

# Fully Convolutional Networks for Dense Water Flow Intensity Prediction in Swedish Catchment Areas

Aleksis Pirinen<sup>1</sup>, Olof Mogren<sup>1</sup> and Mårten Västerdal<sup>2</sup>

<sup>1</sup>RISE Research Institutes of Sweden

<sup>2</sup>Department of City Planning and Sustainability, Kungsbacka municipality

{aleksis.pirinen@ri.se, olof.mogren@ri.se, marten.vasterdal@kungsbacka.se}

**Abstract**—Intensifying climate change will lead to more extreme weather events, including heavy rainfall and drought. Accurate streamflow prediction models which are adaptable and robust to new circumstances in a changing climate will be an important source of information for decisions on climate adaptation efforts, especially regarding mitigation of the risks of and damages associated with flooding. In this work we propose a machine learning-based approach for predicting water flow intensities in inland watercourses based on the physical characteristics of the catchment areas, obtained from geospatial data (including elevation and soil maps, as well as satellite imagery), in addition to temporal information about past rainfall quantities and temperature variations. We target the one-day-ahead regime, where a fully convolutional neural network model receives spatio-temporal inputs and predicts the water flow intensity in every coordinate of the spatial input for the subsequent day. To the best of our knowledge, we are the first to tackle the task of *dense* water flow intensity prediction; earlier works have considered the prediction of flow intensities at a sparse set of locations at a time. An extensive set of model evaluations and ablations are performed, which empirically justify our various design choices. Code and preprocessed data have been made publicly available at <https://github.com/aleksispi/fcn-water-flow>.

## I. INTRODUCTION

As climate change intensifies, hydrological conditions will change. This will manifest itself both in the form of water shortages and as flooding in cases of intense precipitation. According to the Swedish Environmental Protection Agency, the climate in Sweden is becoming warmer and wetter [1], and municipalities are encouraged to increase their climate adaptation efforts, especially regarding mitigating the risks of, and damages associated with, flooding [12], [15]. The effects of climate change on rainfall-runoff will be more severe further north [14]. At the same time, the hydrological conditions in Sweden have been severely disturbed during the last two hundred years, with wetlands being drained and natural streams being straightened, which will further increase the effects of extreme weather events.

Hydrological modeling can shed light on the dynamics of water flow and how it is affected by various aspects of the environment. This can in turn allow for making informed decisions about the efficacy of nature-based climate change adaptation techniques such as wetland restoration, urban greening, and soil protection. Traditional hydrological models are based on expert knowledge and physical properties such as the preservation of volume, which have to be specified a priori. These work well for a certain domain if

they are properly calibrated, but have difficulties generalizing to wider environmental categories [17]. Statistical data driven modeling, including machine learning (ML), is an alternative which has the potential to become more robust as long as the model can be trained on a large enough dataset with a suitable learning signal. This way, not only can the flow intensity be estimated for any given water course following a heavy precipitation event, but also the response time of the given area, i.e. an estimation of the time lapse from the precipitation event to peak flow. Such information is vital to better understand flood risks and the effects of flood and drought mitigation, as well as general hydrological implications from changes in land use.

In this work we propose an ML-based approach for water flow intensity prediction that leverages the physical characteristics of a catchment area. We target the one-day-ahead regime, where a fully convolutional neural network [11] receives spatio-temporal inputs and predicts the water flow intensity at every coordinate for the subsequent day (the same modeling should however be able to handle other time horizons with minor modifications). Two important novelties of our proposed approach are:

- In addition to temporal data (past rainfall and temperatures), we include spatial data as inputs to the modeling, provided as satellite imagery and several derived GIS layers. This allows the model to build internal representations about relationships between temporal and spatial aspects of the local environment (including land cover, soil depth and moisture, and elevation).
- Using a fully convolutional model, we tackle the task of *dense* water flow intensity prediction, as opposed to only predicting flow intensities for a sparse set of spatial locations. To the best of our knowledge, we are the first to consider the dense prediction task.

The remainder of this paper is organized as follows. In Section II we provide a brief overview of the related work. Then, in Section III, we describe in detail the data we have used for modeling, training and evaluation. In Section IV we explain our approach for tackling the water flow intensity prediction task, and our proposed approach is empirically evaluated against alternative methods in Section V-B. Finally, the paper is concluded in Section VI.

## II. RELATED WORK

Water flow prediction (also known as *stream forecasting* or *rainfall-runoff modeling*) for rivers in the U.S. have been modeled using Long Short-Term Memory (LSTM [7]) networks [8], [5], [4], [6]. The modeling follows a traditional setup inspired by earlier physics-based hydrological models such as the U.S. National Water Model (NWM), based on WRF-Hydro [2]. Jia et al. [8] modeled river segments using an LSTM network with graph convolutions. One segment in the river network corresponds to a distance that the water flows during approximately one day. Input features include daily average precipitation, daily average air temperature, date of the year, solar radiation, shade fraction, potential evapotranspiration, elevation, length, slope, and width of each segment. Models were trained using a physics-informed setup where a traditional flow model acted as a teacher for the machine learning model. LSTM networks have also been used for post-processing the output from the NWM [5]. Similar to our work, most of these prior works have focused on next-day predictions. However, there are examples of hourly predictions [6].

Others have also employed convolutional neural networks (CNNs) for stream forecasting [3], [18], [13]. However, in contrast to us, these works do not incorporate spatial data from satellites or GIS, but instead model only the much lower-dimensional data (in single coordinates or very small neighborhoods, not entire areas as in our setup) provided as a feature vector for each time step, similar to the models using LSTMs. More broadly, deep learning has been used for many related tasks, such as groundwater level estimation [21], water quality estimation [19], and rainfall-runoff [10].

While some of the above mentioned works on streamflow estimation include information about the near environment (such as elevation and slope), none of them use detailed spatial information inputs as is proposed in our work. The use of fully convolutional neural networks to encode this information, in combination with traditional inputs such as rainfall and temperature, has the potential of representing more complex relationships and can result in a more detailed view of the near environment. It also enables us to perform *dense* water flow intensity prediction, different to prior works who consider the prediction of flow intensities in a sparse, discrete set of points.

