Prediction and Visualization of Meteorological Data from Central Europe

Petr Košťál

CTU–FIT

May 4, 2025

1 Introduction

The purpose of this work is to create a model to refine the prediction of meteorological data based on the GFS[1] and data measured at meteorological stations. GFS is a numerical model for global weather prediction with a resolution of 13 km, updated every six hours. The computed values at the station locations can then be extrapolated to provide forecasts for the entire region. For visualizing the meteorological data, I created a simple web application using Dash[6].

2 Input data

I received the data from the Data Laboratory (DataLab). They contain a total of 213 meteorological stations in the Czech Republic and Slovakia. However, it is problematic that many values are missing and that each station measures different quantities at different times. Occasionally, some sensors also return false values due to malfunctions. Therefore, I aggregated the data and selected the best features and stations for prediction, which will allow us to evaluate the accuracy of individual models as objectively as possible. When selecting, I also had to assess the suitability of the data for the respective models. For this reason, I had to exclude precipitation, for example, because it is mostly zero, which is problematic for many models. Finally, I selected *air temperature*, *ground temperature* and *relative humidity* for prediction. Most stations measure these quantities, and they do not contain many erroneous values.

3 Problem definition

Individual measurements and forecasts take place always six hours apart from each other. Our goal will therefore be to predict the weather in a time of four steps (24 hours) from the current measurement.

[1]Global Forecast System

3.1 Evaluation metrics

Although the amount of data obtained was not very large, I divided it into training and validation sets. For comparison between individual models, I used RMSE[2] in the validation set.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

For each quantity and station I calculated MAE[3] and MSE[4] on the validation set for a more appropriate interpretation of error rates. It is therefore possible to compare not only the overall accuracy, but also the accuracy of the respective model for a specific quantity or specific station. We can compare the models with each other as well as with the reference GFS from which we start.

3.2 Regression model

When creating our own model, the starting points were always the latest measured values and the GFS forecasts. Several models and optimization techniques were tested as part of the work. First, I tried CatBoost[2] in combination with the Optuna[3] optimization tool. I also tried the machine learning libraries AutoGluon[1] and PyCaret[4], which effectively allowed me to try many different models and their combinations. Last but not least, I also trained a simple MLP[5].

3.3 Time series prediction

When predicting a time series, I always used the last four steps to predict the next four steps. I trained the model to predict the entire time series. However, only the output at the time of 24 hours was again used for validation. I tried prediction using the recurrent neural network LSTM[6]. Finally, I tried the Mamba[5] model, which can also be applied to time series prediction.

[2]Root Mean Square Error
[3]Mean Absolute Error
[4]Mean Square Error
[5]Multilayer perceptron
[6]Long short-term memory

3.4 Ensemble models

Finally, I tried both approaches together, first creating a model for time series prediction and then training a model that attempts to refine the output from this time series. I tried combinations of LSTM and CatBoost.

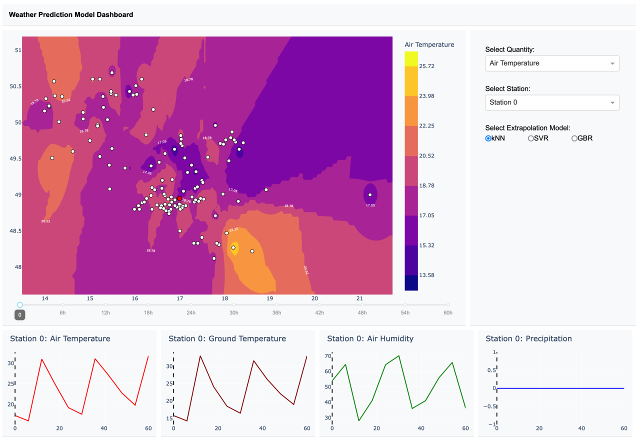

Figure 1: Web application created in Dash

4 Visualization tool

I created a simple visualization tool (figure no.1), which allows data to be displayed clearly. It is a configurable web application that only needs to be presented with data in the required format and the configuration appropriately adjusted. The tool allows data to be displayed in individual time periods and for individual meteorological stations. It is also possible to choose the method of data extrapolation; in the default case, we use a regressor using the nearest neighbor method. The configuration file already contains defined suitable parameters for the demonstration dataset.

5 Results

According to the RMSE metric on the validation set (figure no. 2), the CatBoost model, implemented as a regression model, proved to be the most accurate. Across all quantities (figure no. 3), the prediction is on average more accurate than the reference GFS. The second most accurate model is AutoGluon, which also consistently increases forecast accuracy. The third most accurate model is the ensemble of LSTM and CatBoost, which on average performs better than GFS, but has a higher average loss for ground temperature. Other models tend to perform worse compared to the reference.

