Supplement of Geosci. Model Dev., 18, 921–937, 2025
https://doi.org/10.5194/gmd-18-921-2025-supplement

Supplement of

Advances in land surface forecasting: a comparison of LSTM, gradient boosting, and feed-forward neural networks as prognostic state emulators in a case study with ecLand

Marieke Wesselkamp et al.

Correspondence to: Marieke Wesselkamp (marieke.wesselkamp@biom.uni-freiburg.de)

S1 Data base

The data bases contain the variables described in the main file, in addition with global statistics for all prognostic state variables. These are spatiotemporal minimum, maximum, mean and standard deviation. Data can be requested for the European scale under: https://doi.org/10.21957/n17n-6a68, and for the Global scale under: https://doi.org/10.21957/pcf3- ah06.

S1.1 European subregions for horizons computation

A northern European subset was selected on Southern Scandinavia with a grid box on minimum and maximum latitudes of 55 and 71 degree respectively and a minimum and maximum longitude of 5 and 20 degree respectively. This resulted in a subset of 755 grid cells. For the southern European region, a grid box was created over France with minimum and maximum latitudes of 41.5 and 51.1 degree respectively and a minimum and maximum longitude of -5.1 and 6 degree respectively. Summary statics for the prognostic state variables in these regions are listed in table S1.

Table S1: Exemplary summary statistics of the seven prognostic target variables over two European training data subsets, northern and southern. Mean, standard deviation and their ratio (Signal-to-noise ratio, SNR) are aggregated over times and grid cells.

	Northern Europe			**Southern Europe**		
	Mean	Standard dev.	SNR	Mean	Standard dev.	SNR
SWVL1	0.2858	0.0465	6.399	0.2929	0.0905	3.3495
SWVL2	0.2802	0.0433	6.7156	0.2949	0.0807	3.7471
SWVL3	0.2685	0.0449	6.1867	0.2905	0.0688	4.3294
STL1	278.1943	6.2549	45.6081	285.026	6.9303	41.67
STL2	278.0838	5.6185	50.9871	284.9675	6.007	48.0569
STL3	277.8869	4.4763	65.1102	284.847	4.8378	59.6688

SNOWC	36.5848	37.9657	0.889	2.7402	9.5118	0.1722

S2 Model development

S2.1 LSTM

S2.1.1 Architecture and Hyperparameter selection

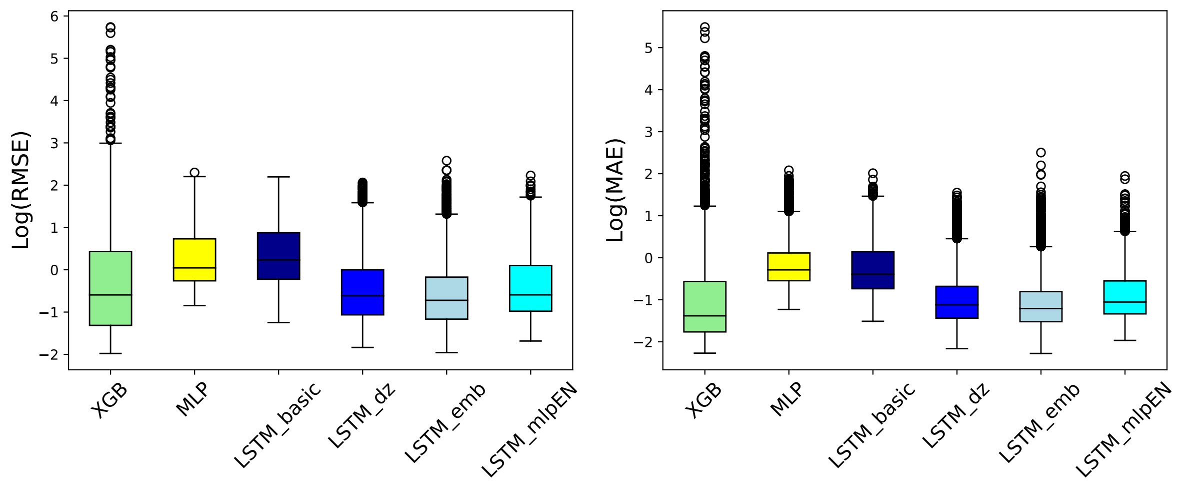


Figure S1: LSTM architecture development. LSTM_basic considers prognostic state variables in the encoder as input, LSTM_dz adds and incremental term in the loss function, LSTM_emb encodes prognostic state variables to inform encoder hidden and cell states and LSTM_mlpEN uses an MLP encoder to inform the hidden states of the LSTM decoder.

The coarse architectural modules of the LSTM were manually selected. In a seeded experiment, we (1) added the first differences to the loss function, (2) added an embedding layer that transfers prognostic states to the initial hidden states of an LSTM encoder, (3) tested an MLP as encoder to the LSTM decoder (see figure S1). While we accepted the methodology of (1) and (2), we rejected (3) and continued with an LSTM encoder network. Detailed architectural choices were made with the Bayesian hyperparameter tuning framework Optuna (Akiba et al., 2019). The best performance was reached with equal parametric capacities in the encoder and decoder part. The final LSTM thus has a hidden size of 200 and in each layer with a depth of 3 in the encoder and decoder part. The parts are connected by a hidden and a cell adapter that consist each of a single linear layer that transfers the hidden and cell state from the encoder to the decoder, performing width.
The hyperparameters for training were a dropout of 0.1265, a learning rate of 0.0005 and weight decay of 0.0001.

```
1  | hidden_encoder   | Linear       | 5.8 K
2  | cell_encoder     | Linear       | 5.8 K
3  | lstm_encoder     | LSTM         | 824 K
4  | hidden_adapter   | Linear       | 40.2 K
5  | cell_adapter     | Linear       | 40.2 K
6  | lstm_decoder     | LSTM         | 824 K
7  | mlp_decoder      | Linear       | 1.4 K

1.7 M     Trainable params
0         Non-trainable params
1.7 M     Total params
6.942     Total estimated model params size (MB)
```

S2.1.2 Training Leadtime

The forget gate mechanism allows LSTMs to store information over long time sequences without the loss old information (e.g. (Nearing et al., 2024)). We conducted a seeded experiment on the effect of the training lead time in the decoder part on the LSTMs predictive accuracy within the capacity of our computational resources. At the exact same hyperparameter setting, the model was trained at six different lead times for 220 epochs. Note that lead times are reported in time steps on the 6-hourly resolution, i.e. a lead time of ten is equivalent to a 2.5 days forecast, a lead time of 20 to 5 days, etc. All models converged in the training period. While predictive accuracy increases at longer training lead times, so does the training runtime (see table S1 and figure S3). The training lead time for results we show in the main manuscript was 40 on the European and 60 on the continental scale.

