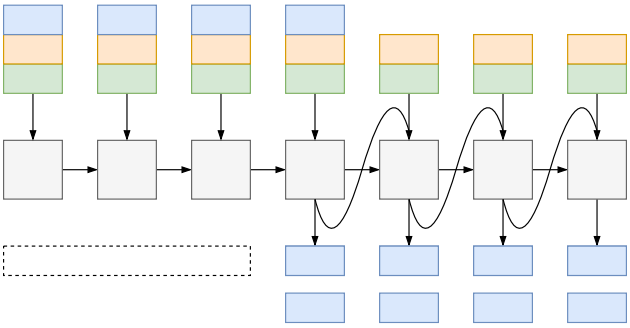
PREDICTING RENEWABLE ENERGY PRODUCTION USING MACHINE LEARNING METHODS

LUIS GENTNER, LEON SENGÜN, DILARA YILDIZ



MACHINE LEARNING METHODS IN MECHANICS
REPORT
-
University of Stuttgart

October 21, 2022

Contents

Acronyms

AE autoencoder

ARRNN autoregressive RNN

CNN convolutional neural network

DWD Deutscher Wetterdienst

GRU gated recurrent unit

LSTM long short-term memory

PCA principal component analysis

ReLU rectified linear unit

RNN recurrent neural network

1 Introduction

The increased share of renewable energies in the total energy mix leads to an increasingly fluctuating power generation. Therefore, measures for a stable energy supply gain importance. Information about future energy production could enable power plant operators to control their plants in advance to match the renewable energy production and customer demand, as well as to avoid paying negative electricity prices. A future prediction could also enable energy storage operators to foresee when energy storage is feasible and when to discharge their storage in a targeted manner. An overview of the German energy mix during one week can be seen in Figure 1.

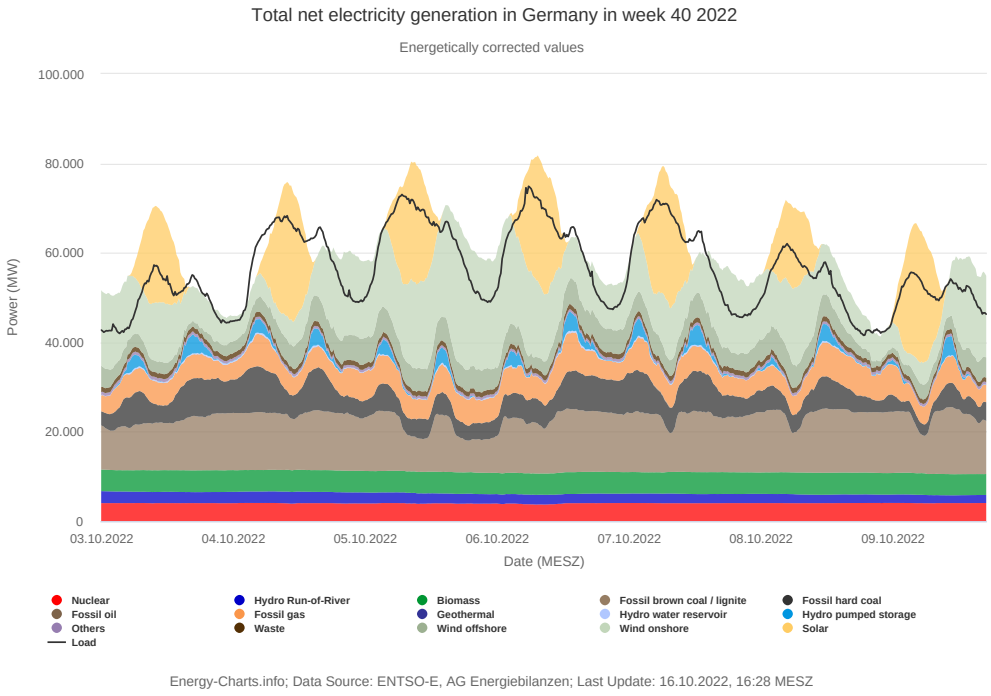


Figure 1: Weekly total net electricity generation in Germany, Week 40, 2022 [1]

With this background, the goal of this work is to develop a machine learning model that predicts the renewable energy production 12 hours into the future.

There are a variety of feasible model architectures to tackle this problem, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) networks. This paper will first explore and evaluate existing models regarding their accuracy, then iteratively develop an optimized, custom model architecture.

2 Data acquisition

2.1 Data sources

In order to predict future energy production from renewable sources, data of the energy production in recent years is needed. Old data sets are not representative because the number of solar and wind power plants in operation has increased significantly in recent years [2], [3]. Therefore, years 2017-2021 were selected to train all neural networks in this paper. The solar and wind energy generation highly depends on local weather conditions. To account for this, a second data set of

recorded weather parameters from the same time frame was chosen as an optional additional input to the models. The following data sets were used in this paper:

- Electricity data (2017-2021) obtained from Energy-Charts (from Fraunhofer ISE), hourly resolution [1]
- Weather data (2017-2021) from Deutscher Wetterdienst (DWD) including 117 meteorological stations in Germany, hourly resolution [4], [5]

2.2 Feature engineering

Raw data cannot necessarily be included directly as a feature in the model. Different pre-processing steps and a thoughtful selection of features can significantly improve and are essential for a meaningful prediction.

