PV Generation Forecasting based on weather data

Olga Klyagina Ekaterina Dorzhieva Mile Mitrovic Mikhail Kirillov Galina Chikunova Anastasia Sozykina Aleksei Shcherbak TA: Egor Zakharov

Abstract

This document provides the final project report at **Deep Learning 2021 course** organized by *Skolkovo University of Science and Technology* (Skoltech). Paper describes several methods of reconstructing missing weather data, weather forecasting and solar panels power prediction.

Github repo: https://github.com/BORDYR/PV-Generation-Forecasting
Video presentation: youtube.com

1. Introduction

A growth of renewable energy sources share is observed during the past decades. Photo-voltaic panels, or PV panels, are now available for installation for energetic companies, businesses, and individuals. However, the output of such sources is unstable, which might be harmful to the connected grid. By the Energetic Power System side, it is essential to have a well-predicted PV Generation for making short or long time planning. Thus, we introduce the use of weather data for PV Generation forecasting. At the same time, the weather data inputs can be missing due to the failures during recording or transmitting the signal, which makes the application of prediction models a complex task.

Deep Learning approaches have performed as powerful tools to predict data in many domains. This particular project aims to implement deep-learning techniques both for imputing the missing weather data and predicting the PV Generation using the weather information.

2. Preliminaries

2.1. Dataset

The dataset was collected during 2013-2014 in the UK and available by the link (UKPowerNetworks). It includes the measured data from 20 substations, 10 domestic PV installations and 9 weather stations installed on the substations. The PV power generation data was collected from the domestic installations with one-minute, ten-minutes and hourly resolution. Since the weather data is available with 30-minutes intervals, we choose ten-minutes power measurements and united them with weather data by matching time and location.
The weather stations measured 33 meteorological parameters. However, only a few weather conditions affect the power generation directly: outside temperature, humidity and wind, which cool down the panels, and solar irradiance. Thus, we ended up with the set of 8 features which represent the listed ones.
Power generation value depends on the solar panel's characteristics, which are different for every household. Thus, for certainty, data from one PV installation was used.
In total, the dataset consists of 7805 measurements collected from June to November, 2014. The first 80% were used for training, 10% for validation and 10% for testing.

3. Related work

3.1. Missing Data Imputation

The majority of papers introducing data-driven methods assume that the inputs for the model are fully-connected and complete. However, only a few works look into the data availability problem profoundly. In practice, for the real-time measurements these approaches could be not effective due to the absence of data points.

The first idea and inspiration for our project was to implement the approach described in Liu et al. (2021). This paper introduces the Super-Resolution Perception (SRPCNN) model, applied to the PV data in order to fill the missing values. With the data recovered by SRPCNN, the authors use randomized learning algorithm SCN (Wang & Li, 2017) to train a regression model for PV generation forecasting.

The authors claim that the proposed method improves the forecasting accuracy even for the low fraction of the missing data.

The second approach for the missing data imputation is an adaptation of the GAN framework. In Yoon et al. (2018) the authors propose Generative Adversarial Imputation Nets (GAIN) and apply it to different real-world datasets.

3.2. Time-Series Forecasting

Time Series Forecasting is a hot topic with many possible applications. There are many approaches for time series prediction as statistical models (for instance, AR, ARIMA, Prophet(Facebook), VAR and others) that could as classical time series forecasting tools, so and Machine Learning models.

In this project we consider Neural Network approach for time series forecasting. Even though forecasting can be considered as a subset of supervised regression problems, some specific tools are necessary due to the temporal nature of observation.

We apply Long-Short-Term-Memory (LSTM) and Transformer architecture. LSTMs process data points sequentially. This architecture maintains a hidden state that is updated with every new input token, representing the entire sequence it has seen. Theoretically, very important information can propagate over infinitely long sequences. However, in practice, this it not the case. Due to the vanishing gradient problem, the LSTM will eventually forget earlier data points. In comparison, Transformers retain direct connections to all previous timestamps, allowing information to propagate over much longer sequences. However, this entails a new challenge: the model will be directly connected to an exploding amount of input.