Table S2: Summary of runtimes and total mean predictive scores at different training lead times. Training was conducted on 2 GPUs, Evaluation on 1 GPU. Runtimes are reported in minutes.

Training Leadtime	Training Runtime	Evaluation Runtime	Total RMSE	Total MAE	Total R2
10	420.72	0.016	1.6520	0.9929	0.9991
20	664.27	0.009	1.3992	0.8012	0.9991
30	905.47	0.009	1.1084	0.5958	0.9992
40	1289.27	0.008	0.9138	0.4983	0.9994
50	1954.62	0.009	0.7411	0.3691	0.9997
60	2338.86	0.009	0.6918	0.3459	0.9998

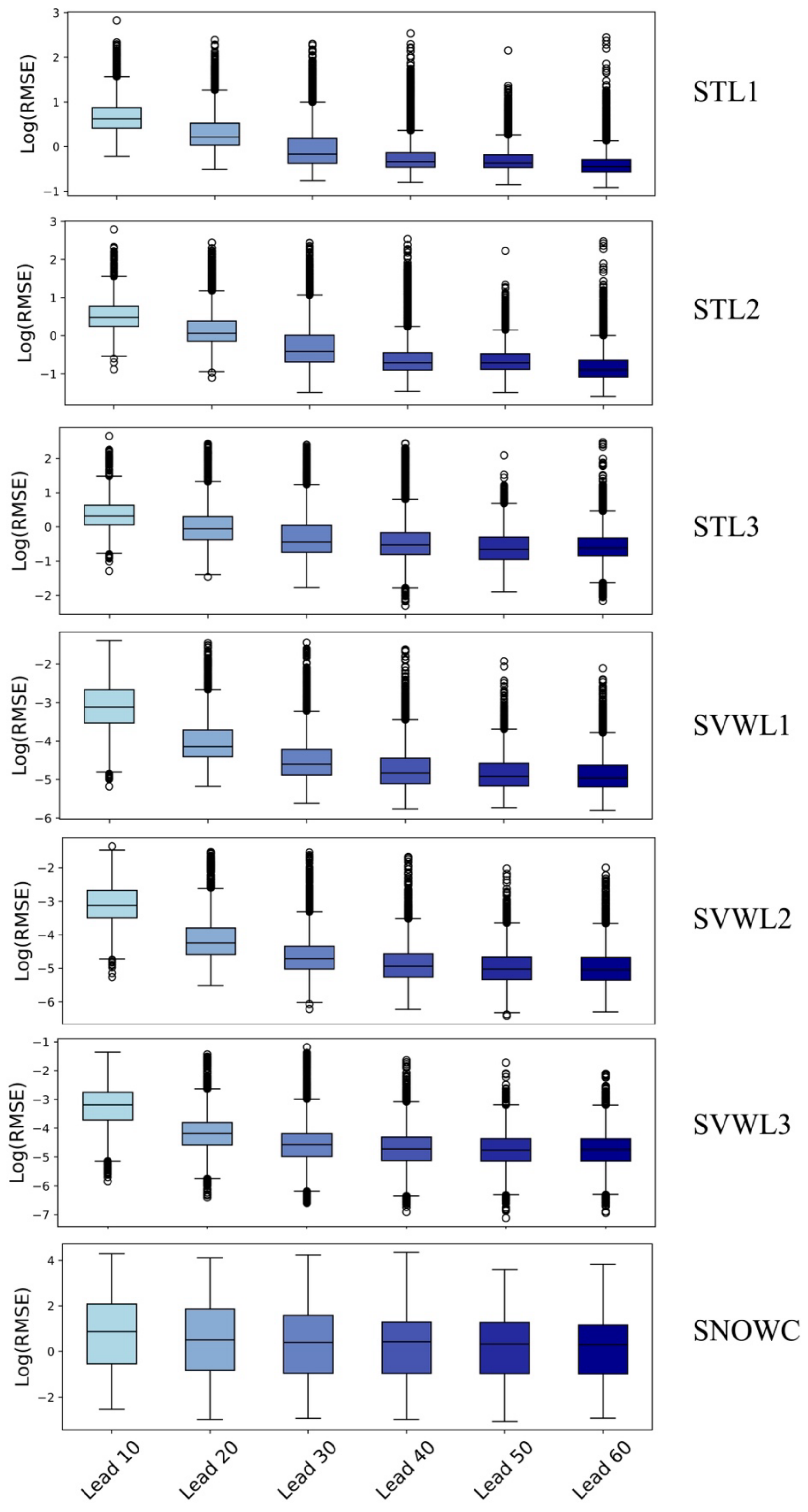


Figure S2: Targetwise predictive accuracy by training at different lead times.

S2.1.2 Encoder sequence length

Like the experiment on the training lead time, we conducted a seeded experiment on the effect of encoder sequence length on predictive accuracies at a training lead time of 40. However, in contrast to varying the training lead time, changing the encoder sequence will change the model structure(Hochreiter and Schmidhuber, 1997). The effect not being as clear as for training lead time, we may hypothesise an advantage of shorter sequence length for the s oil related variables (see figure S3). Models that produced results in the main manuscript were trained with encoder sequence lengths of 24.