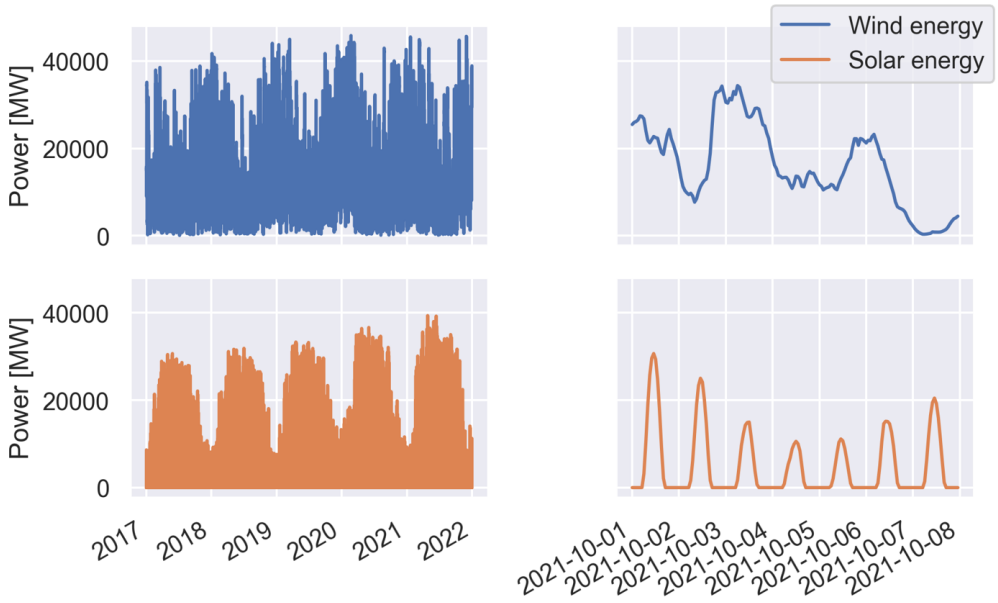
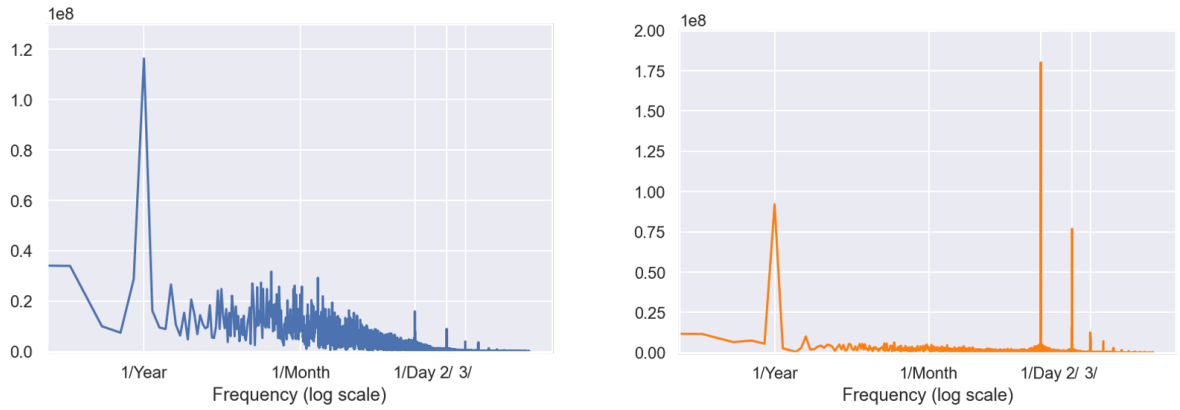


Figure 2: Wind and solar energy production 2017-2021

Figure 2 shows the wind and solar energy production of 2017-2021. Seasonal and daily fluctuations can be observed, as well as a year-over-year trend showing a general increase in power. The fluctuations are further investigated with a Fourier analysis, illustrated in Figure 3.

The Fourier analysis reveals which frequencies appear in the given data. The solar energy production shows the highest peak at a frequency of 1/day, matching the daily elevation of the sun. For both wind and solar energy, the yearly (1/year) and daily (1/day) fluctuations are dominant. To account for these two frequencies, periodic *time of day* and *time of year* signals were created. Each signal is represented by a sine and a cosine function (Figure 4). In addition, two sun elevation features were created. They represent the angle of the sun above or below the horizon, measured at the geometric center of Germany. While the *sun elevation* feature contains positive and negative angles, all negative values of the *clipped sun elevation* feature were set to zero, to account for the almost constant darkness and absence of solar energy production during the night.

The modified time features as well as air pressure, sunshine duration, temperature and wind speed were subjected to a correlation analysis. Figure 5 shows the correlation matrix. Darker colors



(a) Fourier analysis of wind energy production

(b) Fourier analysis of solar energy production

Figure 3: Fourier analysis of electricity data

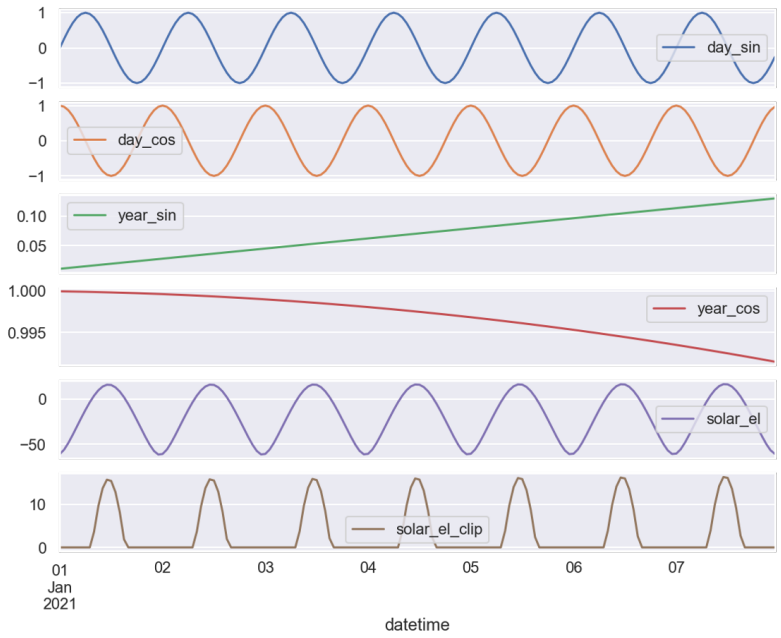


Figure 4: Periodic features

indicate a higher (anti-)correlation. The solar energy production is strongly correlated to the cosine of the day, sunshine duration, temperature and clipped solar elevation. As expected, both features correlate strongly with solar energy production, with the *clipped sun elevation* showing the highest correlation (see Figure 5). In contrast, wind energy production is only strongly correlated to wind speed and the cosine time of year signal. It is slightly correlated to pressure and temperature, because wind energy output is dependent on air density, which is a function of temperature and pressure. Although the sine of day and year do not hold a high correlation, they were included in the feature vector for completeness, since their omission does not significantly reduce the input dimension. The final feature vector is a combination of 2 energy production features, 6 analytically determined time and solar elevation features, and varying amount of weather features, dependant on the number of selected weather stations. All input features are listed in Table 1.