## III. DATASET DESCRIPTION

We use data from 12 locations in Sweden, based on where the Swedish Meteorological and Hydrological Institute (SMHI) has stations for measuring weather and water flow data. These locations are Jönköping (Tabergsån), Knislinge (Almaån), Krycklan, Skivarp (Skivarpsån), Skövde (Ösan), Torup (Kilan), Tumba (Saxbroån), Dalbergsån, Degeå, Hässjaån, Lillån, and Lillån-Blekinge (see Figure 1).

In each location we have access to the following spatial data layers (see also Figure 2):

- satellite RGB image (Sentinel-2) from the Land Survey of Sweden (Lantmäteriet), 10m × 10m resolution;

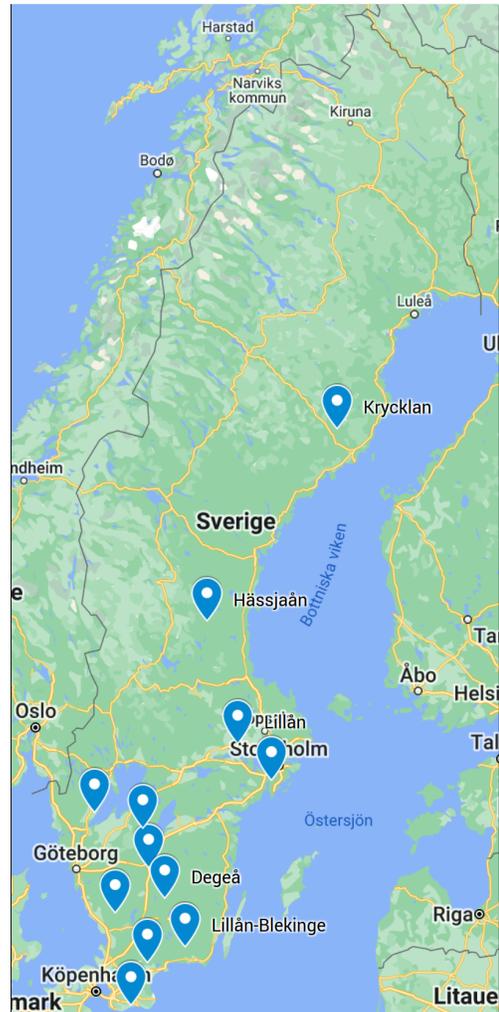


Fig. 1. The data used in this paper comes from 12 locations in Sweden: Jönköping (Tabergsån), Knislinge (Almaån), Krycklan, Skivarp (Skivarpsån), Skövde (Ösan), Torup (Kilan), Tumba (Saxbroån), Dalbergsån, Degeå, Hässjaån, Lillån, and Lillån-Blekinge. Note that Lillån and Lillån-Blekinge are far apart (Lillån is north-west of Stockholm).

- elevation map from the Land Survey of Sweden, 50m × 50m resolution;
- terrain slope map from the Land Survey of Sweden, 50m × 50m resolution;
- soil moisture map the Swedish University of Agricultural Sciences, 2m × 2m resolution;
- land cover map the Swedish Environmental Protection Agency, 10m × 10m resolution;
- soil type map from the Geological Survey of Sweden, 10m × 10m resolution;
- soil depth map from the Geological Survey of Sweden, 10m × 10m resolution;
- hydraulic conductivity map from the Geological Survey of Sweden, 100m × 100m resolution.

The elevation map provides each coordinate's elevation above the ocean level, whereas the terrain slope map provides the slope of each coordinate (obtained by computing differences between adjacent coordinates of the elevation map).

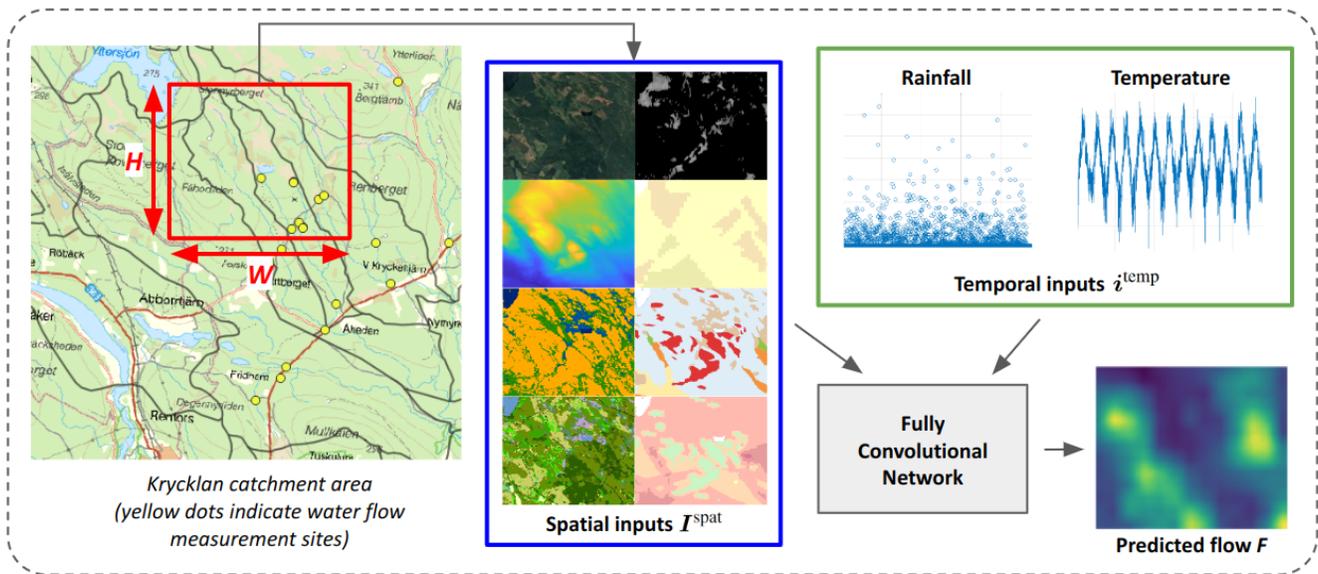


Fig. 2. Overview of our machine learning (ML) approach for dense water flow intensity prediction in catchment areas. The fully convolutional neural network receives both spatial and temporal inputs and produces a map of predicted water flow intensities. The spatial input  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$  represents relevant properties of the region (red rectangle, which can be located anywhere) in which to perform water flow intensity prediction (e.g. elevation map, soil moisture, land cover). The temporal inputs  $\mathbf{i}^{\text{temp}} = \{r_j, \tau_j\}_{j=t-T}^{t-1}$  are the daily average rainfall amounts  $r_j$  and temperatures  $\tau_j$  over the past  $T$  days, provided at times  $t-T, t-T+1, \dots, t-1$ . The model then produces the flow map  $\mathbf{F} \in \mathbb{R}^{H \times W}$ , where each pixel  $(k, l)$  represents the model’s estimate of the daily average water flow intensity at location  $(k, l)$  in the input map during day  $t$ . Note that our model is only trained in a very sparse set of coordinates (since water flow intensity measurement stations are very scarcely located), but the model nonetheless predicts flow intensities in every coordinate, not only in those for which contain flow intensity measurement stations.