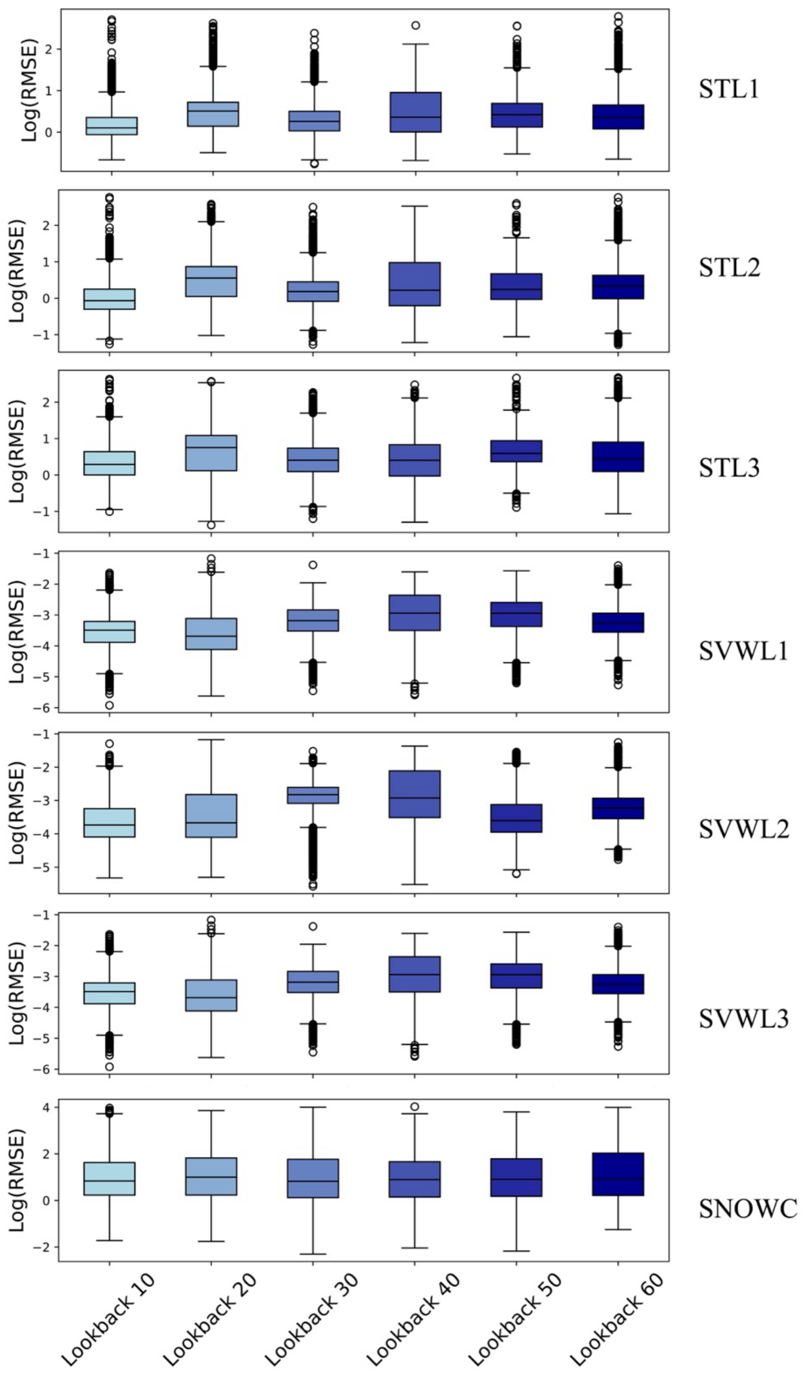


Figure S3: Targetwise predictive accuracy by training with different lookback times, i.e. encoder sequence lengths.

S2.2 MLP

S2.2.1 Methodology

The multilayer perceptron is a neural network regression-type model that approximates a non-linear function $f: x \rightarrow y$, where x in this study is a vector of static, dynamic and prognostic state variables, and y the vector of prognostic state variables. The optimal function f representing this mapping is unknown and its best possible approximation $f * (x)$ is found in a stochastic gradient-based optimization procedure. In practice, n non-linear functions are chained to a feed-forward neural network to create a hierarchically structured latent space with so-called hidden layers, whereby each j-th hidden layer of the network can be expressed as

$$y_j = \varphi_j(\sum_i x_i A_{j,i} + b_j).$$

Here, $A_{j,i}$ constitutes the weight matrix, i.e. the networks parameter, b_j the bias vector, i.e. an estimated intercept, and φ_j a non-linear activation function. The activation function is here the Rectified-Linear Unit (ReLU) that is defined as

$$\varphi := max(y, 0).$$

In a hidden layer, the input x_i is mapped to a predetermined number of hidden nodes, i.e. the layers' size, determined by the second dimension in $A_{j,i}$. The transformation with φ_j returns a weighted version of the node. When weighted to zero, a node is dead unless regularized by the bias. The MLP in trained with dropout, referring to an additional regularization technique that applies a random binary mask to all input and hidden nodes of the network at each training step, where a node with the zero at the mask is dead in this training step. The probability of ones in the mask is defined as a hyperparameter (see below) (Goodfellow et al., 2016).

S2.2.2 Architecture and Hyperparameters

The MLP has four hidden layers of sizes 122, 47, 103 and 117. It is trained with a learning rate of 0.00093, dropout of 0.18526 and a weight decay of 0.00013. The batch size and training rollout were determined by GPU memory and are 4 and 4 respectively. The total numbers of trainable parameters in the MLP is 28.8K.

S2.3 XGB

XGB was trained with a learning rate of 0.3, a maximum depth of ten and 256 trees. In contrast to neural network hyperparameter optimization, only a manual exploration on tuning the learning rate and depth was conducted.

S3 Model performances

S3.1 Model development: Europe

S3.1.1 Objective forecast accuracies

All emulators approximated the numerical model with high total scores on average, i.e. R^2 values larger than 0.99, MAEs smaller than 1 and RMSE smaller than approximately 1.60. The LSTM scored highest across all metrics, followed by XGB and then MLP, even though the latter got second place in RMSE. LSTM improved in MAE by 50% towards XGB (see table S2). These results differentiate for individual target variables. LSTM shows specifically strong performance across scores in forecasting soil water volume.

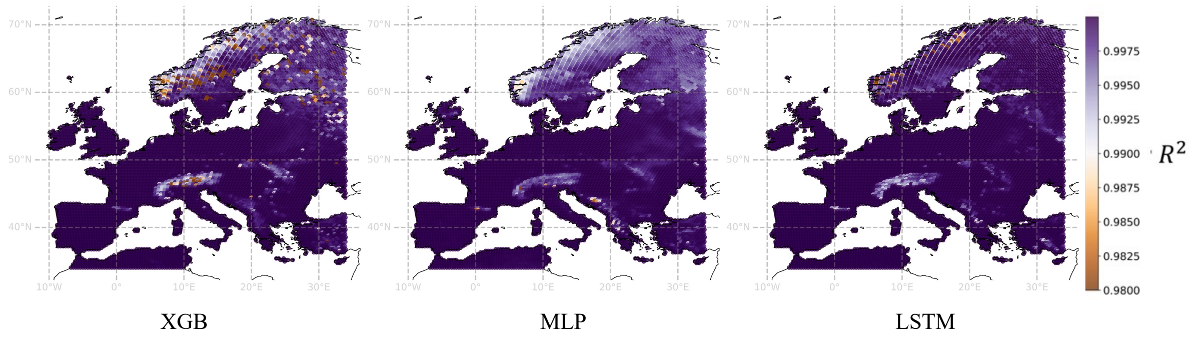


Figure S4: Mean R-squared aggregated per grid cells over 6-hourly lead times on the European subset for model development.

