Renewable energy supply forecasting with Support Vector Regression

Vera Milovanović
Student ID: 1234
MSc Machine Learning, University of Tübingen

Abstract

The rise in renewable energy production requires a transformation of the current electrical grid. However, the unpredictable nature of wind and solar power complicates their integration. Short-term (up to 24 hours) supply forecasting is crucial for maintaining grid stability and better electricity market trading. This seminar paper assesses the Support Vector Regression (SVR) method for forecasting solar, on-shore, and off-shore power supplies 1 and 24 hours in advance. This is done based on a dataset of weather forecasts and historic load and supply of renewables data. It is cured from large-scale raw data, such that it fits the method better. It turns out that such a complex problem cannot be solved to a satisfactory level using an "off-shelf" SVR model.

1 Introduction

An integration of renewable energy into the existing grid has brought about significant changes in the electricity sector. Unlike conventional power plants, wind and solar energy sources are inherently variable and less predictable. This volatility poses a significant challenge to both grid operators and energy generators. On one hand, operators must maintain a constant balance between electricity supply and demand to ensure grid stability and reliability. On the other hand, renewable power generators have to better estimate the generation and bid accordingly in intraday or day-ahead markets (International Renewable Energy Agency (IRENA), 2020).
To address these challenges, one could use some well-studied statistical models, like Support Vector Regression (Boser et al., 1992, Schölkopf and Smola, 2005) for renewables forecast. Although there is a computational overhead for datasets with roughly more than 10 000 samples (*SVR API Reference* 2012), it can be worth the effort to apply this method, given that the training process is relatively simpler than the current state-of-the-art methods that are based on Artificial Neural Networks (ANNs). This is because SVR solves a convex quadratic optimization problem, for which there are many solvers that find a global optimum, given the adequate implementation and a set of hyperparameters. For example, one can use Sequential Minimal Optimization (SMO) (Platt, 1998), which is the one that is used in Scikit-learn SVR implementation *SVR API Reference* 2012, that is used in this seminar paper.
In this work, the time series adapted SVR model is made to predict a multi-step prediction (1 and 24 hours ahead), given the fixed amount of historical data. The dataset encompasses weather information, along with energy production and consumption data in Germany. The contribution of this seminar paper is to adapt and evaluate the SVR model on a time series of energy data, specific to the German market. A code to reproduce experiments is available at `https://github.com/nilskiKonjIzDunava/Renenewable_energy_forecasting/tree/main`.

Seminar Machine Learning for Renewable Energy (Summer Term 2024), Tübingen.

2 Related Work

Support Vector Regression is a statistical learning method that was developed under the series of work on Support Vector Machines (SVMs) (Boser et al., 1992). Different variations of SVMs can be used in classification (Boser et al., 1992), regression (Drucker et al., 1996) and clustering (Ben-Hur et al., 2001) tasks. Although SVM was not originally developed for time series forecasting, subsequent work (Rüping, 2001) adapted it, discussing some design choices specific to time series data. There, the main emphasis was on kernel functions, outlining that the Radial Basis Kernel (RBF), given in Equation 2 was a very good starting point for multivariate time series problems. Therefore, a proposed method in this work also uses the RBF kernel.
When it comes to forecasting a solar power supply using SVR, some works use only weather-related data and try to predict solar power supply (Abuella and Chowdhury, 2017). There, the authors explored how feature selection affects the forecast outcomes. However, since the selection was not done using statistical techniques, the proposed pipeline requires either some expert knowledge or considerable compute and time budgets. In contrast, this work aims to forecast renewable energy supplies by not integrating any prior knowledge about features. It can be justified by the fact that SVR is performing well in high-dimensional spaces and that some information might be in the features that are sometimes overlooked by the domain experts. For the offshore wind power supply forecast, a notable work (Kramer et al., 2013) that evaluated SVR used only wind data and explored the influence of the time horizon. More specifically, how far we can look into the future, and how much information from the past is necessary for a reliable forecast. They found that a reliable forecast at the level of wind grid points was possible for the 20-minute and 2-hour ahead prediction. At the level of a whole wind park, the results showed that even a reasonable six-hour forecast was possible. In contrast, the setting in this work is arguably more difficult, as the supply data is available only at the scale of all wind turbines in Germany, whereas the weather data is available for fine-grained locations on a map.