All input data were provided as spatially aligned sets of raster maps, each of size  $825 \times 1244$  pixels. For all locations except the Krycklan catchment area, there is exactly one water flow intensity measurement station. In Krycklan there are 14 such measurement stations, so in total there are 28 water flow intensity measurement stations in the dataset. In each measurement site, the daily average water flow intensity ( $\text{m}^3/\text{sec}$ ) is provided. The specific length of each time series varies, but generally span several decades (in average about three decades).

In each location we also have access to two additional time series – daily cumulative rainfall (mm) and daily average temperature (degrees Celsius). These were obtained from the Swedish Meteorological and Hydrological Institute (SMHI), and are often measured some distance away from the water flow intensity measurement sites (typically 1-3 kilometers).

It is common with missing measurements in the time series. For those time series which are used as model inputs (see Section IV), this is remedied by linearly interpolating the missing values between end points. Note that this is not done for the regressor (water flow intensity), since we only want the model to learn on and be evaluated on actual measurements.

**Data preprocessing.** We perform normalization of both the spatial and temporal data. Specifically, the spatial inputs are normalized to the  $[0, 1]$ -range by dividing with the maximum value (layer-wise) across all locations. A similar  $[0, 1]$ -normalization is performed for the temporal inputs (rainfall, temperature, and water flow intensity). We

also tried another common normalization technique, where the variables are normalized to zero mean and unit variance, but empirically found that the  $[0, 1]$ -normalization works best in our setup.

#### IV. METHODOLOGY

In this section we provide an overview of the approach that we have developed for tackling the water flow intensity prediction task. See Figure 2 for an overview of our model and setup.

Our model leverages both spatial inputs  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$  and temporal inputs  $\mathbf{i}^{\text{temp}} = \{r_j, \tau_j\}_{j=t-T}^{t-1}$  (cf. Section III), in order to predict water flow intensities  $f_t$  at time<sup>1</sup>  $t$ . The task of the model is to predict the water flow intensity  $f_t^{h,w}$  at every coordinate  $(h, w)$  in a given geographical area of size  $H \times W$  given  $\mathbf{I}^{\text{spat}}$  and  $\mathbf{i}^{\text{temp}}$ . For the temporal inputs, we have chosen to only use readily available rainfall  $r_{t-T}, \dots, r_{t-1}$  and temperature data  $\tau_{t-T}, \dots, \tau_{t-1}$  for the past  $T$  days (with  $T = 20$  in our setup), and not water flow intensity data  $f_{t-T}, \dots, f_{t-1}$ , which is often unavailable in practice. In particular, note that water flow intensity is only measured at a very sparse subset of all coordinates in each location – and there are many locations in Sweden (and beyond) where no such measurement setups exist at all. Hence, for the model to be useful in a much larger set of contexts, it does not rely on past water flow intensities as input. However, in Section V we also compare with model variants that include past water flow intensities when predicting future flow intensities.

<sup>1</sup>Similar to most prior works, we target next-day prediction, but the model and approach can be extended to predict further into the future.

The spatial input  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$  contains relevant information regarding land and topological properties that affect the water flow intensity in any given coordinate. These spatial input layers were introduced in Section III. In our setup we let  $H = W = 100$ , which corresponds to a real-world area of size  $1\text{km} \times 1\text{km}$ . The number of layers  $C$  is 10 in our case (three layers for the RGB satellite images, and one layer each for the other types of spatial input).

Since the task is to predict the water flow intensity in every coordinate in a map of size  $H \times W$ , based (in part) on spatial inputs of size  $H \times W \times C = 100 \times 100 \times 10$ , we have opted for a fully convolutional neural network<sup>2</sup> (FCN) [11]. This architecture expects a spatial input at one end, and gives a spatial output at the other end. To achieve this, we first concatenate the spatial and temporal data  $\mathbf{I}^{\text{spat}}$  and  $\mathbf{i}^{\text{temp}}$  into a unified input  $\mathbf{I} \in \mathbb{R}^{H \times W \times (C+2T)}$ . The first  $C$  channels are identical to  $\mathbf{I}^{\text{spat}}$ , while the last  $2T$  channels are obtained by tiling rainfall  $r_{t-T}, \dots, r_{t-1}$  and temperature data  $\tau_{t-T}, \dots, \tau_{t-1}$  into  $H \times W$ -dimensional maps that are concatenated along the channel-dimension (each such map contains  $HW$  copies of a single value  $r_i$  or  $\tau_i$ , for  $i \in \{t-T, \dots, t-1\}$ ). Given an input  $\mathbf{I}$ , the water flow intensity mapping is straightforward:  $\mathbf{F} = g_{\theta}(\mathbf{I})$ , where  $\theta$  denotes the learnable parameters of the FCN  $g$ . We also evaluate and compare with other architecture variants in Section V.

#### A. Model Training

We randomly set aside 9 of the 12 data locations for training and 3 for validating the models. Specifically, the models are evaluated in Jönköping (Tabergsån), Hässjaån and Lillån, and trained on the other 9 locations; please refer to Section III for details about the dataset. In particular, note that Lillån-Blekinge is in the training set, but it is at a vastly different location than the validation set location Lillån (cf. Figure 1).

Each training input  $\mathbf{I}$  in a batch is generated by randomly sampling a location and time period from the training set. Once a specific location has been randomly selected, we randomly sample a sub-region of size  $H \times W$  which contains a water flow measurement site at the given location – see Figure 3. This results in the spatial input  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$ . After having sampled a spatial location, we then concatenate the temporal information  $\mathbf{i}^{\text{temp}}$  from a randomly sampled time interval (of  $T$  consecutive days), to obtain the input  $\mathbf{I} \in \mathbb{R}^{H \times W \times (C+2T)}$ .