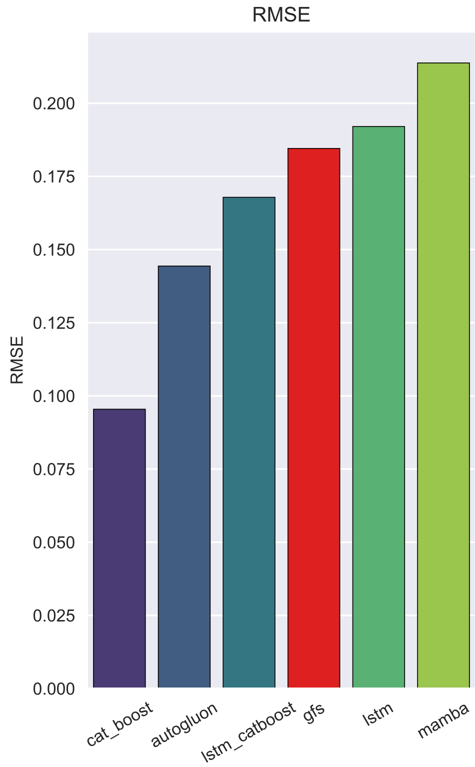

Figure 2: Average RMSE loss on the validation set for normalized data

6 Conclusion

As part of this semester project, I succeeded in creating a predictive model that improves the accuracy of weather forecasts for specific variables. Although I was significantly limited by computational capacity—which heavily penalized some models—I am satisfied with the result. I enjoyed working on this topic and had the opportunity to try out many new machine learning methods.

References

[1] Autogluon: Automl for text, image, and tabular data. online. [cit. 2025–11–1] `https://auto.gluon.ai/stable/index.html`.

[2] Catboost: a library for gradient boosting on decision trees. online. [cit. 2025–11–1] `https://catboost.ai/docs/en/`.

[3] Optuna: A hyperparameter optimization framework. online. [cit. 2025–11–1] `https://optuna.readthedocs.io/en/stable/`.

[4] Pycaret: Low-code machine learning library in python. online. [cit. 2025–11–1] `https://pycaret.gitbook.io/docs`.

[5] Tri Dao Albert Gu. Mamba: Linear-time sequence modeling with selective state spaces.

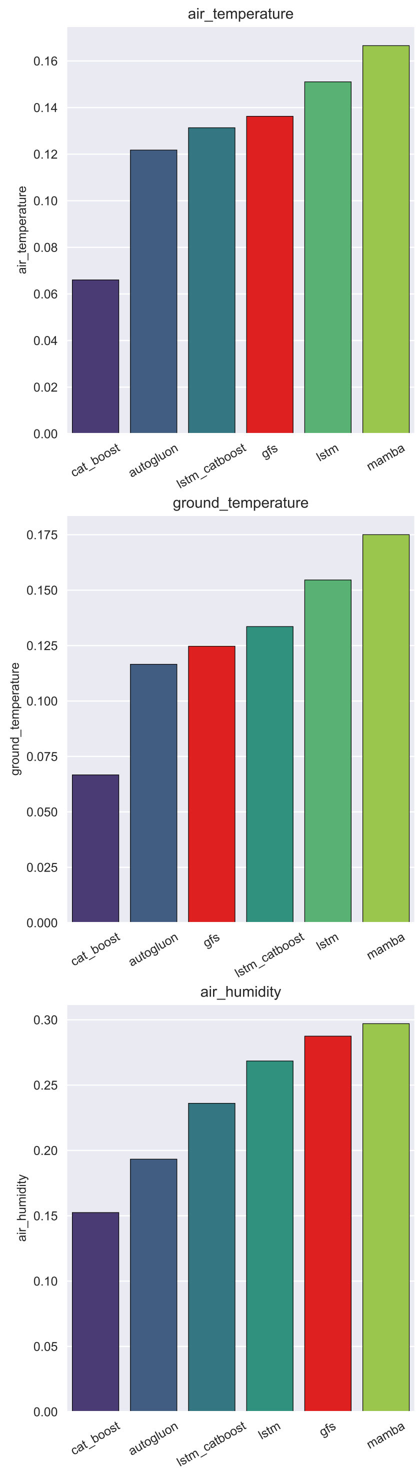

Figure 3: Average RMSE loss on the validation set for normalized data for individual quantities

online, 2023. [cit. 2025–11–1] `https://arxiv.org/abs/2312.00752`.

[6] Plotly. Dash by plotly. online. [cit. 2025–11–1] `https://dash.plotly.com/`.