4. Algorithms and Models

4.1. Frameworks for the Weather Missing Data Imputation

4.1.1. Super-Resolution Perception Convolutional Neural Network (SRPCNN)

The first chosen approach is to recover the missing data using the SRPCNN model. It consists of the Feature extraction, Information supplement and Reconstruction layers (see Fig.1).
The input data $X \in R^{p \times d_f}$ is incomplete: it has p instances and d_f features, while the full number of features is $d_c > d_f$.
The feature extractor contains one convolutional layer, which receives the input $X \in R^{p \times d_f}$, extracts the features and handles them as m feature vectors of length d_f.
Information supplement has n residual blocks with local connections and one global residual connection. This part adds the missing information to the feature vectors. Each residual block consists of two convolutional layers and the Rectified Linear Unit (ReLU) as an activation function.
Finally, in the reconstruction part, the feature vectors are in-

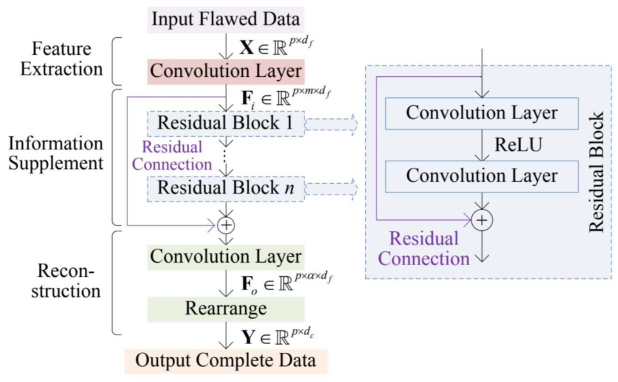


Figure 1. The structure of SRPCNN. Adapted from Liu et al. (2021).

tegrated into α sub-vectors by the convolutional layer. After that the sub-vectors are rearranged to perform the complete data $Y \in R^{p \times d_c}$.

4.1.2. Generative Adversarial Imputation Net (GAIN)

Second method consists of a generator and discriminator. The generator (G) imputes the missing components based on real data. The discriminator (D) then determines which components of completed data were actually observed and which were imputed. The input of D also takes a hint vector that provides information about the missingness of the original sample. This hint ensures that G does in fact learn to generate according to the true data distribution.
In this case, we have changed the approach to data skipping. In this approach, data may be missing in all features in random order.
The G takes missing data - $\hat{X}$, mask - M, and some noise Z that it puts instead of missing values (1). The mask matrix has ones where data is not lost and zeros otherwise. Noise are numbers from a sample from 0 to 0.01 (the dataset is pre-normalized). The G output is connected to the original dataset to keep the non-missing values unchanged (2).

$$\bar{X} = G(X, M, (1 - M) * Z) \tag{1}$$

$$\hat{X} = M * X + (1 - M) * \hat{X} \tag{2}$$

The generator and discriminator consist of 3 linear layers with relu activation in the first two and sigmoid activation in the latter case.