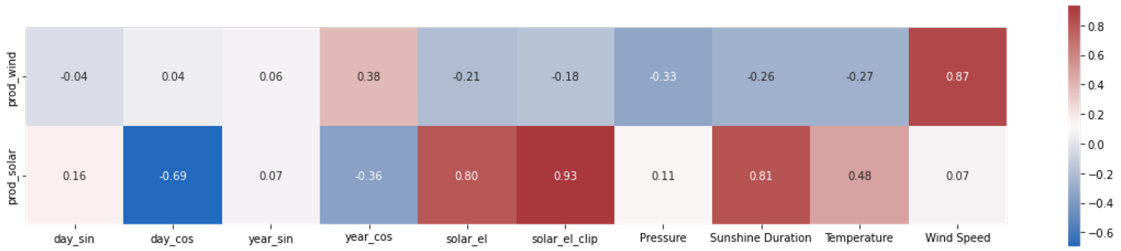


Figure 5: Correlation analysis

Table 1: Features

Energy production	Time and solar elevation	Weather
Solar energy prod.	Time of day sine & cosine	Temperature*
Wind energy prod.	Time of year sine & cosine	Wind speed*
	Solar elevation	Pressure*
	Clipped solar elevation	Sunshine duration*

* Multiple features per weather parameter: One feature per selected DWD station

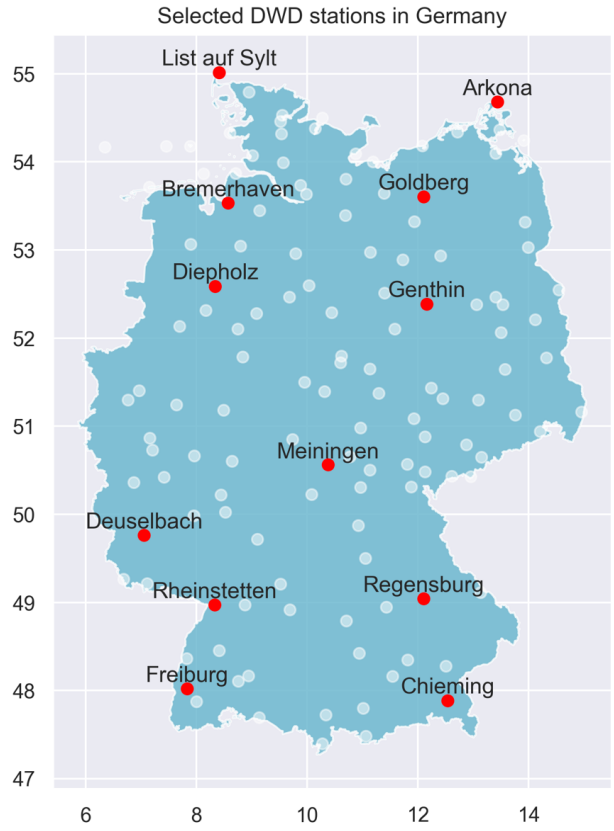


Figure 6: Selected weather stations in Germany. Red points indicate the selected stations, grey points indicate all usable stations that match the required data reliability.

Out of over 800 meteorological stations in Germany, only 117 provide reliable data for all the required parameters in the time frame of 2017-2021. Of these remaining stations, 12 were selected as an approximate representation of the German weather. Decisive for the selection was the spatial proximity to offshore and solar installations as well as a broad distribution across Germany. The final weather stations are illustrated in Figure 6.

3 Methods and model architectures

The goal of this work is to predict solar and wind energy production 12 hours into the future. Current research offers a variety of machine learning models for this problem. Therefore, as a first step, different models were applied to the problem and evaluated regarding their prediction accuracy. Unpromising approaches were then discarded while successful models were further explored and improved. The R^2 score was used as a measure of the accuracy of the prediction. The following models were investigated, and were based on the models found in [6]:

Repeat last step The non-trainable model repeats the last given time step, leading to a constant solar and wind energy prediction.

Repeat yesterday The non-trainable model outputs the time steps with the same local time from the day before.

Linear The model consists of one `Dense` layer with linear activation.

Dense A deep neural network, consisting of three rectified linear unit (ReLU) activated `Dense` layers.

CNN 3x256 Convolutional neural network consisting of one ReLU activated `Conv1D` layer with 256 filters. The model *CNN 3x256* was fed with the last three time steps.

CNN 24x256 Like the previous model, but this model was fed with the last 24 time steps.

SRNN 32 A RNN consisting of one `SimpleRNN` layer with 32 units. The model is fed with the last 24 time steps and predicts the next 12 time steps in a single shot with an attached `Dense` layer

GRU 32 Like the previous model, but consisting of a `GRU` layer. Gated recurrent units (GRUs) add an update gate in comparison to SimpleRNNs.

LSTM 32 Like the previous model, but consisting of a `LSTM` layer. Long short-term memorys (LSTMs) add an forget and output gate in comparison to GRUs.

LSTM 32^2 Deep LSTM with two layers each containing 32 units.

Since RNNs are specialized in time series, a variety of different model architectures were investigates and further developed.

3.1 Single-shot RNNs

Single-shot RNNs are the simplest recurrent neural networks. After the warmup phase, they predict a predefined number of future steps at once. Although they depict the temporal dependencies in time series, their structure does not allow to vary the length of the prediction output without retraining.