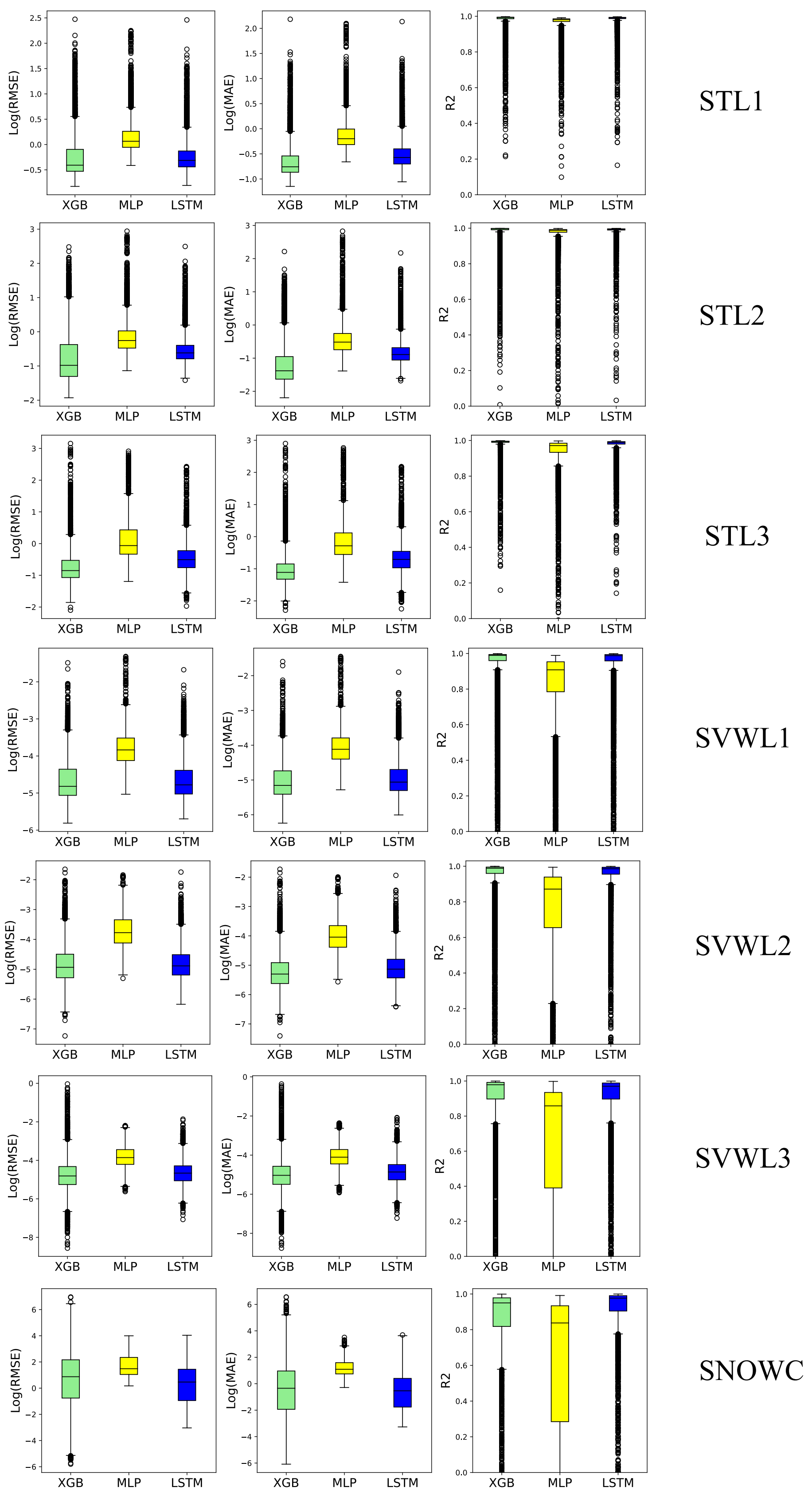

Log(RMSE)
Log(MAE)
R2
XGB
MLP
LSTM
STL1
STL2
STL3
SVWL1
SVWL2
SVWL3
SNOWC

Figure S5: Total distribution of mean scores, aggregated over 6-hourly lead times by grid cell, variability here thus refers to performance differences among grid cells.

Table S3: Emulator total mean scores, aggregated over variables, time and space.

Variable	**Model**	$RMSE$	MAE	R^2
All variables	XGB	1.6035	0.8091	0.9960
	MLP	1.6013	0.9611	0.9991
	LSTM	**0.8507**	**0.4361**	**0.9996**

Table S4: Emulator mean scores on soil water volume forecasts for the European subset, aggregated over space and time.

Variable	**Layer**	**Model**	$RMSE$	MAE	R^2
Soil water volume	1	XGB	0.0122	0.0084	0.84420
		MLP	0.0249	0.0192	0.7340
		LSTM	**0.0114**	**0.0083**	**0.8655**
	2	XGB	0.0104	**0.0070**	0.8512
		MLP	0.0280	0.0216	0.5781
		LSTM	**0.0097**	0.0073	**0.8543**
	3	XGB	0.0149	0.0112	0.6426
		MLP	0.0252	0.0197	0.2380
		LSTM	**0.0114**	**0.0092**	**0.7379**

Table S5: Emulator mean scores on soil temperature forecasts for the European subset, aggregated over space and time.

Variable	**Layer**	**Model**	$RMSE$	MAE	R^2
Soil temperature	1	XGB	0.8730	**0.5735**	**0.9750**
		MLP	1.2629	0.9601	0.9352
		LSTM	**0.8450**	0.6347	0.9642
	2	XGB	**0.6449**	**0.3843**	**0.9721**
		MLP	0.9984	0.7580	0.3130
		LSTM	0.6563	0.4852	0.9480
	3	XGB	**0.6221**	**0.4368**	**0.9126**
		MLP	1.3464	1.0020	-0.5478
		LSTM	0.7530	0.5884	-0.5807

Table S6: Emulator mean scores on snow cover forecasts for the European subset, aggregated over space and time.

Variable	**Layer**	**Model**	$RMSE$	MAE	R^2
Snow cover	top	XGB	9.0471	4.2423	**0.5325**

	MLP	7.5232	3.9469	0.4383
	LSTM	**3.6676**	**1.3196**	0.4345

S3.2 Model testing: Europe

S3.2.1 Quantile Correlations

We visualised quantile correlations for each prognostic state variable. The mean and standard deviation of quantiles were computed in 10% steps for emulator and ECland forecasts and plotted against each other. The results highlight the state values where model predictions align perfectly, i.e. quantiles are found on the correlation line, and where the emulator overestimate (quantiles above regression line) or underestimate (quantiles below regression line) ECland prognostic states (see figure S6).

