributed, physically based modeling on the lower mesoscale. presented at the AGU Chapman conference, 23–26 September, 2014 (Abstract 2093) (2014)
10. Jones, J.E., Woodward, C.S.: Newton–krylov–multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. *Adv Water Resour* pp. 763–774 (2001). [https://doi.org/10.1016/S0309-1708\(00\)00075-0](https://doi.org/10.1016/S0309-1708(00)00075-0)
11. Kassambara, A.: Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning, vol. 1 (2017)
12. Kaufman, L., Rousseeuw, P.: Clustering by means of Medoids. In *Statistical Data Analysis Based on the L1Norm and Related Methods* (1987)

13. Kollet, S.J., Maxwell, R.M., Woodward, C.S., Smith, S., Vanderborght, J., Vereecken, H., Simmer, C.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. *Water Resour. Res.* (4) (2010). <https://doi.org/10.1029/2009WR008730>
14. Maxwell, R., Condon, L., Kollet, S.: A high-resolution simulation of groundwater and surface water over most of the continental us with the integrated hydrologic model parflow v3. *Geosci. Model Dev* p. 923 (2015). <https://doi.org/10.5194/gmd-8-923-2015>
15. Netzel, P., Stepinski, T.: On using a clustering approach for global climate classification. *J. Clim* pp. 3387–3401 (2016). <https://doi.org/10.1175/JCLI-D-15-0640.1>
16. Pearson, K.: VII. mathematical contributions to the theory of evolution.iii. regression, heredity, and panmixia. *Philos. Trans. Royal Soc. A* pp. 253–318 (1896). <https://doi.org/10.1098/rsta.1896.0007>
17. Shobha, N., Asha, T.: Monitoring weather based meteorological data: Clustering approach for analysis. In: 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). pp. 75–81 (2017). <https://doi.org/10.1109/ICIMIA.2017.7975575>
18. Türkeş, M., Taht, H.: Use of the spectral clustering to determine coherent precipitation regions in turkey for the period 1929–2007. *Int. J. Climatol* pp. 2055–2067 (2011). <https://doi.org/10.1002/joc.2212>
19. Zarnani, A., Musilek, P., Heckenbergerova, J.: Clustering numerical weather forecasts to obtain statistical prediction intervals. *Meteorol. Appl.* pp. 605–618 (2014). <https://doi.org/10.1002/met.1383>
20. Zehe, E., Ehret, U., Pfister, L., Blume, T., Schroeder, B., Westhoff, M., Jackisch, C., Schymanski, S.J., Weiler, M., Schulz, K., et al.: Hess opinions: From response units to functional units: a thermodynamic reinterpretation of the hru concept to link spatial organization and functioning of intermediate scale catchments. *Hydrol. Earth Syst. Sci.* pp. 4635–4655 (2014). <https://doi.org/10.5194/hess-18-4635-2014>