3 Method

3.1 Basic Idea and Optimization

The goal of regression is to approximate a function $f(x)$ from a given noisy data set $\{(X_i, Y_i)\}_{i=1}^{N}$. The basic idea of SVR is to first implicitly embed the input data into a high-dimensional feature space using a kernel function $k(x_i, X_j) = \langle \Phi(x_i), \Phi(X_j) \rangle$ via a non-linear mapping $\Phi(\cdot)$. Then, a liner regression is used to estimate the function $f(x)$ in the new feature space:

$$f(x) = \sum_i \beta_i k(x, X_i) + b, \tag{1}$$

where x is a point to embed, and $X_i, i = 1, ..., N$ are all other points in the training set. In this way, kernel methods can effectively learn non-linear functions by projecting data to a new high-dimensional space where a linear method could solve the task at hand. To address the high computational costs in high-dimensional spaces, kernel methods utilize what is known as the "kernel trick." This trick involves representing the data solely through a set of pairwise similarity comparisons between the original data observations x (in their original lower-dimensional coordinates), rather than explicitly transforming the data using $\Phi(x)$ and representing it by these transformed coordinates in the higher-dimensional feature space.
The most commonly used kernel is Radial Basis Kernel (RBF):

$$k(x_i, X_j) = exp(-\gamma \left\| X_i - x_j \right\|^2), \gamma > 0. \tag{2}$$

In order to estimate the function $f(x)$, a loss function called Vapnik's ϵ-insensitive function is given as:

$$|y - f(x)|_\epsilon = \begin{cases} 0 & \text{if } |y - f(x)| \leq \epsilon, \\ |y - f(x)| - \epsilon & \text{otherwise,} \end{cases} \tag{3}$$

in which ϵ is the tolerance to error and only those deviations larger than ϵ are considered errors (Zhu et al., 2013).
The coefficients $\beta_i \in \mathbb{R}^d, b \in \mathbb{R}$ can be determined from the data by minimizing the primal problem 7, given in Appendix 7.1. The primal problem 7 can be converted into the dual formulation, by using

Lagrange multipliers (Schölkopf and Smola, 2005). In many cases, solving the dual problem is more practical, as primal has $d + 1 + N$ variables and $2N$ constraints, while dual has N variables and $2N + 1$ constraints. The dual problem, given by Equation 4 is used in a popular Scikit-learn SVR library (*SVR API Reference* 2012), which is relevant for this seminar paper.

$$\max_{\alpha \in \mathbb{R}^N} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j \langle X_i, X_i \rangle \tag{4}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{C}{N}, \quad \sum_{i=1}^{N} \alpha_i Y_i = 0, \quad i = 1, ..., N.$$

Note that $\alpha \in \mathbb{R}^N$, whereas $\beta \in \mathbb{R}^d$ (from Equation 1), as the dimensionality of a space where data is projected is the same as the number of data points (N).

3.2 Hyperparameters of SVR

Hyperparameters of the SVR pipeline control the regularization. They are sophistically connected, so making a good combination can be quite challenging.
The most important ones are C, ϵ of SVM, given in Equations 3 - 4, and kernel hyperparameters - γ from the Equation 2. Hyperparameter ϵ influences the width of the margin, and thus regularizes the fitted function. A greater margin corresponds to a more regularized function. Hyperparameter C penalizes slack and can be thought of in terms of its relationship to the margin as $C \sim \frac{1}{\text{margin size}}$.
Hyperparameter γ controls the influence of a point on the fitted function. Smaller γ makes samples far away from the fitted function influence it, leading to a more regularized function.
It turns out that tuning the mentioned hyperparameters is essential to obtain a model that is expressive enough to model the high-frequency signal dynamics. For this seminar project, due to the lack of computational resources, the set of hyperparameters was chosen based on the author's intuition that roots in the hyperparameters' effects stated above.

4 Experiments

4.1 Dataset

Experiments are performed on a dataset that was curated based on multiple other datasets. They incorporate weather, energy (supply, demand, installed capacity), and energy prices data in Germany. In the next subsections, design choices, that are made during a curation of the final dataset, are discussed. They are mainly motivated by the fact that SVR is not suitable for datasets with a large number of samples - training time complexity lies between $\mathcal{O}(dN^2)$ and $\mathcal{O}(dN^3)$ (*SVR API Reference* 2012), where N is a number of samples, d is a number of features.

