PeakWeather temperature forecasting

GDL project - Fall 2025

This project investigates the potential of spatiotemporal graph neural networks (STGNNs) to enhance the accuracy of numerical weather forecasts across Switzerland. The work is in collaboration with the Federal Office of Meteorology and Climatology (MeteoSwiss), which provides access to meteorological observations from the SwissMetNet station network and numerical weather predictions generated by one of the operational physics-based forecasting systems. The project involves implementing deep spatiotemporal learning architectures and empirically evaluating their performance to assess the improvements these models can offer over traditional forecasting approaches.

Project description

Have a look at the PeakWeather dataset released on [HuggingFace](), which is documented in preprint paper https://arxiv.org/abs/2506.13652. It introduces a dataset with several meteorological variables measured from different weather stations across Switzerland and baseline models to address forecasting tasks. In this project we'll consider temperature forecasting, a relatively well-behaved variable with clear seasonal patterns and significant real-world applications.

The PeakWeather dataset provides temperature measurements from two types of meteorological stations, referred to as weather stations and rain gauges (see the paper for a detailed description). Weather stations provide better quality temperature sensors and a larger suite of sensors, whereas rain gauges are equipped with pluviometers and supporting temperature sensors, with the latter being primarily used to differentiate between solid and liquid precipitations. We ask you to consider data from the weather stations first. At a second stage, we ask you to work with both types of sensors.

The goal of the project is to assess (i) the effectiveness of spatio-temporal graph neural networks to forecast temperature from ground stations, and (ii) whether temperature forecast accuracy can improve from considering additional temperature observations, although of a lower quality.

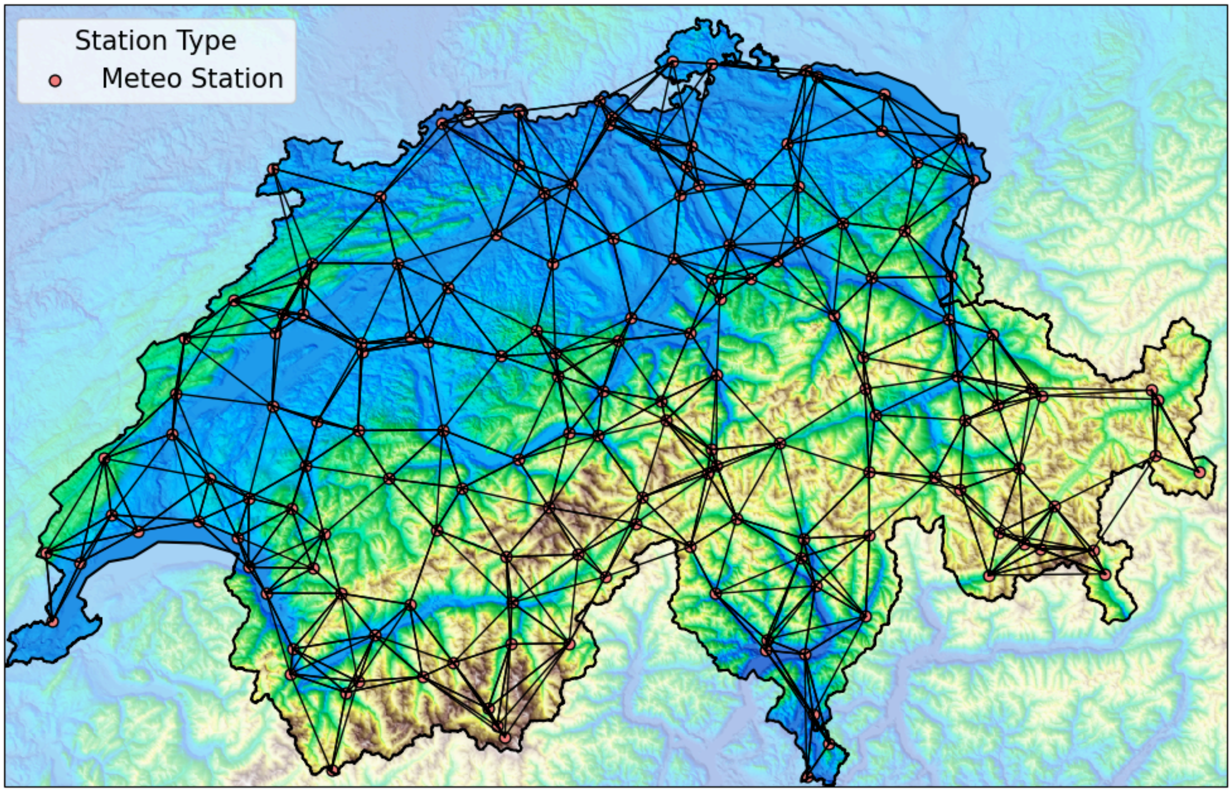

Fig.1: The SwissMetNet network of weather stations with a graph based on distance.