There are roughly  $25 \cdot 100 \cdot 100 = 250,000$  different spatial training inputs (25 measurement sites, and at each site there are roughly  $100 \cdot 100$  possible locations for an enclosing rectangle of size  $H \times W = 100 \times 100$  – see Figure 3). Note however that there is a spatial overlap between all different rectangles at a given site, which significantly reduces the data variability, compared to if all 250,000 different rectangles would have come from different locations. The sites have on average roughly three decades of daily rainfall, temperature and water flow averages, which means there are  $30 \cdot 365 \approx$

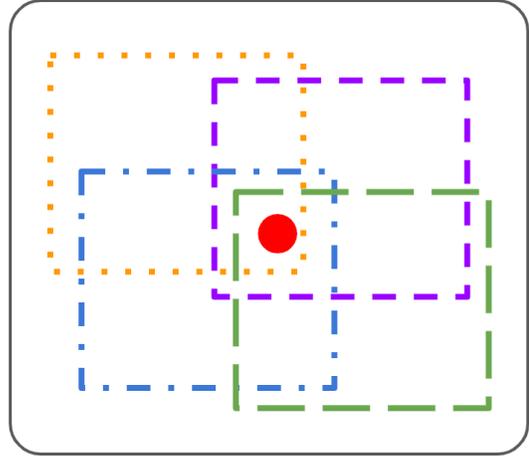


Fig. 3. Examples of possible sampled spatial locations (colored, dashed rectangles) that contain a water flow intensity measurement station (red dot). The random sampling increases the data variability (compared to e.g. always requiring the measurement station to be at the center). Since each possible spatial location has size  $H \times W$ , with  $H = W = 100$  in our setup, the union of all possible such rectangles covers roughly  $200 \times 200$  pixels, which corresponds to a real-world area of size  $2\text{km} \times 2\text{km}$ .

11,000 different temporal inputs per site. Hence, in total there are roughly  $250,000 \cdot 11,000 = 2.75 \cdot 10^9$  spatio-temporal training inputs. Again, however, note that there is an overlap between a large majority of these training inputs, so the training set is effectively much smaller.

For model evaluation (see Section V), the end objective is to minimize the root-mean-square error (RMSE) of the predicted water flow intensities (thus note that the error is measured in  $\text{m}^3/\text{sec}$ ), assessed based on ground truth flow intensities. During training, in order to balance loss smoothness with robustness to outliers, we use the Huber loss:

$$\mathcal{L}(f, f^{\text{gt}}) = \begin{cases} \frac{1}{2} (f - f^{\text{gt}})^2 & \text{if } \|f - f^{\text{gt}}\| \leq \delta \\ \delta (\|f - f^{\text{gt}}\| - \frac{1}{2}\delta) & \text{else} \end{cases} \quad (1)$$

where  $f$  and  $f^{\text{gt}}$  denote predicted and ground truth flow intensity, respectively. We set  $\delta = 1$  by default, as it is shown to result in the best performance (see Section V, where we also compare with other loss functions). The Huber loss can be seen as a combination of the commonly used MSE- and L1-losses, where the MSE-loss is applied when the error is smaller than the threshold  $\delta$ , and the L1-loss is applied otherwise.

Note that for each predicted flow map  $\mathbf{F}$ , the loss (1) is only given for an extremely sparse set of coordinates (most commonly in a single point). This is because in the ground truth flow map  $\mathbf{F}^{\text{gt}}$ , we only have access to water flow measurements in very few coordinates (since the measurement sites are so sparsely located in the data). Despite this extreme loss sparsity, we show in Section V that the model generalizes well to unseen data. This finding is in line with earlier works that have shown that it is possible to train semantic segmentation models from extremely few annotated pixels [16].

<sup>2</sup>We use the FCN8 model from the open-source FCN library [20].

TABLE I

EXPERIMENTAL RESULTS ON THE VALIDATION SET FOR OUR MAIN MODEL, ITS ABLATED VARIANTS, AND BASELINES. WE REPORT THE ROOT-MEAN-SQUARE ERROR (RMSE; LOWER IS BETTER). COLUMN 1 REPRESENTS OUR MAIN FCN MODEL, CF. SECTION IV. COLUMNS 2-4 REPRESENT MODEL VARIANTS WHICH OMIT SOME OF THE SPATIAL INPUT LAYERS. COLUMNS 4-6 REPRESENT VARIANTS WHICH OMIT SOME TEMPORAL INPUTS. COLUMNS 7-8 REPRESENT BASELINE METHODS AGAINST WHICH TO COMPARE THE RESULTS IN COLUMNS 1-5. NOTE THAT *PREVIOUS FLOW* LEVERAGES PAST WATER FLOW INTENSITY INFORMATION THAT IS UNAVAILABLE TO THE OTHER APPROACHES.

Main model	No-elev	Only-elev	No-soil	No-temp	No-rain	Half-time-hist	Mean-per-site	Previous flow
1.35	4.74	3.22	4.49	1.91	5.91	1.40	2.05	0.59

For model parameter optimization, we use Adam [9] with batch size 64 and learning rate  $2 \cdot 10^{-4}$ . The model is trained for 250,000 batches, which takes about 48 hours on the Titan V100 work station that is used for experimentation. To improve model generalization towards unseen data, we resort to the customary deep learning training technique of augmenting the data by horizontal and vertical flips of the inputs (an independent probability of 50% per flip).

## V. EXPERIMENTS

In this section we present the results of our empirical model evaluations on the validation set. We first describe the various baselines and model variants in Section V-A. Then, in Section V-B, the empirical results are presented.

### A. Baselines and Model Variants

We compare our main model described in Section IV against the following baselines:

- **Mean-per-site:** For each water flow intensity time series  $\mathbf{f}^i = \{f_j^i\}_{j=t_1}^{j=t_{N^i}}$ , where  $i$  indexes the  $i$ :th spatial location for a water flow measurement site, we return the mean  $\hat{f}^i$  and use that as the predicted water flow intensity at time  $t$  (for each day  $t$ ) at the  $i$ :th site. Note that this provides the optimal prediction (in terms of RMSE) in case only spatial information would be used as model input.
- **Previous flow:** Provides  $f_{t-1}$  as the predicted water flow intensity at time  $t$ . Note that this baseline leverages information that our model does not have access to; our model only obtains past rainfall and temperature information, not past water flow intensities. Thus *previous flow* can be regarded as a proxy for an upper bound in terms of model performance.