4.1.1 Weather Data

The weather data represents a forecast, and it is given for the 15 weather variables at 80 locations across Germany, with a resolution of one hour. Weather variables encompass the following measurements: *geopotential, clear-sky direct solar radiation at surface, mean sea level pressure, total cloud cover, u-component of wind at 10m, v-component of wind at 10m, u-component of wind at 100m, v-component of wind at 100m, temperature 2m above ground, surface net solar radiation, top net solar radiation, sunshine duration, forecast surface roughness, total precipitation and boundary layer height*. Weather variables at nearby locations are correlated, and SVR could benefit from removing such features, at least by reducing the time and space complexities. Hence, locations are assigned to a coarse grid of four regions: *top left, top right, bottom left, bottom right*. An illustration of the grouping is given in Figure 4 in Appendix 7.3. Then, data is flattened out on the *location* feature, so the shape is transformed from to a shape $(n_{dataset}, n_{locations} \cdot n_{features})$, where $n_{dataset}$ is number of samples in the original weather dataset, $n_{locations} = 4$, and $n_{features} = 15$ (number of weather variables).
Weather data from Germany used both Central European Time (CET) and Central European Daylight Time (CEDT) time frames, while energy data was in UTC. All data was converted to UTC for

consistency. Duplicate entries caused by daylight saving time were removed, and missing values were interpolated.

4.2 Energy Data

Supply and demand are given for Germany in total, with a time resolution of 15 minutes. Since the time resolution needs to be the same for all data used, the resolution of energy data is decreased to 1 hour, so it matches the weather data. It is done by taking a mean of energy data features within one hour.
Installed capacities data has a time resolution of 1 year, so it is decreased to 1 hour and the forward filling method is performed.

4.2.1 Scaling features

Since SVR is feature scale sensitive, some scaling needs to be done before applying the method, such as the features' ranges are similar. A min-max scaler is used, which puts all features in a range between 0 and 1. It is given by:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{5}$$

where X_{max} and X_{min} are the maximum and the minimum values of the feature, respectively. This is done on all features, except for the time-related features (month, day, hour).
Time stamps are divided into years, months, days, and hours. Month, day, and hour features are encoded by sine/cosine encoding, such that a cyclical nature is enforced.

4.2.2 Data split

Time series data needs some additional pre-processing, such that it can be used to feed the models. Here, a fixed-size rolling window approach is used. It splits time series data into "historic data" within a specified time *window*, and a target which is *horizon* away ahead. To predict a target, the features from a dataset and the previous target values within a window size are used. Such split is done for short-term forecast (1 hour ahead) and long-term forecast (24 hours) ahead. For both horizons, a window size of 3 days is used.
For training, data from 2019 to 2021 is used, and for testing data from 2022. Since hyperparameter optimization is very limited, and some regularization techniques such as early stopping cannot be applied to SVR, a validation set is not used. After all preprocessing mentioned in Section 4.1, the final training sets are of the shape ($n_{samples}, n_{features}$) = (17449, 5544) and test sets are of the shape (8641, 5544).

4.3 Evaluation

To quantitatively evaluate the forecast, a Root Mean Squared Error metric is used, given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2}. \tag{6}$$

The summary of RMSE is given in Table 1.

Table 1: Train and test RMSE of the SVR models

Source of energy	test RMSE		train RMSE	
	1h	24h	1h	24h
Solar energy	938.90	2763.38	292.50	2202.48
Onshore wind energy	1363.61	1861.95	785.15	1280.17
Offshore wind energy	257.07	392.35	155.53	313.63

The qualitative results of predictions on a smaller range of test (held out) data are shown in Figures 1-3.