Project instructions

To carry out your project, refer to the following resources:

- PeakWeather paper - provides all information you might need to understand the problem and the data.
- PeakWeather library and its documentation - provides utilities to download and load the data in ML-ready formats,
- peakweather-wind-forecasting repository - implements a use case in wind forecasting (described in the paper) alongside baseline models you can reuse.

The following are your mandatory tasks:

Deliverable 1 - implementation

Provide working implementations of the following models on the provided data:

- **Model0**: Implement a baseline that does not use graphs, such as an RNN and/or TCN. (it must differ from those implemented in the repo)
- **Model1** (baseline RNN): Check and - if needed - modify the RNN architecture defined by class RNN with embeddings, and motivate your changes.

- **Model2** (baseline STGNN): Check and - if needed - modify the STGNN architecture defined by class STGNN, and motivate your changes.
- **Model3**: Implement a competitive STGNN supported by the literature. Favor simple and motivated ideas over convoluted architectures.

Deliverable 2 - performance assessment

Assess the performance of all models on both MAE and CRPS (EnergyScore in the paper) similarly to what was done in the paper. Alongside your models' performance metrics report the NWP performance as reference. As NWP forecasts come in 11 samples, you can use the ensemble median for the MAE (you can look at SampleMAE metric) and the EnergyScore metric for the CRPS (make sure to keep the feature dimension to 1). Since the performance of the NWP model declines with increasing lead time, the CRPS and MAE should also be specifically reported at t = 1, t = 3, t = 6, t = 12, t = 18, t = 24. You are expected to perform some tuning of your model and training hyperparameters, such as type and number of layers, number of hidden units, dropout probabilities, learning rate, and scheduler algorithm. At this stage consider only the weather stations, and no rain gauges (see the station_type argument in the dataset constructor). You can contact us for feedback before running all training processes. Model performance should be assessed only on the best configurations found on the validation MAE for each architecture. The four selected models should be trained and tested starting from 5 different seeds. Let the test set start from April 1st, 2024.

A few remarks:

- To compare with the NWP forecasts a subset of the test set must be considered to accommodate the fact that forecasts are issued every three hours (please refer to the paper for detailed explanation).
- Each station is equipped with up to 8 types of sensors. Please use all variables to make temperature predictions. Note also that not all variables are available at every time step. A mask of missing observations is provided with the data.
- Have a look at the repository peakweather-wind-forecasting, which shows ways to deal with above points.

Deliverable 3 - adding low-quality sensors

Now include the rain gauges in the pipeline. Note that they are equipped with lower grade temperature sensors.

- Train on all nodes, but test on weather stations only. Does forecasting of the station temperature improve when using rain gauges as additional input information?
- Train and test on all nodes. Compute the metrics on rain gauges and weather stations separately. Do you see any evident difference in the prediction error on the two types of nodes?

Bonus task 1

Consider the provided graph and try answering the following questions:

- How sensitive model performances are to the graph hyperparameters? Discuss possible reasons.
- Construct a graph from a different heuristic. Does model performance change sensibly?

Bonus task 2

Implement or design a new model that learns a graph alongside model parameters. Try to describe the graphs that result from learning. Do they make any sense to the human eye? Are the learned relations consistent over different training runs?

Bonus task 3

Fetch some additional weather variables from other sources that you deem relevant. Train a model accounting for them and improve, if you can, the prediction performance of the same model without that extra data.

Bonus task 4

Compare the performance of one or more foundation time series models of your choice with your models. Do your best to make the comparison fair. Discuss advantages and limitations.

For any questions, refer to Michele Cattaneo (michele.cattaneo@meteoswiss.ch) or Daniele Zambon (daniele.zambon@usi.ch).