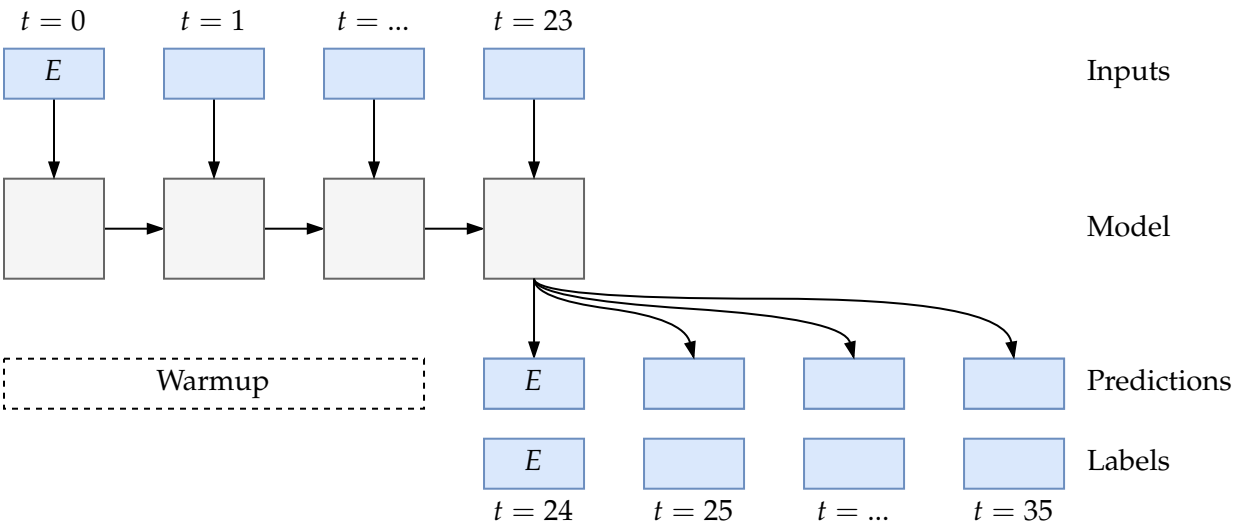


Figure 7: Single-shot RNN with energy production (E) input

3.2 Autoregressive RNNs

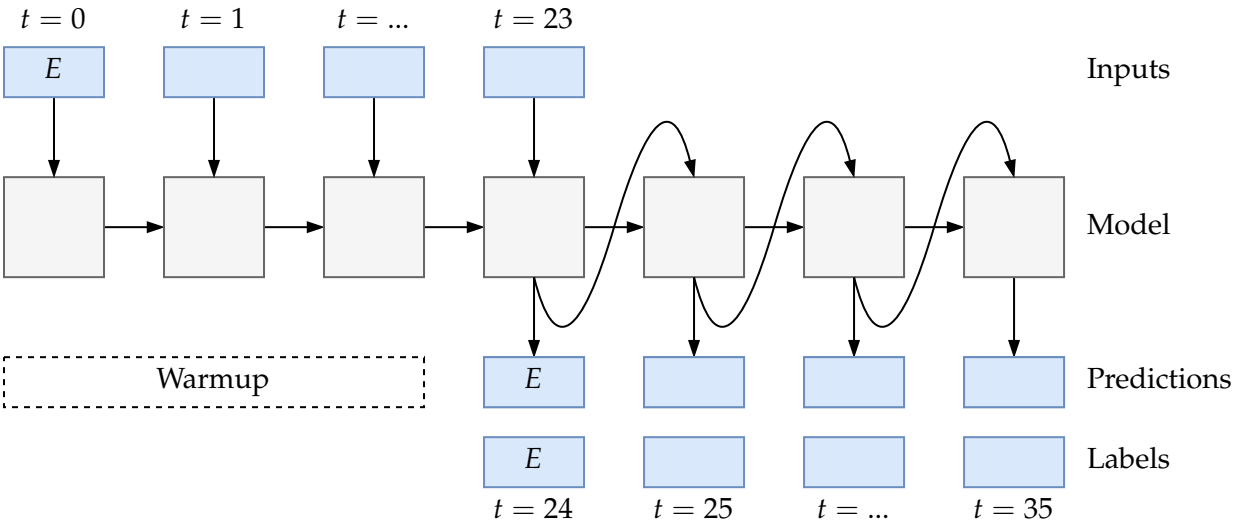


Figure 8: ARRNN with energy production (E) input

Autoregressive neural networks work differently: The RNN cells predict only a single time step into the future. After the warmup phase, the prediction the cell is used as input for the next prediction. This architecture enables a customisable output length without retraining.

3.3 ARRNNs with continuous input

One disadvantage of the previously described models is that they can not take into account forecasted data that might be available for future time steps. Even though time, solar elevation, and

weather forecasts are known for the future, they cannot be included in the previous models due to their given architectures. Therefore, a new model architecture, called ARRNN with continuous input, was developed. It makes use of a custom-built Tensorflow model class with modified `call` and `warmup` methods. During `warmup`, which acts as a *past phase*, the model works like a traditional RNN, where consecutive time steps of power, time, and weather data are fed to an RNN layer. When all available time steps are fed into the layer, it makes a single prediction one time step into the future. During *prediction phase*, the model is only fed with time and weather inputs. In this stage the model works autoregressively - the last power prediction is concatenated with the time and weather inputs and fed into the RNN cell for the next prediction.

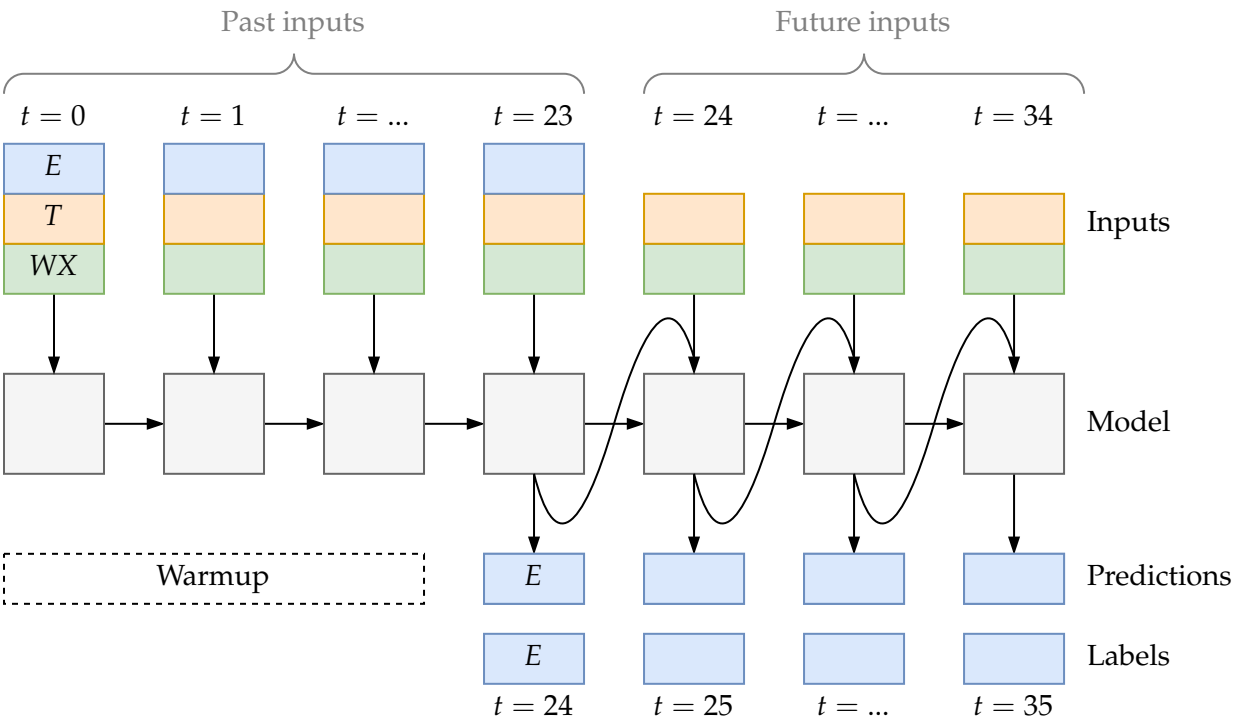


Figure 9: ARRNN with energy production (E) input and continuous input of time and solar elevation (T) and weather parameters (WX)

The continuous ARRNN was realized with both GRUs and LSTMs. Different numbers of layers and units were investigated.