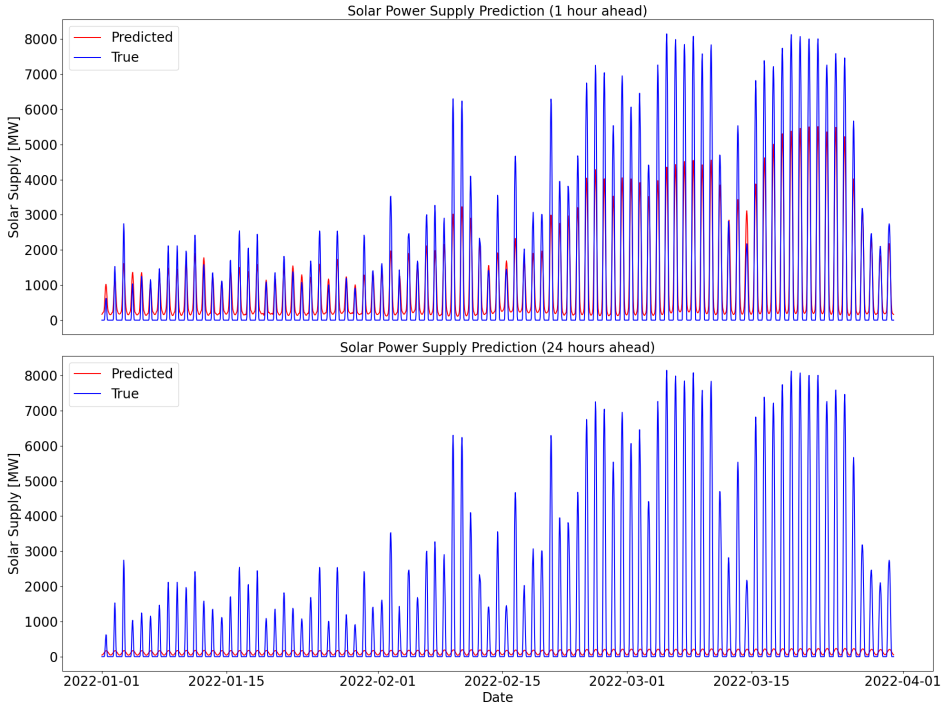

Figure 1: Solar energy predictions

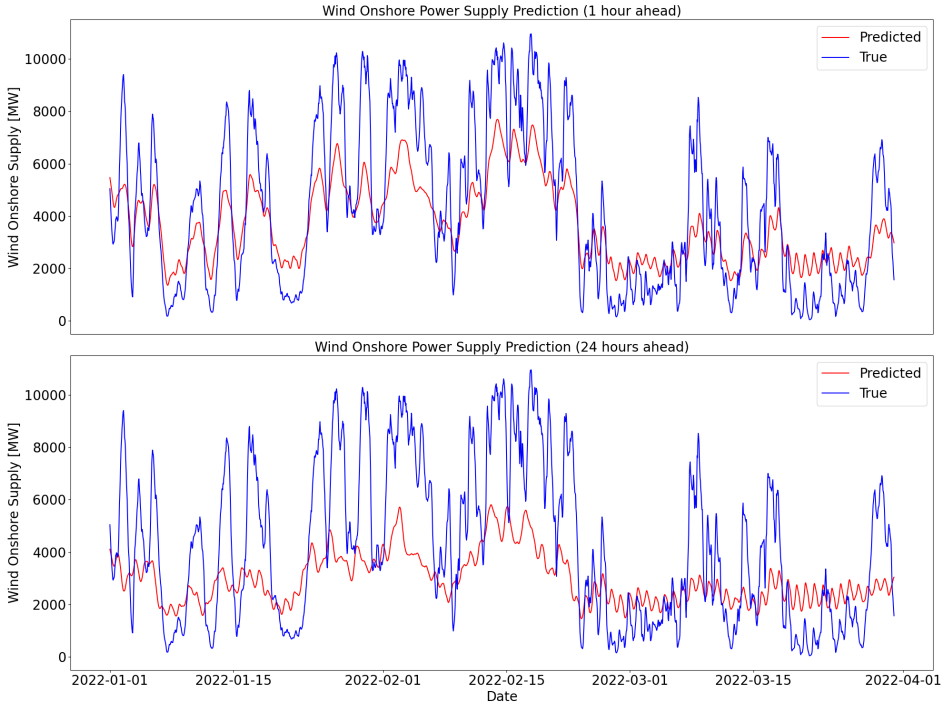

Figure 2: Wind onshore energy predictions

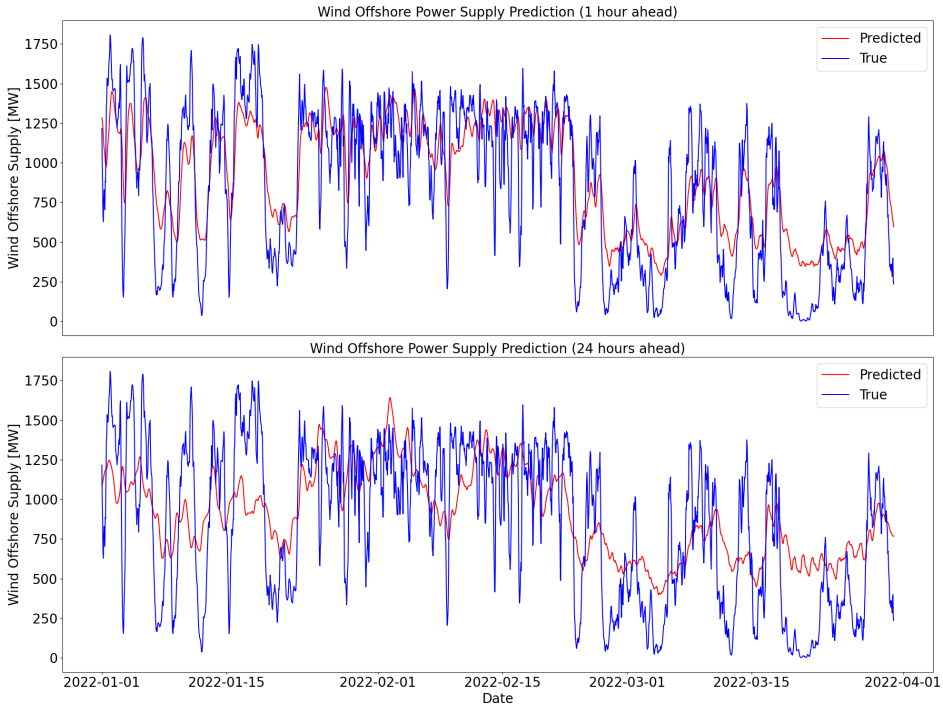

Figure 3: Wind offshore energy predictions

5 Discussion

Given the high RMSE on train data in Table 1, it can be concluded that all of the models underfit the training data. Given this observation, and poor generalization capabilities, RMSE for test data is significantly higher than train RMSE, which is not desired. In Figures 1-2, it can be seen that the fitted functions are smoother and that models can capture low-frequency dynamics of the target. Additionally, as expected, it is much more challenging for a model to predict a long-horizon target. When considering implementation efforts, data pre-processing was more laborious, so this goes in favor of an easy-to-use implementation of the method in the Scikit-learn library.
Given these results, and the discussion in the previous sections, several limitations could be addressed in the hope of improving the models. Most importantly, given more compute resources, models would greatly benefit from performing hyperparameter tuning. However, this comes at a high price that one needs to pay in time, given the same resolution, because SVR is not scalable. The first problem is a large number of samples and a quadratic to cubic time complexity of SVR in a number of samples. The second limitation is that no GPU support for parallelizable training is available. Next, some design choices could be revisited, such as choosing the optimal window size and scaling method. Also, better feature selection could be done, by applying some statistical tools. Moreover, one could try to address the existence of outliers and their effect on the performance of the model.

6 Conclusion

Accurate forecasting of fluctuating renewable energy sources is crucial for maintaining grid stability. Better predictions of renewable energy supply can significantly enhance the reliability of the power system. However, this can be challenging for statistical model-based methods, because of the sophisticated design choices that one has to make, as well as the bias-variance tradeoff that needs to be addressed. Here, it is shown that a vanilla version of the Support Vector Regression model, given the reported design choices, a large context length at the high resolution (three years of measurements every 1 hour), is not easily applicable to this problem, and suffers from oversmoothing the target function. Therefore, a recommendation would be to apply some methods that are known to be more

expressive, such as Artificial Neural Network based architectures, like Convolutional, Recurrent Neural Networks or Transformer-based models.

References

Abuella, M. and B. Chowdhury (2017). "Solar power forecasting using support vector regression". In: *arXiv preprint arXiv:1703.09851*.

Ben-Hur, A., D. Horn, H. T. Siegelmann, and V. Vapnik (2001). "Support vector clustering". In: *Journal of machine learning research* 2.Dec, pp. 125–137.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

Drucker, H., C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik (1996). "Support vector regression machines". In: *Advances in neural information processing systems* 9.

