Supplement of Nat. Hazards Earth Syst. Sci., 23, 809–822, 2023
https://doi.org/10.5194/nhess-23-809-2023-supplement

Natural Hazards and Earth System Sciences
Open Access
EGU

Supplement of

Transferability of data-driven models to predict urban pluvial flood water depth in Berlin, Germany

Omar Seleem et al.

Correspondence to: Omar Seleem (seleem@uni-potsdam.de)

Hyperparameter tuning

- **Convolutional neural network (U-Net)**

Table S1, Table S2 and

Table S3 show the computed performance indices for different combinations of U-Net hyper-parameter (depth, number of filters in the first encoder block, and filter size) combinations using the best performance training data combination (SA1 & 2). We considered a total of 18 hyper-parameter combinations and selected the best combination based on the model performance in the training domain. Table S1 and Table S2 show that the best performance model for both SA1 and SA2 (study areas included in the training domain) had a number of filters =32, filter size = 3, and depth = 4. It has 7,771,809 parameters and a training time of 8 hours 21 minutes.

- **Random forest model (RF)**

We trained RF models with different numbers of trees in the forest also using the best performance training data combination (SA1 & 2), increasing the number of trees in the forest increased the computational cost and the training time (from 10 minutes to 3 hours) but it had no significant performance gain on the model prediction. Our findings agree with previous studies (Oshiro et al., 2012; Zahura et al., 2020). Therefore, we used the number of trees in the forest = 10 for further predictions in this paper.

The holdout validation method showed that the RF models overfitted the training dataset. Therefore, we carried out a hyper-parameter tuning (number of iterations = 30) for the best performing model (RF – SA1&2) with k-fold cross-validation and a smaller training dataset (number of samples =100,000) as shown in Table S4. Then, we trained a model with the best hyper-parameters combination on the entire dataset. The trained model performance within the training domain reduced slightly (Nash Sutcliffe Efficiency (NSE) values dropped from 0.84 and 0.9 to 0.75 and 0.84 for SA1 and SA2 respectively). Outside the training domain (the NSE increased from -0.67 to 0.03 for SA0). The calculated performance indices showed that the model trained using the best hyperparameter combination and k-fold cross-validation still could not generalize outside the training domain.

We investigated also the impact of limiting the depth of decision trees on the holdout validation method. We implemented several models (RF- SA1&2) with varying the depth of the decision tree. Figure S1 shows the computed NSE values for all the implemented models. It points out that reducing the maximum depth of the decision tree enhanced the model performance outside the training domain at the cost of reducing the model performance inside the training domain.

The goal of our study is to assess the model transferability in space. To do this, we selected the best combination of hyperparameters for each model based on their performance on the validation dataset. We then evaluated the models' performance on the testing dataset. We did not specifically choose the hyperparameter combination to optimize the models' transferability

to other regions; Table S1 shows that a CNN model with 16 filters, a filter size of 7, and a network depth of 3 was superior for predictions outside of the training domain. However, other models were superior inside the training domain.

Table S1. Calculated performance indices for SA0 for all the computed hyperparameter combinations for the best training dataset combination (SA1 & 2)

	Filter_num	Filter_size	Depth	NSE	RMSE	CSI
0	16	3	3	0.216079	0.144989	0.534608
1	16	3	4	0.450139	0.121430	0.506848
2	16	5	3	0.405210	0.126293	0.528408
3	16	5	4	0.465831	0.119684	0.539143
4	16	7	3	0.602343	0.103265	0.542547
5	16	7	4	0.429672	0.123669	0.491252
6	32	3	3	0.079667	0.157098	0.518367
7	32	3	4	0.531396	0.112099	0.555347
8	32	5	3	0.533752	0.111817	0.538400
9	32	5	4	0.431954	0.123421	0.512028
10	32	7	3	0.141585	0.151721	0.527111
11	32	7	4	-0.068188	0.169247	0.503513
12	64	3	3	0.530566	0.112198	0.511400
13	64	3	4	0.349439	0.132082	0.536829
14	64	5	3	0.415645	0.125180	0.525844
15	64	5	4	0.387801	0.128128	0.531205
16	64	7	3	0.450000	0.121345	0.510000
17	64	7	4	0.470000	0.121345	0.500000

Table S2. Calculated performance indices for SA1 for all the computed hyperparameter combinations for the best training dataset combination (SA1 & 2)