3.4 Input compression

For better predictions, the amount of weather stations could be increased. One possibility would be to directly feed the weather parameters of additional weather stations into the RNN. To prevent an overloaded feature vector, the idea was to increase the feature density instead. Therefore, two approaches were tested: Feature compression by using an autoencoder (AE) and using a principal component analysis (PCA).

Autoencoder An autoencoder was created to verify the quality of the compression. The AE inputs and outputs 468 weather parameters of 117 individual DWD stations and has a latent space vector of size 48. The latent space representation was then fed into the RNN, which means

that the number of input features stayed constant to the previous model, that is directly fed with 48 weather parameters of 12 stations. The original, and reconstructed data, as well as the latent space of a single time step are shown in Figure 11.

PCA A linear PCA was used to compress the number of features even further, from 468 to 8. The result for a single time step is shown in Figure 12.

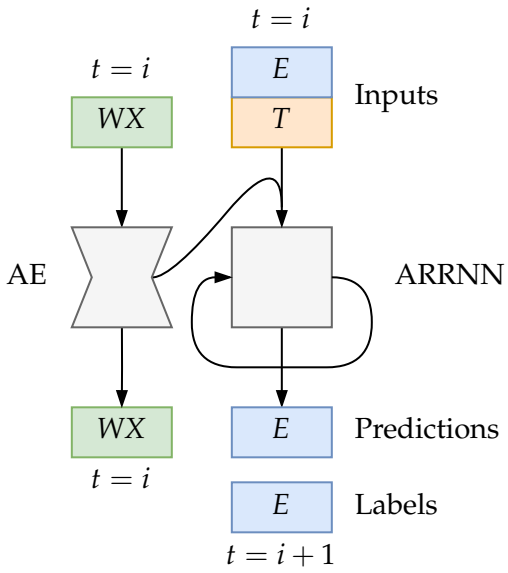


Figure 10: Schematic of the combined ARRNN and AE model. The energy production (E) and time and solar elevation (T) input is fed directly into the ARRNN, while the weather parameters are first compressed with an AE and then fed into the ARRNN in their latent space representation.

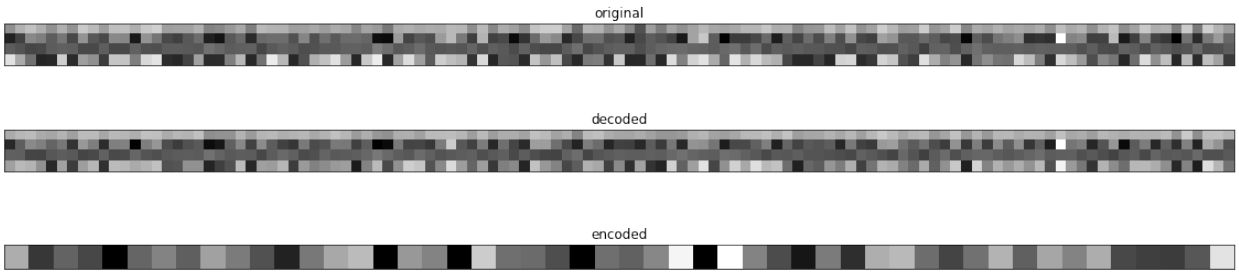


Figure 11: Original, decoded and encoded weather data using an AE

3.5 Final model architecture

Figure 13 shows a summary of the iterative model design process, going from less complex to more advanced models.

Although the PCA can only model linear relationships and compresses the data to only eight features, its performance is almost on par with that of the more complex encoder. Therefore, it was used in the final model:

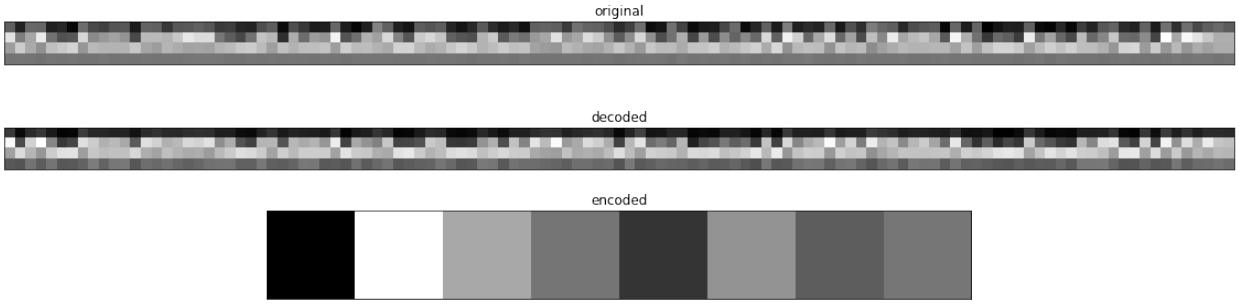


Figure 12: Original, decoded and encoded weather data using PCA

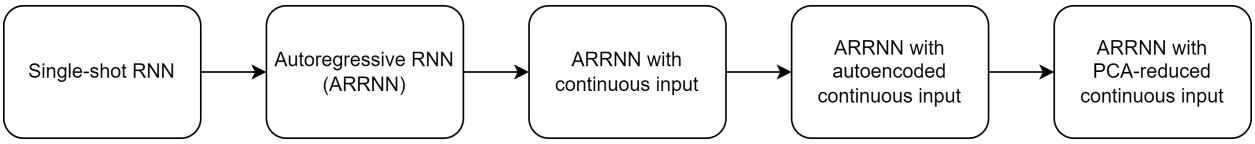


Figure 13: Iterative model development

- Deep autoregressive GRU with continuous input
- 3 layers [32, 32, 32]
- Inputs (24 time steps):
  - Solar and wind energy production (during past/warmup phase)
  - 8 time and sun elevation features
  - 8 PCA reduced weather features
- Prediction of 12 time steps

3.6 Optimization

To further improve the prediction accuracy of the final model, hyperparameter optimization was performed. Due to an incompatible Tensorflow version on the institute server, the optimization was run on desktop-grade hardware. This limited the number of possible optimization trials to 25. Optuna, an open source hyperparameter optimization framework was used to optimize the following three hyperparameters:

- Learning rate
- Units per layer
- Past input steps

Figure 14 displays the hyperparameter importances for the trial duration. While learning rate and the number of past steps show a large impact on the duration, the number of units per layer plays only a minor role.