International Renewable Energy Agency (IRENA) (2020). *Innovation Landscape Brief: Advanced Forecasting of Variable Renewable Power Generation*. Abu Dhabi.

Kramer, O., N. A. Treiber, and F. Gieseke (2013). "Support Vector Machines for Wind Energy Prediction in Smart Grids." In: *EnviroInfo*, pp. 16–24.

Platt, J. (1998). "Sequential minimal optimization: A fast algorithm for training support vector machines". In.

Rüping, S. (2001). *SVM kernels for time series analysis*. Tech. rep. Technical report.

SVR API Reference (2012). `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html`, Scikit-learn SVR. [Online; accessed 18-Avg-2024].

Schölkopf, B. and A. Smola (2005). "Support vector machines and kernel algorithms". In: *Encyclopedia of Biostatistics*. Wiley, pp. 5328–5335.

Zhu, L., Q. H. Wu, M. S. Li, L. Jiang, and J. S. Smith (2013). "Support vector regression-based short-term wind power prediction with false neighbours filtered". In: *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 740–744. DOI: 10.1109/ICRERA.2013.6749851.

7 Appendix

7.1 Appendix A

The primal optimization problem of SVR:

$$\min_{\beta\in\mathbb{R}^d, b\in\mathbb{R}, \xi\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i,j}\beta_i\beta_j k(x_j, x_j) + \frac{C}{N}\sum_{i=1}^{N}\xi_i \tag{7}$$

$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{N}\beta_j k(x_j, x_i) + b\right) \geq 1 - \xi_i, \quad i = 1, ..., N.$$

Note that in Equation 7 vector $\xi \in \mathbb{R}^d$ is also optimized, which stands for the slack variable that allows for some data points to deviate from the ϵ tube. It represents the deviation of training samples outside the ϵ-insensitive zone.

7.2 Appendix B

Training and test wall clock times for only one run and a set of hyperparameters are given below, in Table 2. Many of the runs ended up in a crashing kernel, so they are not reported here. The details about the hardware used to train and test the models are given in Table 3.

7.3 Appendix C

Thanks to Okan Coskun from the University of Tübingen for creating Figure 4.

Table 2: Train and test wall clock times of the SVR models, **format**: hours:minutes:seconds

Source of energy	train		test	
	1h	24h	1h	24h
Solar energy	00 : 47 : 48	00 : 29 : 02	00 : 15 : 39	00 : 29 : 28
Onshore wind energy	00 : 28 : 49	00 : 23 : 10	00 : 11 : 07	00 : 10 : 34
Offshore wind energy	00 : 51 : 46	00 : 22 : 47	00 : 11 : 19	00 : 10 : 58

Parameter	**Value**
Architecture	x86_64
CPU(s)	8
Core(s) per socket	4
Thread(s) per core	2
CPU max MHz	3900.0000
CPU min MHz	400.0000
Model name	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
Vendor ID	GenuineIntel

Table 3: System Architecture Details

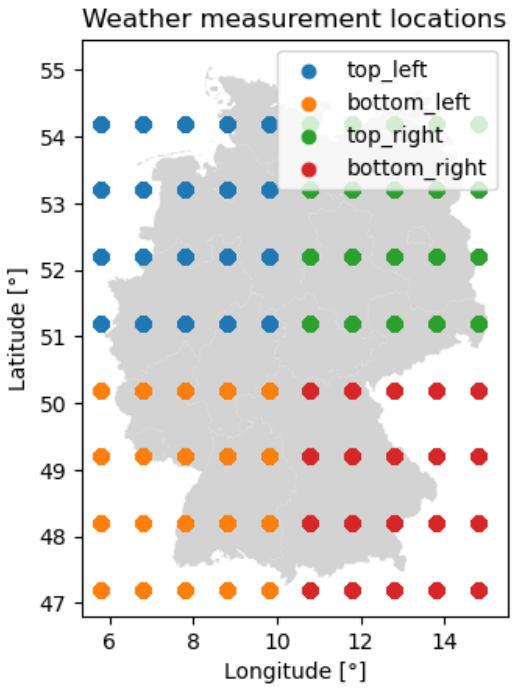

Figure 4: Weather measurements locations from a dataset. Colors represent the grouping of the locations, which are used as new features.