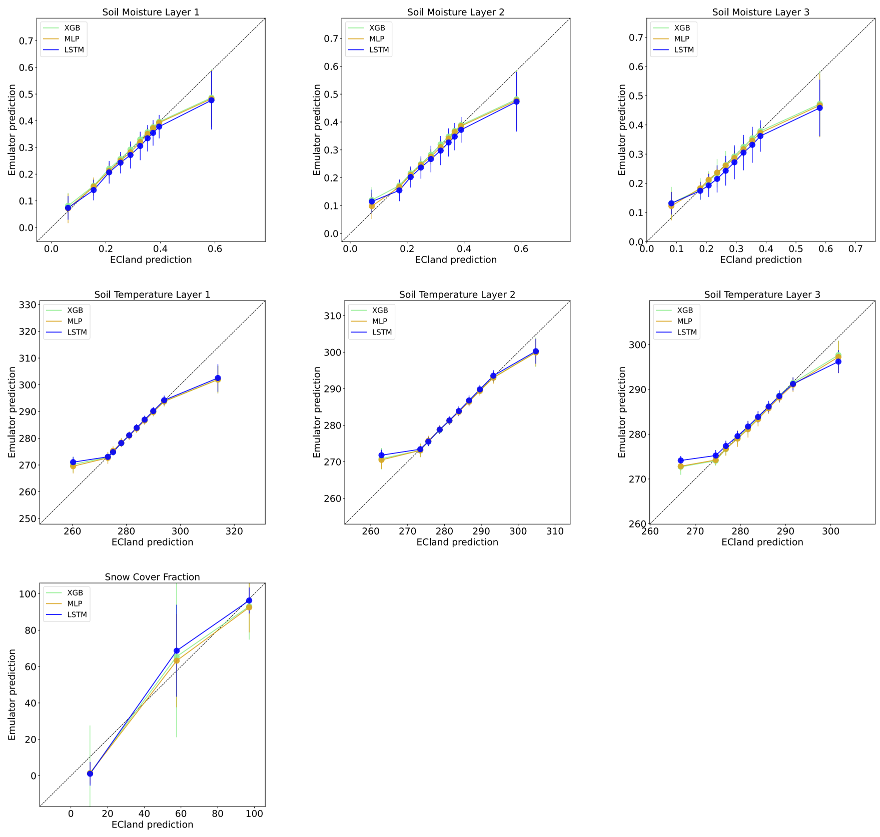


Figure S6: Quantile correlations for all prognostic target variables and all emulators. Emulator quantile predictions are on the y-axis, ECland predictions on the x-axis. The dashed black line indicates their perfect correlation.

S4 Evaluation

S4.1 Forecast horizons: climatology

Below we show examples of forecast horizons computed for three single prognostic state variables, soil water volume and temperature at layer one and snow cover (figures S7-9). Disentangling these highlights at the example of snow cover that in aggregating the anomaly correlation over prognostic state variables, negative and positive effects may cancel each other out: the snow cover limitation in the southern European subregion for the MLP forecasts is not as visible in the total horizons (see main manuscript).

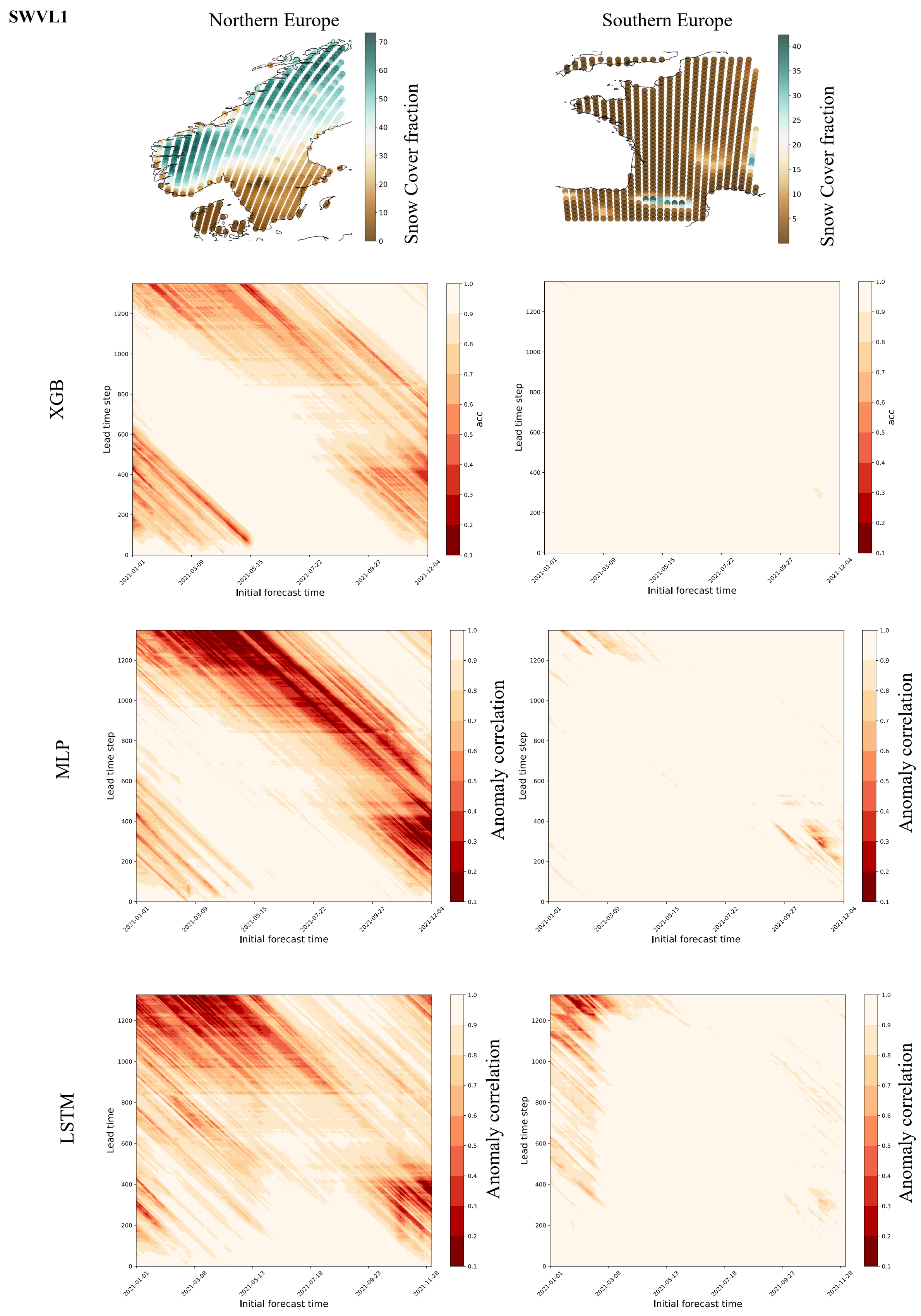


Figure S7: Forecast horizons for Soil water volume layer 1.