Examining the optimization curve in Figure 15, it can be seen that after the 25 trials, only a minor optimization was achieved, reducing the loss by about 0.005. In order to further improve the results, optimization should be performed with significantly more trials.

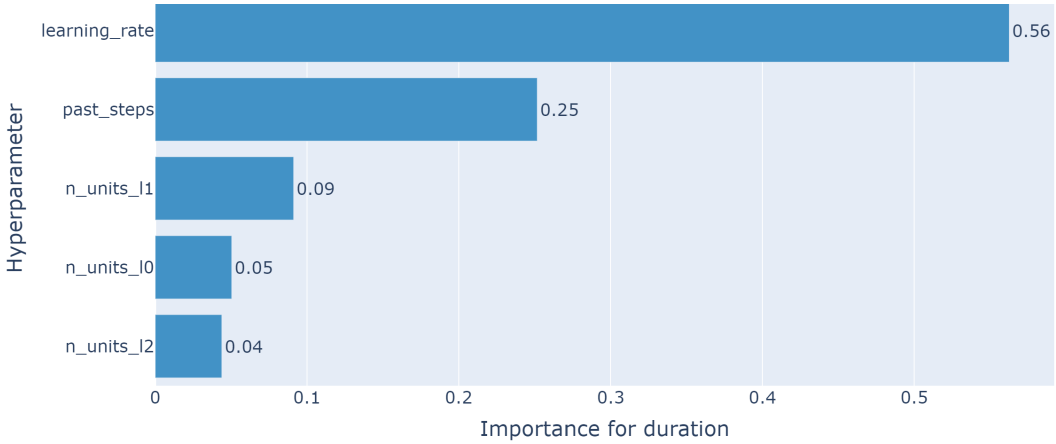


Figure 14: Hyperparamter importances

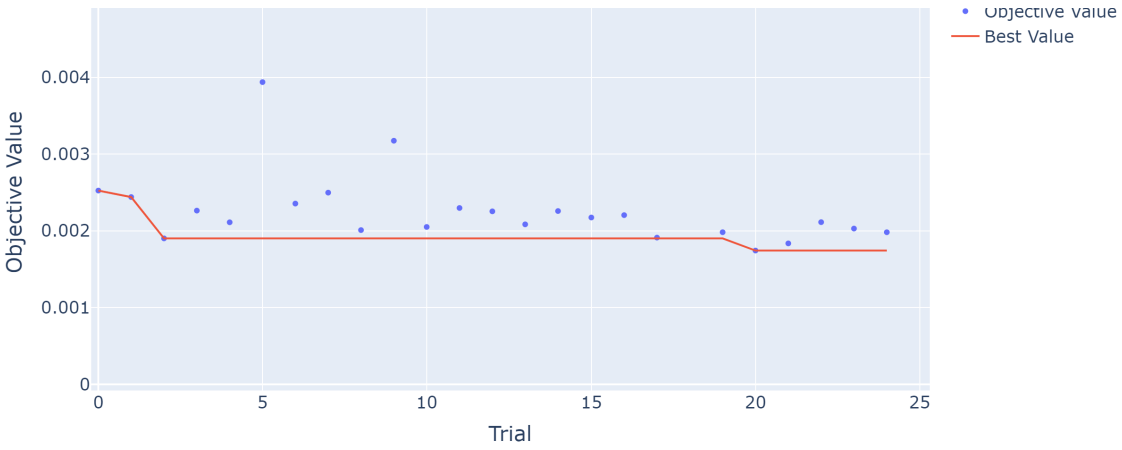


Figure 15: Hyperparameter optimization history

Table 2 displays the optimized hyperparameters in comparison to the previous model. Striking is a fourfold increase in the number of past steps as well as a higher learning rate. The units per layer were all reduced to 10-15. However, due to the insufficient number of trials, the reliability of the hyperparamter optimization results is questionable. This is also reflected in the R^2 score, which differs by only 0.5 %.

3.7 Cross-validation

To ensure that the test and validation data sets were representative for the whole data set and did not skew the model performance, a cross-validation was performed using the K-fold method. For this purpose, the entire data set was split in 5 different subsets, whereby in each fold a different subset served as validation set, while the remaining subsets formed the training set. The model was then trained for each fold separately. The results are listed in Table 3. It can be seen that apart from fold 1, the position of the validation set does not have a significant influence on the model performance.

Table 2: Comparison between the non-optimized and optimized final model

Parameter	Non-optimized model	Optimized model
Past steps	24	92
Learning rate	5×10^{-4}	1.7×10^{-3}
Units 1st layer	32	15
Units 2nd layer	32	10
Units 3rd layer	32	11
Test R^2 score	**0.9616**	**0.9567**

Table 3: K-Fold cross-validation

Fold	Val R^2	Test R^2
1	0.9617	0.9315
2	0.9690	0.9574
3	0.9686	0.9607
4	0.9659	0.9558
5	0.9516	0.9444
Average	**0.9633**	**0.9500**

4 Results

The main idea of this work is to predict the solar and wind energy production 12 hours into the future. Therefore different models and architectures were introduced (section 3). In this chapter, the results of these models are presented allowing them to be compared and evaluated.

Figure 16 shows the R^2 scores, that were reached by the different models. Since the models *Repeat last step* and *Repeat yesterday* only repeat the the energy production input, their performance is very low ($R^2 < 0.5$). While the *Linear* model still shows low accuracy, deep neural network approaches reach scores over 0.8. Recurrent neural networks perform better, because their structure is specialized for time series. The advanced types of recurrent units, GRUs and LSTMs, provide the most accurate results and were therefore used for further model development.

As the performance of single-shot RNNs is limited, more complex network architectures, ARRNNs, were investigated.