We also train and evaluate the following variants of our proposed ML model:

- **No-elev:** Omits the elevation and terrain slope maps from the set of spatial input maps.
- **Only-elev:** For the spatial part of the input, this model only uses the elevation and terrain slope maps. It omits the other spatial input layers.
- **No-soil:** Omits the soil information spatial layers (soil type, soil moisture, soil depth, land cover) from the set of spatial input maps.
- **Half-time-history:** Uses temporal information from the past  $T = 10$  days (instead of  $T = 20$  as is default).
- **No-temp:** Omits temperature information as an input.
- **No-rain:** Omits rainfall information as an input.

- **Flow-(t-k):** In addition to all the spatial and temporal inputs of our main model, this model has water flow intensity information  $f_{t-T-k+1}, f_{t-T-k+2}, \dots, f_{t-k}$  as an additional temporal input when predicting the water flow intensity  $f_t$  at time  $t$ . We train and evaluate models with  $k \in \{1, 2, 3\}$ , i.e. models that have temporal information up to between three and one day prior to the day for which flow intensities are predicted.

Finally, we also train and evaluate the effect of variations to the main model architecture (cf. Section IV):

- **Alt-rain-temp:** Uses a more efficient temporal input representation, which results in the input  $\mathbf{I}$  having dimension  $H \times W \times (C+T)$  instead of  $H \times W \times (C+2T)$ . This is achieved by having two unique values per temporal layer (instead of only one), where every second element (spatially) is a rainfall measurement, and every second element is a temperature measurement. Note that the convolutional filters (even the first one) will have sufficiently receptive fields to observe all relevant temporal inputs in this case as well.
- **FC-early:** Instead of performing a concatenation of the raw temporal data along the channel dimension, this model first processes the temporal inputs through two fully connected (FC) layers, the last of which produces a 20,000-dimensional vector. It then reshapes this vector into size  $H \times W \times C^{\text{temp}} = 100 \times 100 \times 2$  and concatenates with  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$ . This data volume of dimension  $H \times W \times (C + C^{\text{temp}})$  is then run through all the layers of the FCN, as for the main model.
- **FC-mid:** Similar to *FC-early*, this model first processes the temporal inputs through two FC layers, but here the resulting vector has dimension  $2888 = 38 \cdot 38 \cdot 2$ . It then reshapes this vector into size  $H^{\text{mid}} \times W^{\text{mid}} \times C^{\text{temp}} = 38 \times 38 \times 2$ . Different to *FC-early*, this model does not perform the concatenation with the raw spatial data  $\mathbf{I}^{\text{spat}} \in \mathbb{R}^{H \times W \times C}$ ; instead it first processes  $\mathbf{I}^{\text{spat}}$  through the first third of the convolutional layers of the FCN. It then performs the concatenation at this stage of the FCN, followed by joint processing for the remaining two thirds of the network.

### B. Empirical Results

As mentioned in Section IV-A, we randomly set aside 9 of the 12 data locations for training and 3 for validating the models. The results of our experiments on the validation set are shown in Figure 4 and Table I - IV. The evaluation metric that we report is the root-mean-square error (RMSE).

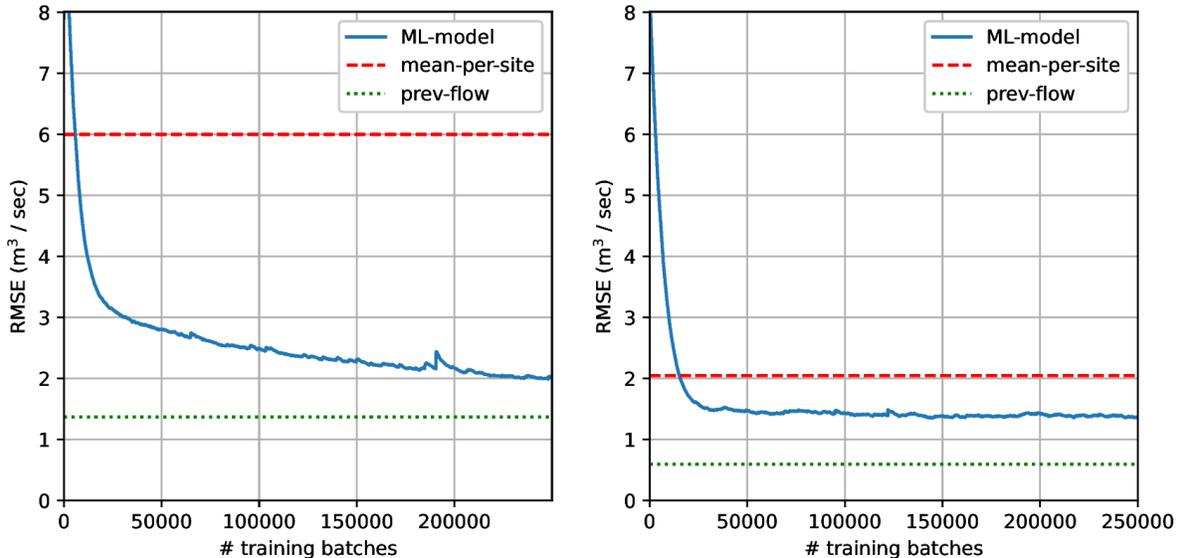


Fig. 4. Training (left) and validation (right) RMSE curves during model training for our main ML model (FCN) described in Section IV. It can be seen that the validation RMSE curve of our model flattens (marginally decreases) throughout training, i.e. the model does not begin to overfit on the training data despite the relatively small size of the training set. *Mean-per-site* and *previous flow* represent baselines, of which *previous flow* can be seen as an oracle that leverages past water flow intensity information that is unavailable to our model.

TABLE II

COMPARISON TO MODELS WHICH OBTAIN PAST WATER FLOW INTENSITIES AS INPUT. COLUMN 1 IS OUR MAIN MODEL THAT DOES NOT OBTAIN PREVIOUS FLOWS AS INPUT. PROVIDING PAST FLOW INFORMATION IMPROVES PREDICTION ACCURACY SIGNIFICANTLY, BUT NOTE THAT IN MANY CASES SUCH INFORMATION IS NOT AVAILABLE.