	Filter_num	Filter_size	Depth	NSE	RMSE	CSI
0	16	3	3	0.738003	0.070887	0.593455
1	16	3	4	0.715503	0.073868	0.537106
2	16	5	3	0.765911	0.067005	0.575877
3	16	5	4	0.724726	0.072661	0.575724
4	16	7	3	0.633788	0.083808	0.459598
5	16	7	4	0.772377	0.066073	0.561517
6	32	3	3	0.794891	0.062720	0.618499
7	32	3	4	0.837990	0.055743	0.651536
8	32	5	3	0.729301	0.072054	0.527954
9	32	5	4	0.703284	0.075438	0.479789
10	32	7	3	0.810939	0.060217	0.595246
11	32	7	4	0.779307	0.065060	0.590154
12	64	3	3	0.695888	0.076372	0.548835
13	64	3	4	0.760771	0.067737	0.604587
14	64	5	3	0.759953	0.067853	0.587598
15	64	5	4	0.833772	0.056464	0.605823
16	64	7	3	0.669744	0.079587	0.515753
17	64	7	4	0.740098	0.070604	0.558408

Table S3. Calculated performance indices for SA2 for all the computed hyperparameter combinations for the best training dataset combination (SA1 & 2)

	Filter_num	Filter_size	Depth	NSE	RMSE	CSI
0	16	3	3	0.728378	0.086787	0.551695
1	16	3	4	0.683640	0.093662	0.491272
2	16	5	3	0.775044	0.078981	0.551749
3	16	5	4	0.745083	0.084076	0.537122
4	16	7	3	0.565798	0.109728	0.377808
5	16	7	4	0.795849	0.075240	0.548188
6	32	3	3	0.853574	0.063721	0.611738
7	32	3	4	0.859656	0.062384	0.620996
8	32	5	3	0.737983	0.085239	0.492455
9	32	5	4	0.747101	0.083743	0.462011
10	32	7	3	0.841518	0.066292	0.592339
11	32	7	4	0.810015	0.072583	0.594880
12	64	3	3	0.665841	0.096267	0.479617
13	64	3	4	0.767345	0.080326	0.579144
14	64	5	3	0.770667	0.079751	0.542615
15	64	5	4	0.856114	0.063170	0.584332
16	64	7	3	0.685174	0.093441	0.489645
17	64	7	4	0.729622	0.086594	0.474623

Table S4. Selection of best combination of parameters for the RF model based on hyperparameter tuning using K-fold cross-validation method.

Parameter	Range	Best combination
Number of trees in random forest	100 < n_estimator< 2000	2000
Number of features to consider at every split	[auto , sqrt]	auto
Maximum number of levels in tree	10 < max_depth < ∞	50
Minimum number of samples required to split a node	min_samples_split = [2, 5, 10]	5

Minimum number of samples required at each leaf node	min_samples_leaf = [1, 2, 4]	2
Method of selecting samples for training each tree	bootstrap = [True, False]	True

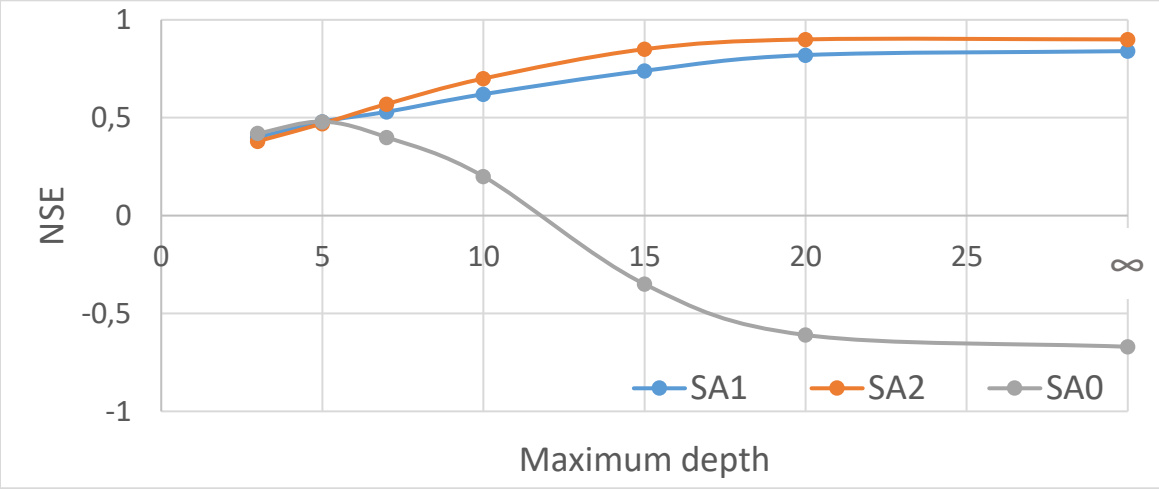

Figure S1. Shows the impact of varying the maximum depth of the decision trees on the RF – SA1 &2 model performance using holdout validation method. The X-axis shows the implemented maximum depth of the decision trees while the Y-axis denotes the computed Nash Sutcliffe Efficiency (NSE)

Flood maps:

Figure S2 and Figure S3 compare the water depths from different models and TELEMAC-2D model for 50 and 140 mm precipitation events for SA0. The figures show that the models performance enhances with increasing the precipitation depth.