STL1

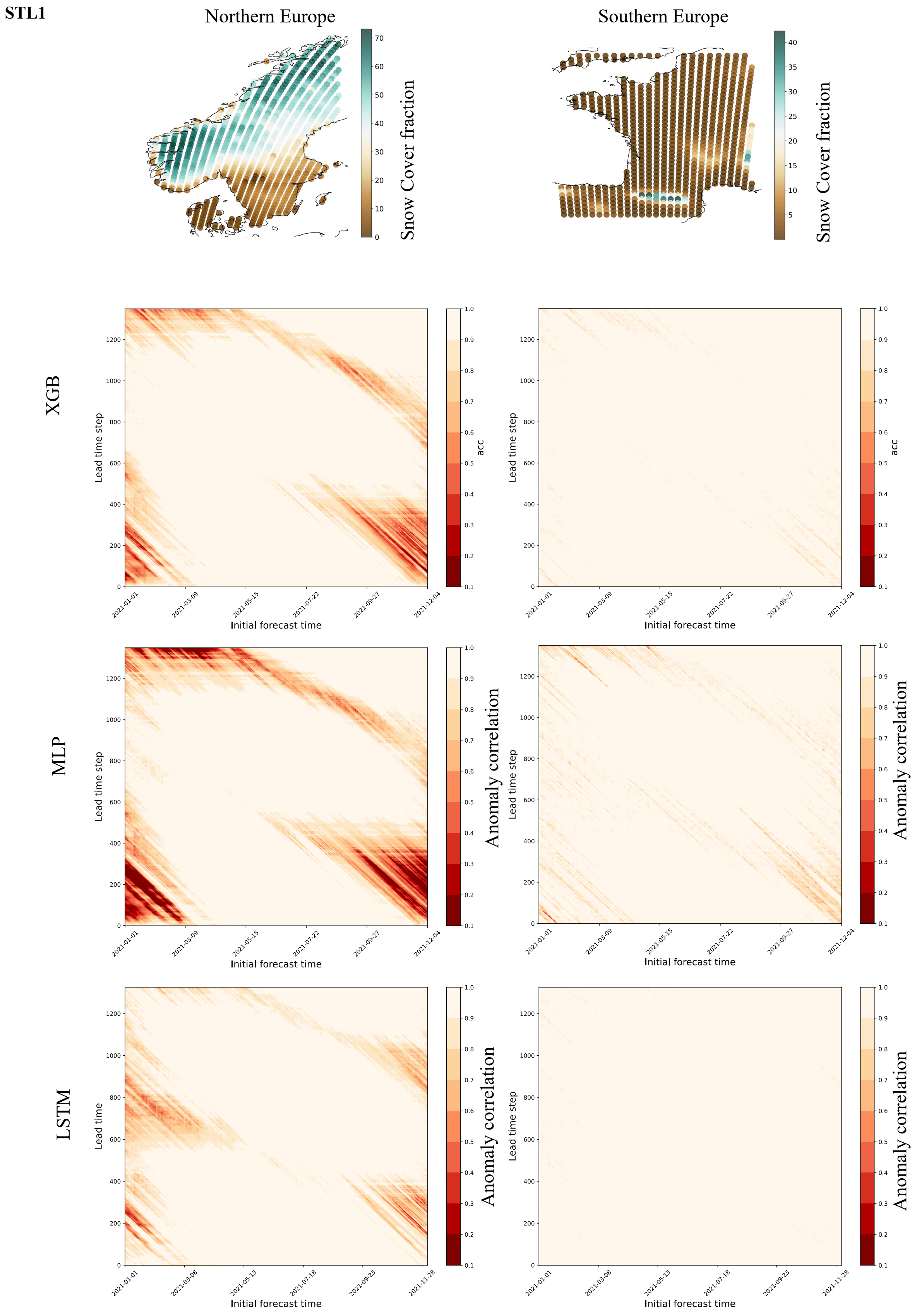


Figure S8: Forecast horizons soil temperature Layer 1.

SNOWC

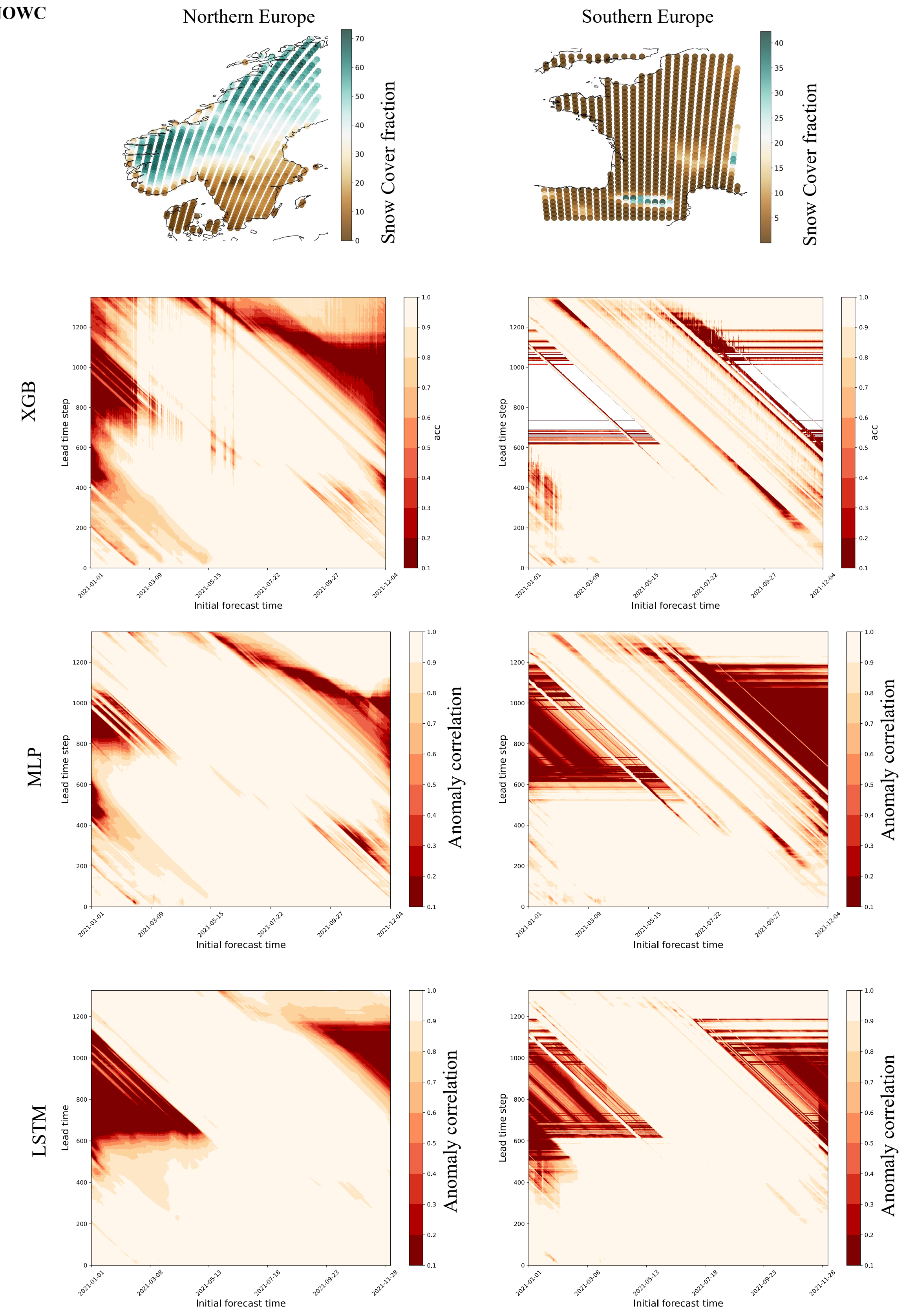

Northern Europe
Southern Europe
Snow Cover fraction
Snow Cover fraction
XGB
MLP
LSTM
Lead time step
Lead time
Initial forecast time
acc
Anomaly correlation
2021-01-01
2021-03-09
2021-05-15
2021-07-22
2021-09-27
2021-12-04
2021-03-08
2021-05-13
2021-07-18
2021-09-23
2021-11-28

Figure S9: Forecast horizons snow cover.