No flow	Flow-(t-3)	Flow-(t-2)	Flow-(t-1)
1.35	0.94	0.80	0.63

Due to the small size of the overall dataset (12 distinct locations), we have not yet considered a proper train-val-test split of the data. This will be done once more data of the appropriate type has been acquired. Currently however, when comparing other model variants and baselines to our main model, we report in the tables the *best* validation RMSE that was obtained during training of the respective model variant. Different to many of the alternative approaches, however, our main model’s RMSE on the validation set monotonically improves throughout training (see Figure 4), and hence the results of the main model have not been ‘cherry picked’ at a certain optimal iteration number based on the validation set. Thus any reported improvements of the main model relative to alternatives may in fact be larger if assessed on a withheld test set.

**Main results.** It can be seen in Table I that our main model outperforms its ablated variants *no-elev*, *only-elev*, *no-soil*, *no-temp* and *no-rain*. In particular, the elevation and terrain slope maps are crucial, as is past rainfall information. Past temperature information is not as important, but omitting it still results in a higher error. Using rain and temperature information from the past  $T = 10$  (instead of  $T = 20$ ; see *half-time-hist*) days leads to similar results for this data.

Furthermore, our main FCN method is significantly better than the *mean-per-site* baseline, which indicates that our model has learnt to properly leverage spatio-temporal information. Our approach does however not outperform *previous flow*, which is a very strong baseline that leverages past water flow intensity information. Since such information is often hard to come by in practical scenarios, we have opted for a model that does not require past flow intensities as input, since it makes the model much more broadly applicable. Model variants which obtain previous water flow intensity information are however evaluated in Table II.

In Figure 4 we show training and validation RMSE curves during model training for our main FCN model. Note that the validation RMSE curve flattens (marginally decreases) throughout training, i.e. the model does not begin to overfit on the training data despite the relatively small size of the training set.

**Effect of providing previous water flow intensity information as model input.** As seen in Table II, models that receive flow information for  $T = 20$  past consecutive days until three (*flow-(t-3)*), two (*flow-(t-2)*), or one (*flow-(t-1)*) day before the prediction day are significantly more accurate at predicting water flow intensity. Note however that in many practical scenarios such information is not available.

**Effect of loss function.** In Table III we compare the effect of using different loss functions during training; cf. (1). It is clear that the Huber loss (with  $\delta = 1.0$  or  $\delta = 1.1$ ) yields the best results, whereas the L1-loss results in the worst results.

**Effect of model architecture.** In Table IV we compare the effect of using different model architectures. The more

TABLE III

LOSS ANALYSIS. THE HUBER LOSS YIELDS THE LOWEST RMSE, WITH  $\delta = 1.0$  AND  $\delta = 1.1$  BEING BEST. THE HUBER LOSS OUTPERFORMS THE MSE LOSS, AND THE L1 LOSS YIELDS POOR RESULTS.

Huber-1.0	MSE	L1	Huber-0.8	Huber-1.1
1.35	1.55	3.17	1.47	1.35

TABLE IV

MODEL ARCHITECTURE COMPARISONS. THE *FC-EARLY* AND *FC-MID* ARCHITECTURES YIELD SIGNIFICANTLY WORSE RESULTS THAN THE MAIN MODEL AND THE *ALT-RAIN-TEMP* ARCHITECTURE.

Main model	Alt-rain-temp	FC-early	FC-mid
1.35	1.47	2.57	2.49

efficient *alt-rain-temp* architecture yields almost as good results as our main architecture, so it would be suitable to consider if compute is a limiting factor. The *FC-early* and *FC-mid* architectures yield significantly worse results.

**Qualitative examples.** Several qualitative examples for our main model are shown in Figure 5 - 6. Different to the quantitative experimental results above – which are only performed for the sparse set of spatial ground truth locations that are available in the data<sup>3</sup> – these qualitative examples shed more light into the full spatial extents of the model predictions. For example, as can be expected, higher water flow intensities are typically predicted where the terrain slope is high.

## VI. DISCUSSION AND CONCLUSIONS

In this work we have introduced a fully convolutional approach for dense water flow intensity prediction in catchment areas. Our specific results were shown for Swedish basins, but the general methodology is expected to be transferable to other geographical regions.

The proposed model is able to learn and generalize from a limited training dataset. In this work, we have used training data from merely 25 measurement points (28 in total; 3 were used for evaluation). The fact that we obtain such high performance may be attributed to the training setup. In particular, the model generalization is alleviated by the fact that the model sees many slight variations of each measurement site during training, since there are many ways to select a viewpoint around a given measurement site (cf. Figure 3). To the best of our knowledge, this is the first work which models water flow intensity using a fully convolutional neural network, which allows us to provide *dense* flow predictions – in effect, we predict one flow intensity per coordinate, even though we only have annotations for 28 specific coordinates.

Since our main FCN method is significantly better than the *mean-per-site* baseline, we conclude that our model has

<sup>3</sup>One can however argue that a form of (semi-)dense evaluation is being performed also in the quantitative results, since we vary the spatial coordinates of the ground truth flow intensity measurement stations in each spatial input example (cf. Figure 3). Note that the model is never aware of the spatial coordinates in which it is being evaluated.

learnt to properly leverage spatio-temporal information. Our approach does however not outperform the *previous flow* baseline, which is a very strong baseline that leverages past water flow intensity information. Such information is not available to our model, and is often hard to come by in practical scenarios. As can be seen in Table II, the FCN model variants which utilize past flow information also obtain better results. A potential avenue of future work is thus to consider our setup through a privileged learning lens, wherein flow information could be leveraged during training, but where the model must perform inference using only rainfall and temperature information (in addition to spatial information).

We hope that our work will serve as a solid stepping stone and an inspiration for further research within dense water flow modeling, which in turn could deliver useful information when it comes to future climate adaptation planning (e.g. within flood risk management) in Sweden and beyond.