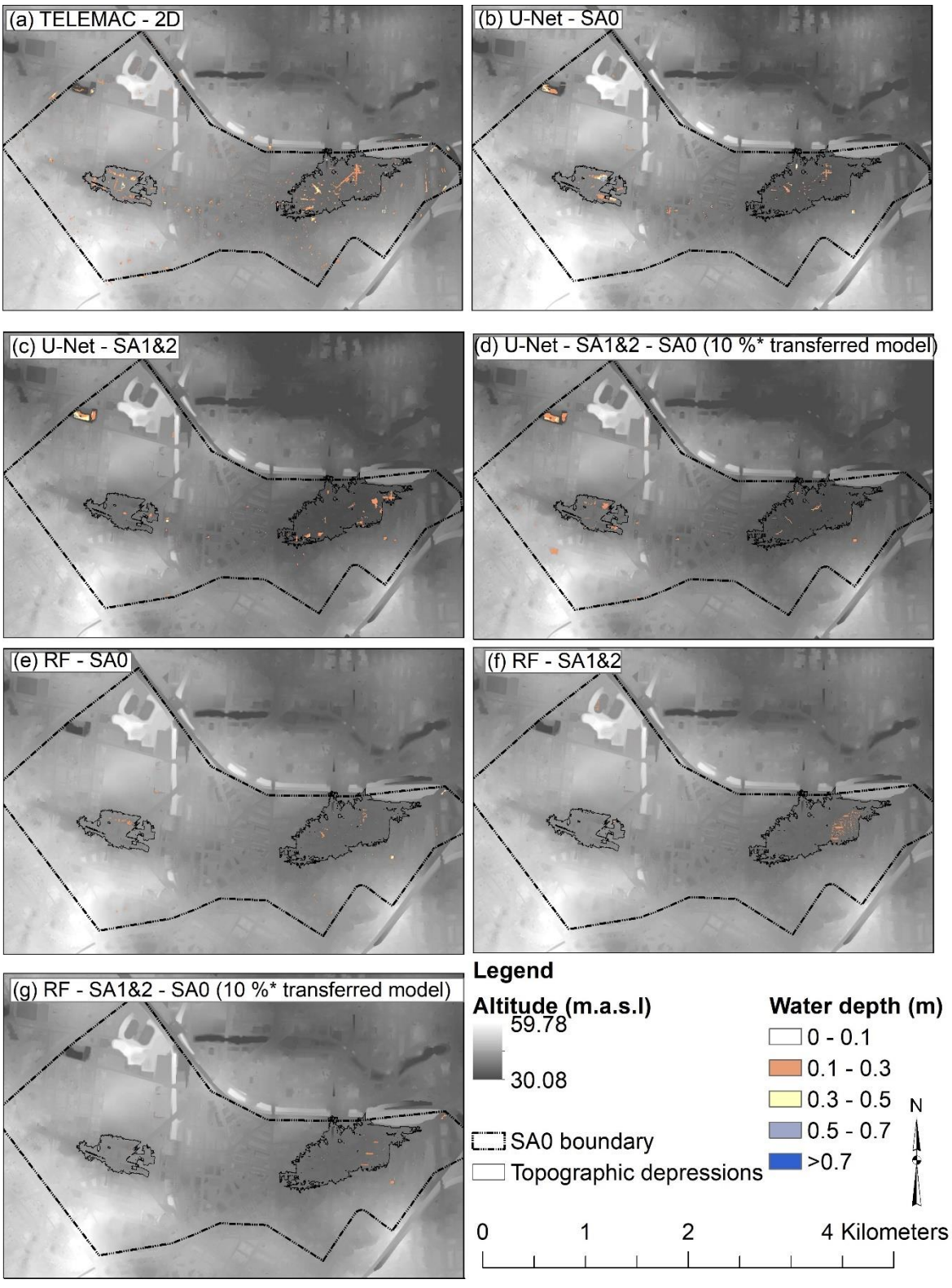

Figure S2 Comparison of water depths from different models and TELEMAC-2D model for a 50 mm precipitation event for SA0. The figure highlights the boundary of two topographic depressions within SA0 where runoff accumulates. The altitude is shown in the background