Figure 17 shows the R^2 scores of the various autoregressive networks. The number behind the model name indicates the number of layers as well as the number of units per layer. "+ wx" marks the models where weather data was included in form of a continuous ARRNN. Since GRUs and LSTMs were on par regarding their performance, the less complex GRUs were selected as the basis for advanced model versions with encoder or PCA-compressed input.

When the ARLSTM model with 32 units is additionally fed with weather data, a 4 % increase in the R^2 score is achieved. This can be easily explained by the additional, strongly correlating information that is provided, leading to an improved accuracy. The GRU shows similar scores to the LSTM. However, an increase of the number of units up to 96 resulted in a 2 % decrease of accuracy. One assumption is that the increased number of trainable parameters cannot be

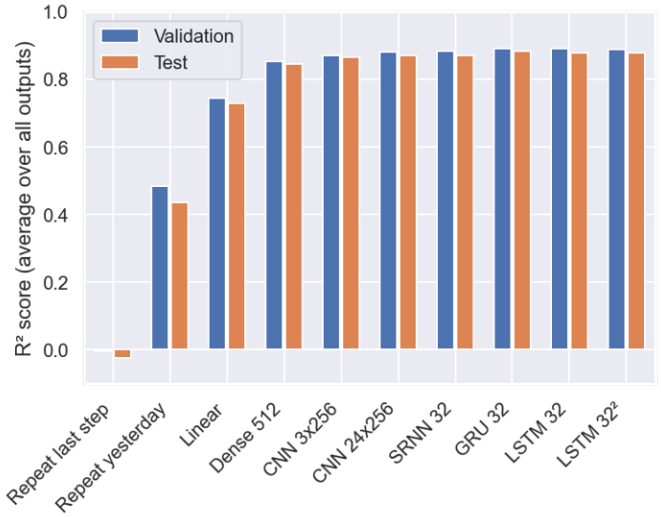


Figure 16: R^2 scores of benchmark and single-shot prediction networks

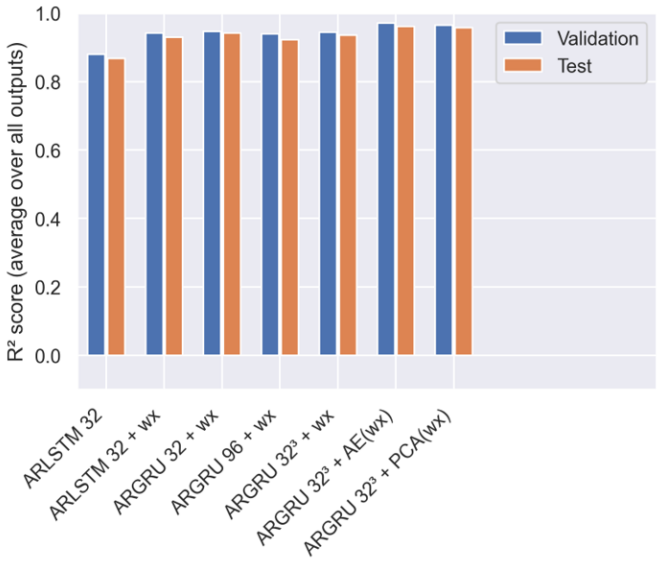


Figure 17: R^2 scores of different ARRNN architectures

adequately determined due to the limited amount of training data. In contrast, deep ARRNNs, consisting of three RNN layers instead of one, did lead to a slight score improvement.

A further improvement in accuracy was achieved with the encoder and PCA reaching R^2 scores up to approximately 0.95. Compressing the weather data including all reliable weather stations in Germany in the encoder or PCA, allows the neural network to incorporate more weather information without overloading the feature vector. This compressed weather data input lead to more accurate predictions. However, it is unclear why the ARGRU with upstream PCA leads to similarly accurate results than the more complex encoder, despite its linearity and more aggressive compression. One possible explanation is that the important information actually follows linearity and consists of only 8 condensed relevant features. The ARRNN with PCA-compressed weather data was chosen as the final model, because the reduced input size results in faster training and

optimization times.

Figure 18 displays the prediction of solar and wind energy production in randomly selected time frames using the final model. The solar energy predictions are close to the labels with only minor deviations, whereas the wind energy prediction has larger discrepancies from ground truth. The reason can be seen in the correlation analysis (Figure 5) - while solar energy production is highly correlated with more than three of the given features, wind energy only shows a strong correlation to wind speed. Accordingly, the prediction of wind energy based on these features is also lower.

5 Conclusion

The goal of this paper was to predict solar and wind energy production 12 hours into the future. Therefore a variety of existing machine learning approaches were investigated and new model architectures were developed. The project has shown that the use of RNNs is beneficial for these types of tasks, as they are best suited for modeling time series data and also enable autoregressive architectures. The addition of weather data in the input improved the accuracy of the predictions significantly, especially the prediction of wind power. The particular idea of this work was to also allow future inputs to the autoregressive network to fully exploit available time, sun elevation and weather forecasts for future time steps. For this purpose an autoregressive RNN with continuous input was developed, which significantly improved the accuracy of the predictions. Furthermore, reducing the dimensionality of the weather data shortened the training time significantly and enabled optimization of the hyperparameters without compromising too much accuracy. This dimensionality reduction was performed using two methods, an autoencoder and a linear principal component analysis. Although PCA only maps linear relationships, it achieved similar scores with less complexity and computation time. A hyperparameter optimization was performed, but due to a lack of processing power, only 25 trials could be completed. Therefore, improvement achieved by the optimization was fairly limited.