## REFERENCES

- [1] Claes Bernes. En varmare värld: Växthuseffekten och klimatets förändringar-tredje upplagan, 2017.
- [2] Brian Cosgrove, David Gochis, Edward P Clark, Zhengtao Cui, Aubrey L Dugger, Greg M Fall, Xia Feng, Mark A Fresch, Jonathan J Gourley, Sadiq Khan, et al. Hydrologic modeling at the national water center: Operational implementation of the wrf-hydro model to support national weather service hydrology. In *AGU Fall Meeting Abstracts*, volume 2015, pages H53A–1649, 2015.
- [3] Shiheng Duan, Paul Ullrich, and Lele Shu. Using convolutional neural networks for streamflow projection in california. *Frontiers in Water*, 2:28, 2020.
- [4] Jonathan M Frame, Frederik Kratzert, Daniel Klotz, Martin Gauch, Guy Shelev, Oren Gilon, Logan M Qualls, Hoshin V Gupta, and Grey S Nearing. Deep learning rainfall–runoff predictions of extreme events. *Hydrology and Earth System Sciences*, 26(13):3377–3392, 2022.
- [5] Jonathan M Frame, Frederik Kratzert, Austin Raney, Mashrekur Rahman, Fernando R Salas, and Grey S Nearing. Post-processing the national water model with long short-term memory networks for streamflow predictions and model diagnostics. *JAWRA Journal of the American Water Resources Association*, 57(6):885–905, 2021.
- [6] Martin Gauch, Juliane Mai, and Jimmy Lin. The proper care and feeding of camels: How limited training data affects streamflow prediction. *Environmental Modelling & Software*, 135:104926, 2021.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [8] Xiaowei Jia, Jacob Zwart, Jeffrey Sadler, Alison Appling, Samantha Oliver, Steven Markstrom, Jared Willard, Shaoming Xu, Michael Steinbach, Jordan Read, et al. Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 612–620. SIAM, 2021.
- [9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [10] Frederik Kratzert, Daniel Klotz, Claire Brenner, Karsten Schulz, and Mathew Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018.
- [11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [12] Karin Mossberg Sonnek, Annika Carlsson-Kanyama, and Carl Denward. Värmens påverkan på samhället–en kunskapsöversikt för kommuner med faktablad och rekommendationer vid värmebölja. *Myndigheten för samhällsskydd och beredskap, Report No.: MSB870*, 2015.

- [13] Ana Ramos Oliveira, Tiago Brito Ramos, and Ramiro Neves. Stream-flow estimation in a mediterranean watershed using neural network models: A detailed description of the implementation and optimization. *Water*, 15(5):947, 2023.
- [14] Tobias Salmonsson. Assessing the impacts of climate change on runoff along a climatic gradient of sweden using persist, slu masters thesis, 2013, 2014.
- [15] Lisbeth Schultze, Carina Keskitalo, Irene Bohman, Robert Johansson, Erik Kjellström, Henrik Larsson, Elisabet Lindgren, Sofie Storbjörk, and Gregor Vulturius. National expert council for climate adaptation assessment report nr 1. 2022.
- [16] Gyungin Shin, Weidi Xie, and Samuel Albanie. All you need are a few pixels: semantic segmentation with pixelpick. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1687–1697, 2021.
- [17] Wen-Ping Tsai, Dapeng Feng, Ming Pan, Hylke Beck, Kathryn Lawson, Yuan Yang, Jiangtao Liu, and Chaopeng Shen. From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nature communications*, 12(1):5988, 2021.
- [18] Song Pham Van, Hoang Minh Le, Dat Vi Thanh, Thanh Duc Dang, Ho Huu Loc, and Duong Tran Anh. Deep learning convolutional neural network in rainfall–runoff modelling. *Journal of Hydroinformatics*, 22(3):541–561, 2020.
- [19] Charuleka Varadharajan, Alison P Appling, Bhavna Arora, Danielle S Christianson, Valerie C Hendrix, Vipin Kumar, Aranildo R Lima, Juliane Müller, Samantha Oliver, Mohammed Ombadi, et al. Can machine learning accelerate process understanding and decision-relevant predictions of river water quality? *Hydrological Processes*, 36(4):e14565, 2022.
- [20] Ketaro Wada. pytorch-fcn: PyTorch Implementation of Fully Convolutional Networks. <https://github.com/wkentaro/pytorch-fcn>, 2017.
- [21] Andreas Wunsch, Tanja Liesch, and Stefan Broda. Groundwater level forecasting with artificial neural networks: a comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX). *Hydrology and Earth System Sciences*, 25(3):1671–1687, 2021.