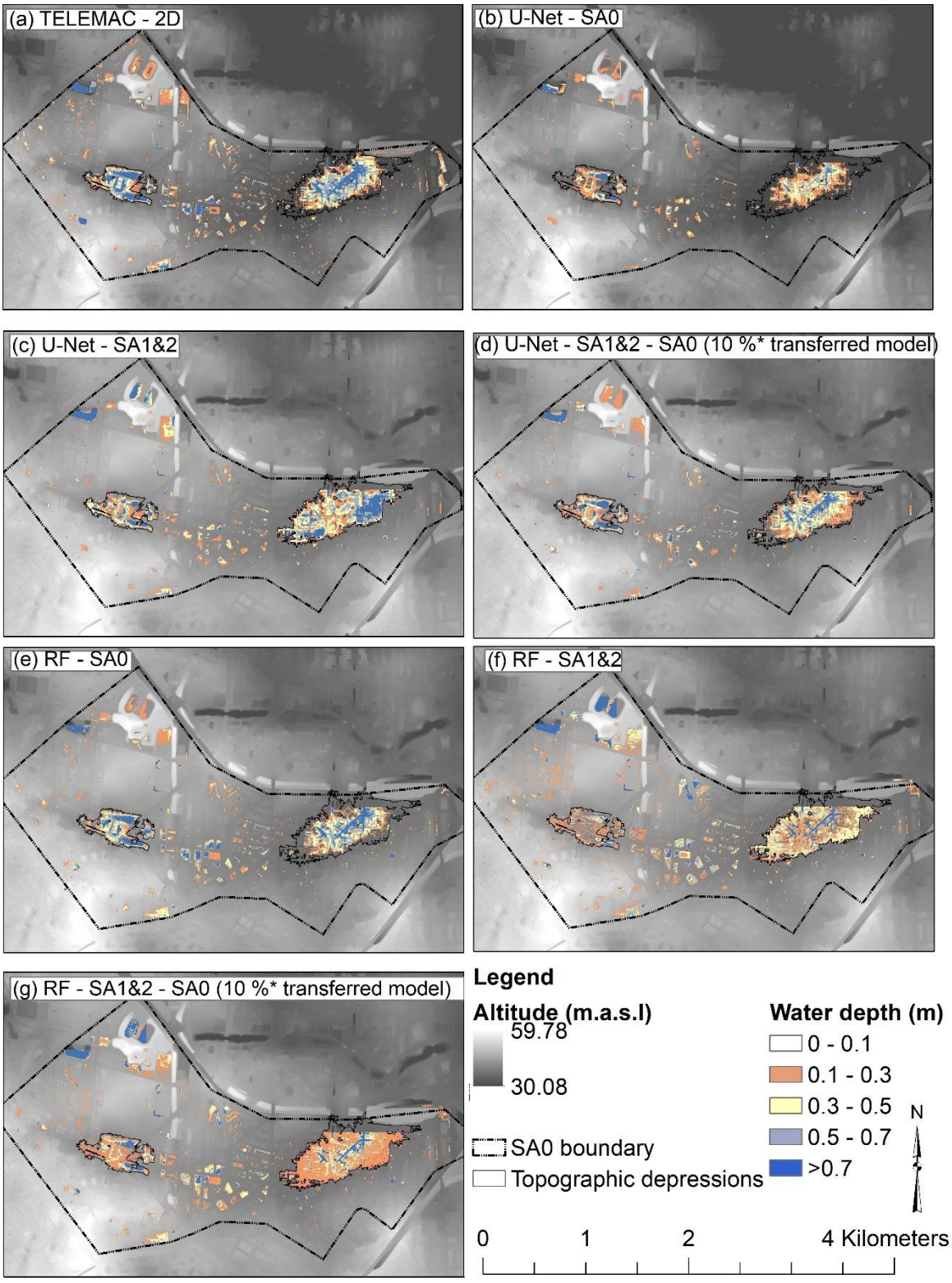

Figure S3 Comparison of water depths from different models and TELEMAC-2D model for a 140 mm precipitation event for SA0. The figure highlights the boundary of two topographic depressions within SA0 where runoff accumulates. The altitude is shown in the background

References

Oshiro, T. M., Perez, P. S., and Baranauskas, J. A.: How many trees in a random forest?, in: International workshop on machine learning and data mining in pattern recognition, pp. 154–168, Springer, 2012

Zahura, F. T., Goodall, J. L., Sadler, J. M., Shen, Y., Morsy, M. M., and Behl, M.: Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community, Water Resources Research, 56, e2019WR027 038, 2020.