6 Outlook

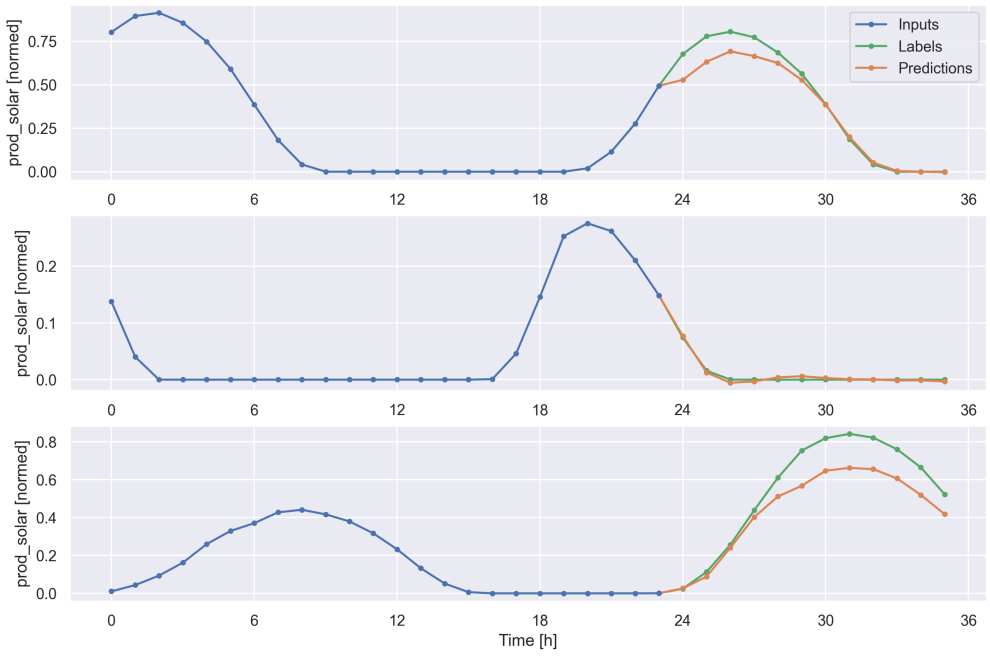
Topics that were out of the scope of this project, but could be implemented in a further development include the account for weather forecast inaccuracy by distorting the weather with noise, e.g. a random walk, to account for the increasing uncertainty of the weather forecast over time. Doing a correlation analysis of actual forecast weather data and recorded values for of same time step could reveal the extent of inaccuracy. Testing with real forecast data could reveal the real-world usability of the model in a commercial scenario.

Moreover, by using a CNN autoencoder or a graph neural network could help conserve spatial relationships between the weather stations, which are lost by flattening the data set into a vector.

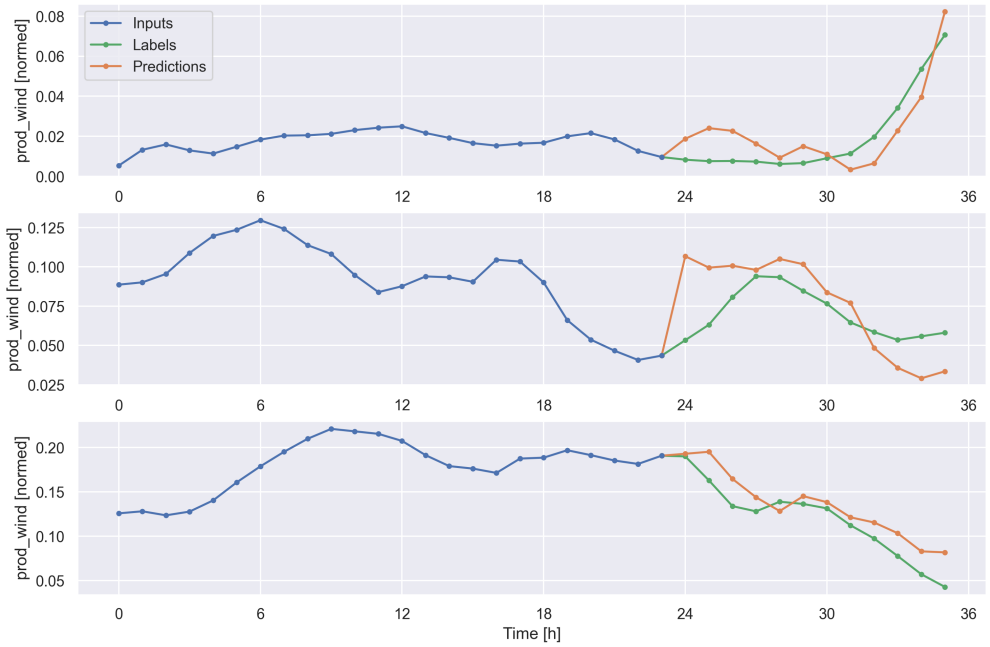
Statistical methods, like Bayesian structural time series [7], [8] could be employed as a comparison to the neural network approach.

References

[1] Energy-Charts. “Public net electricity generation in germany,” Fraunhofer ISE. (2022), [Online]. Available: https://www.energy-charts.info/charts/power/chart.htm?l=en&c=DE (visited on 10/20/2022).



(a) Solar energy production



(b) Wind energy production

Figure 18: Prediction of solar and wind energy production of the final optimized model

[2] M. Jaganmohan. "Cumulative solar photovoltaic capacity in germany from 2017 to 2021," Statista. (May 2022), [Online]. Available: https://www.statista.com/statistics/497448/connected-and-cumulated-photovoltaic-capacity-in-germany/ (visited on 10/20/2022).

[3] M. Jaganmohan. "Installed wind power capacity in germany from 2008 to 2021," Statista. (Apr. 2022), [Online]. Available: https://www.statista.com/statistics/421797/tracking-wind-power-in-germany/ (visited on 10/20/2022).

[4] Earthobservations. "Wetterdienst documentation." (2022), [Online]. Available: https://wetterdienst.readthedocs.io/en/latest/overview.html (visited on 10/20/2022).

[5] Deutscher Wetterdienst. "Wetter und Klima - Deutscher Wetterdienst - Leistungen - CDC-Portal." (2022), [Online]. Available: https://www.dwd.de/DE/leistungen/cdc_portal/cdc_portal.html (visited on 10/20/2022).

[6] Tensorflow contributors. "Time series forecasting," TensorFlow. (May 9, 2022), [Online]. Available: https://www.tensorflow.org/tutorials/structured_data/time_series (visited on 10/21/2022).

[7] S. L. Scott and H. R. Varian, "Predicting the present with bayesian structural time series," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 5, no. 1, p. 4, 2014, ISSN: 2040-3607, 2040-3615. DOI: 10.1504/IJMMNO.2014.059942. [Online]. Available: http://www.inderscience.com/link.php?id=59942 (visited on 10/21/2022).

[8] Tensorflow contributors. "Structural time series modeling case studies: Atmospheric CO2 and electricity demand," TensorFlow. (Jan. 26, 2022), [Online]. Available: https://www.tensorflow.org/probability/examples/Structural_Time_Series_Modeling_Case_Studies_Atmospheric_CO2_and_Electricity_Demand (visited on 10/21/2022).