S4.2 Time series sample: Northern Europe

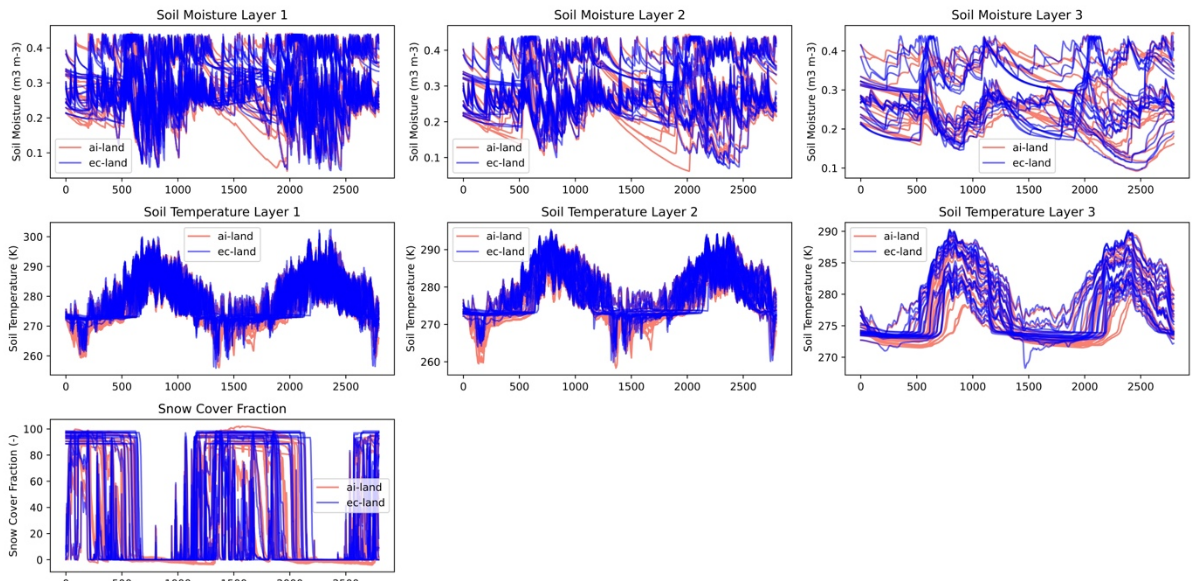


Figure S10: MLP forecast on two test years 2021, 2022 for a random selection of grid cells from the northern European region.

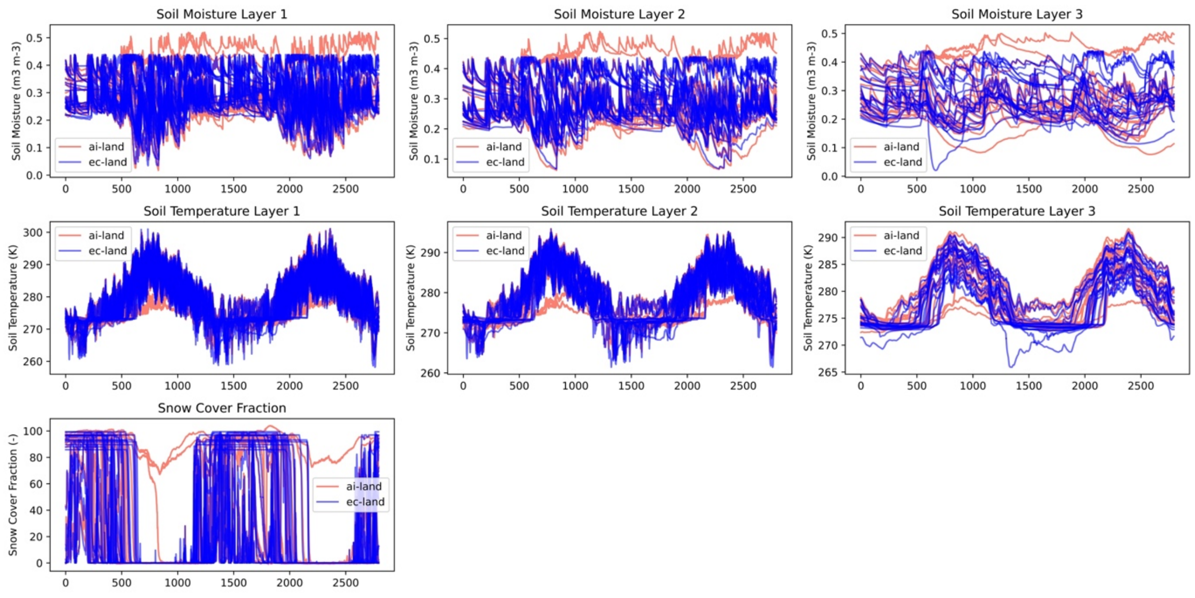


Figure S11: LSTM forecast on two test years 2021, 2022 for a random selection of grid cells from the northern European region.