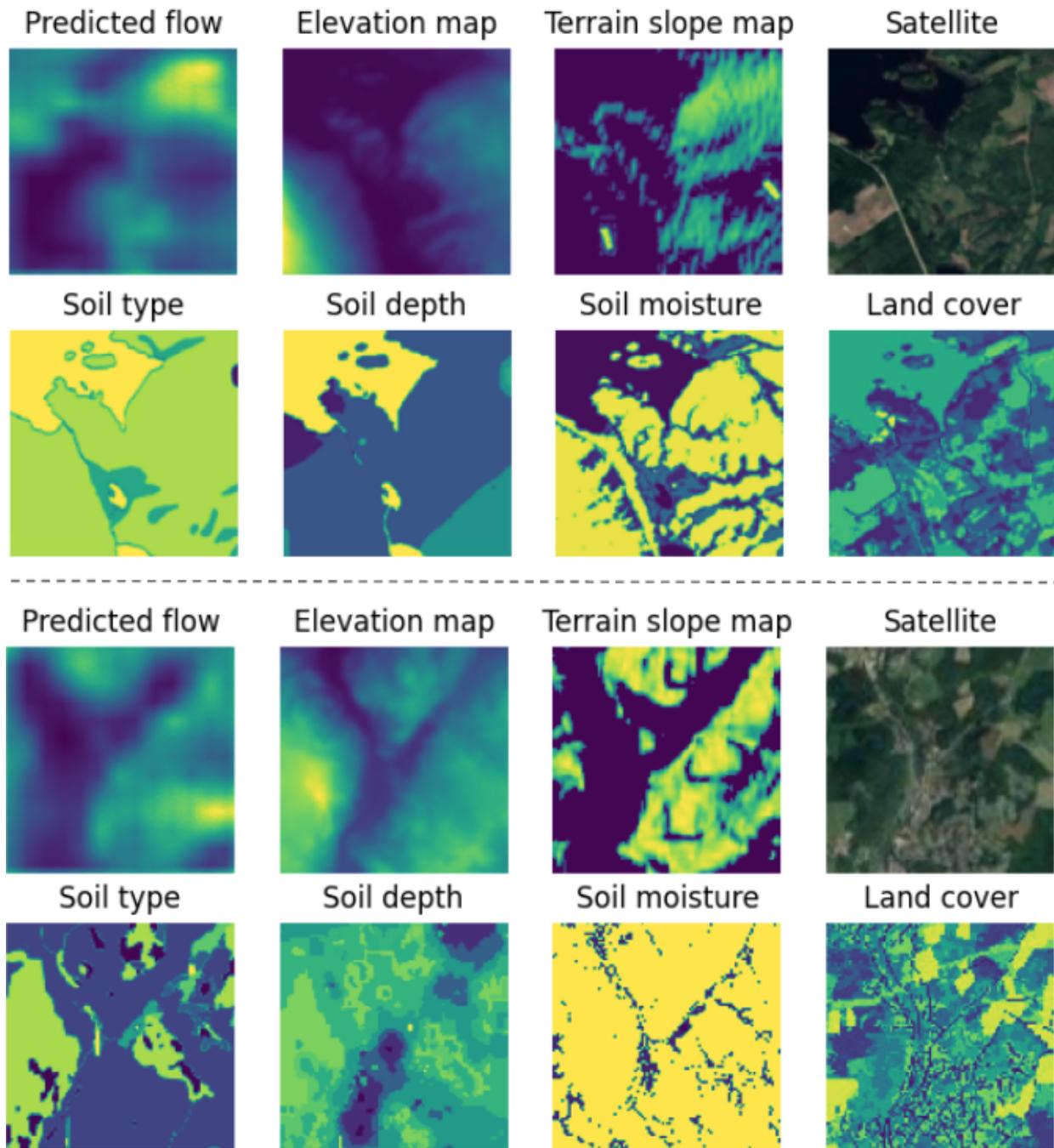


Fig. 5. Two qualitative examples for our main model on the validation set (examples differentiated by the dashed horizontal line). In each example, the top-left image represents the predicted water flow intensities at the given area; darker blue means lower intensity, while brighter yellow means higher intensity. The other seven images represent various spatial input layers to the FCN model. For all images except the satellite image, the maximum color intensity is individually normalized so that variations within images become as visible as possible. As can be expected, in both examples, higher flow intensities are typically predicted where the terrain slope is higher. In the example above the dashed line, the model also predicts relatively high flow intensities on the lake that can be observed at the top-left of the satellite image. Note however that the training set contains no ground truth water flow intensities on lakes, and thus the model has never been able to adapt to what is reasonable in terms of flow intensity on lakes.

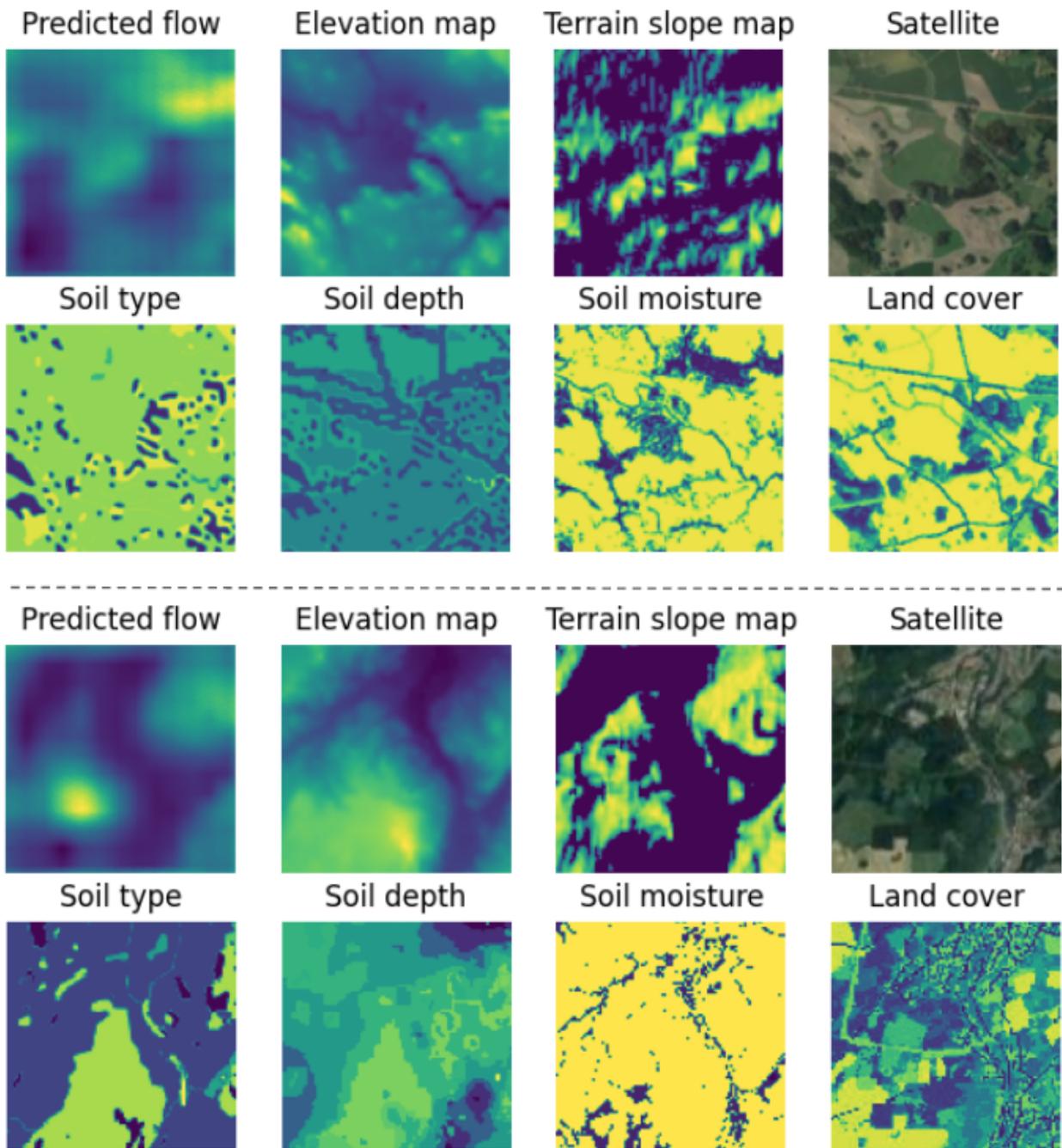


Fig. 6. Two additional qualitative examples for our main model on the validation set (examples differentiated by the dashed horizontal line). In the example above the dashed line, the predicted flow is moderately high within most of the map. A peak in terms of predicted flow can be seen at a corresponding peak within the terrain slope map. In the example below the dashed line, it can again be seen that the predicted flow is typically relatively higher where the terrain slope map is higher