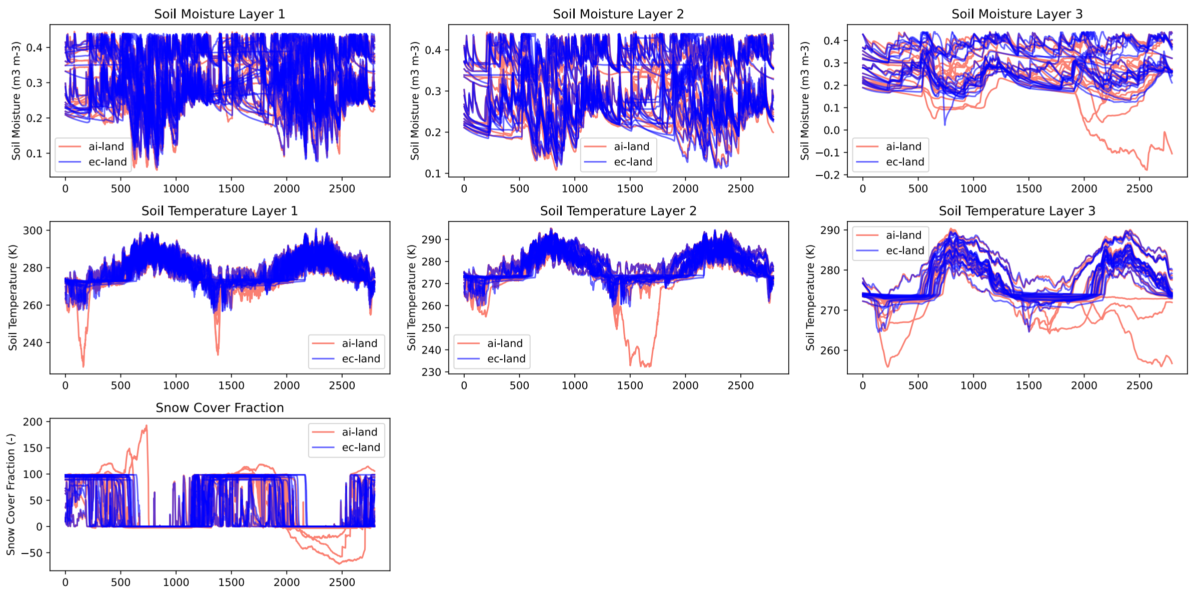


Figure S12: XGB forecast on two test years 2021, 2022 for a random selection of grid cells from the northern European region.

S4.3 Model extrapolation: Globe, low-resolution (TCO199)

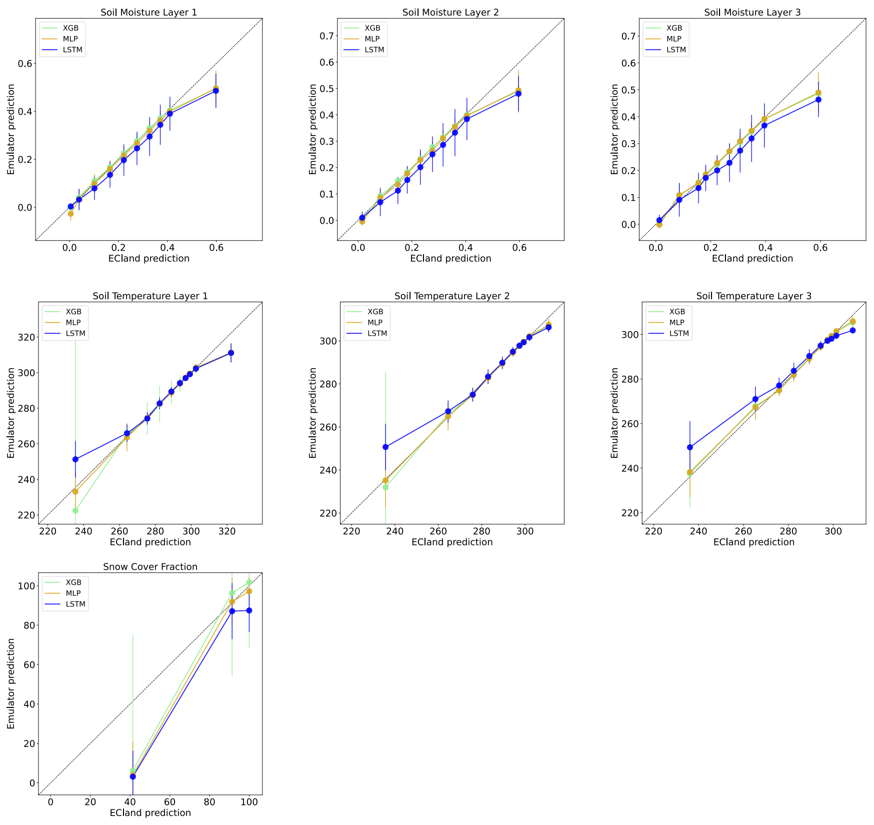
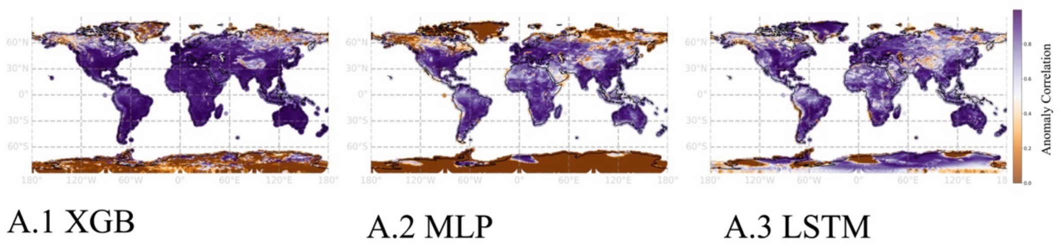
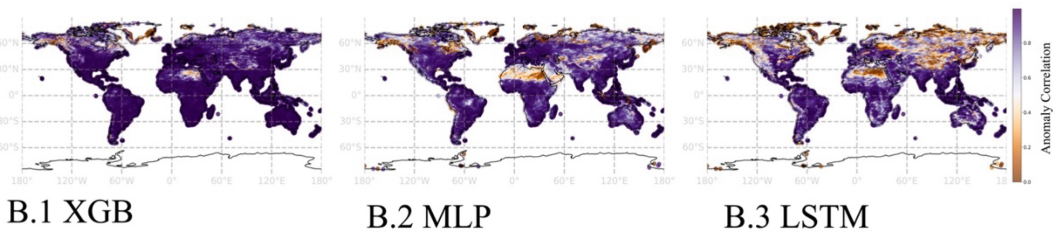


Figure S13: Quantile correlations, visualised as described in section 3.2.1 for continental model testing.

A. Soil temperature (L1)



B. Soil moisture (L1)



C. Snow Cover

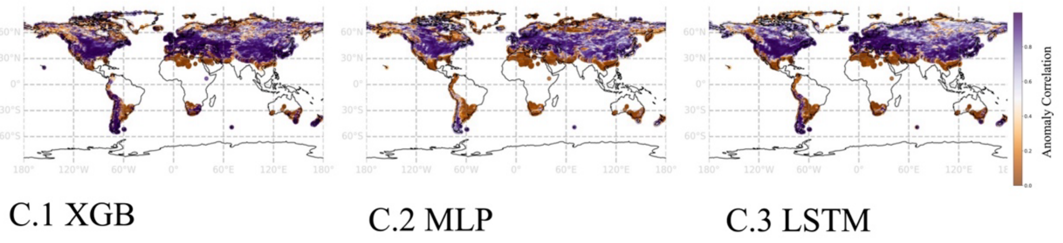


Figure S14:Global distribution of Anomaly Correlation for three prognostic state variables. Uncoloured areas indicate regions where the ACC is not defined

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Anchorage AK USA, 2623–2631, https://doi.org/10.1145/3292500.3330701, 2019.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep learning, The MIT Press, Cambridge, Massachusetts, 775 pp., 2016.

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T. Y., Weitzner, D., and Matias, Y.: Global prediction of extreme floods in ungauged watersheds, Nature, 627, 559–563, https://doi.org/10.1038/s41586-024-07145-1, 2024.