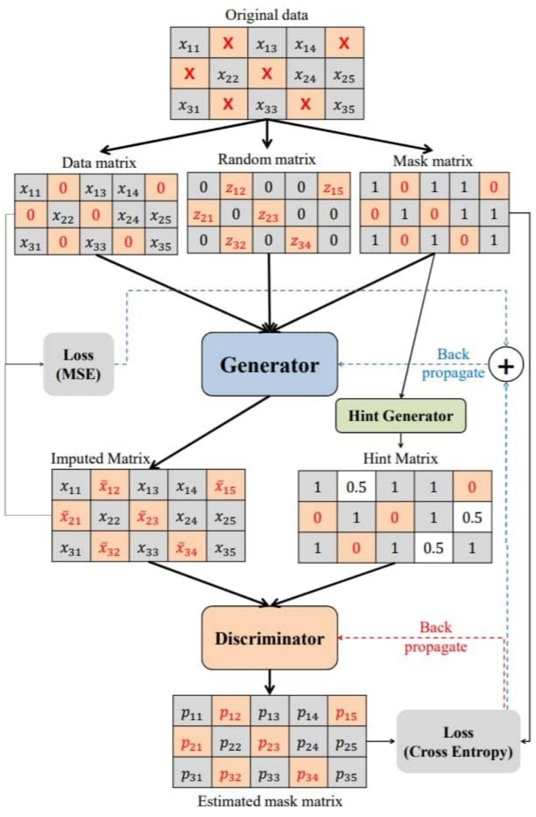


Figure 2. The structure of GAIN. Adapted from Yoon et al. (2018).

4.2. Frameworks for the PV Generation Prediction

4.2.1. Long short-term memory (LSTM)

LSTM architecture was chosen as the baseline architecture for prediction as the one which is widely used in time-series forecasting. It is a recurrent neural network that was introduced in 8 and has the ability to remember the previous data and learn the long-time dependences. One LSTM cell consists of four layers: forget gate sigmoid layer, which decides either to forget the information or not; input gate sigmoid layer, which maps the values to be updated; cell updating layer and the output layer. The hidden state and old cell state are passed to the input ow the next cell as well as the new time step input. The model architecture that was used in this study is following:

- Number of recurrent layers - 2
- Number of layers - 2
- Hidden size - 100
- Optimizer - Adam
- Learning rate - $1e-3$

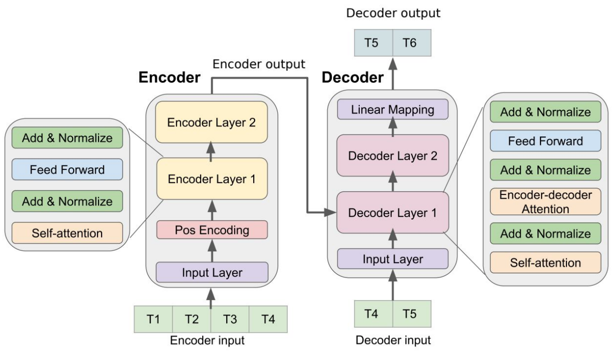


Figure 3. Architecture of Transformer-based forecasting model (Wu et al., 2020).

- Number of epochs - 80
- Criterion - MSELoss

4.2.2. Transformer

Transformer model (Vaswani et al., 2017) as the state-of-the-art model in NLP tasks may be applied for time series prediction. Transformer related experiments are based on (Wu et al., 2020) article, the architecture is shown on Fig. 3). In Transformer related part we combine 2 previous tasks - to predict features and predict target in one model. So the output of the transformer is one step ahead of the prediction set of all features and target.

Moreover, we consider and apply approach described in (Lee-Thorp et al., 2021), the main idea of this research is replace Attention Blocks in Encoder to Fast Fourier Transform (Fig. 4). The method provides less training and inference time with little or even no loss of quality. Besides, our contribution is that we apply this approach for time series forecasting though the authors used the method for NLP problems.

4.2.3. Direct PV power prediction

In terms of PV output prediction based on the weather data a radial neural network (RNN), multilayer perceptron (MLP) and artificial neural network (ANN) models were used in the relevant paper (Lo Brano et al., 2014). Gamma memory processing element (PE) additionally was used there as an element of dynamic systems to remember past signals. From the technical point of view, generation ability of PV batteries depends on the current values of their operation conditions only and mostly depends on the temperature and solar irradiance (Lo Brano et al., 2014). Thus, a simple model for the correct prediction of a PV output should be enough. Because of this technical reason and for the simplicity (we focused more on the forecasting and restoring

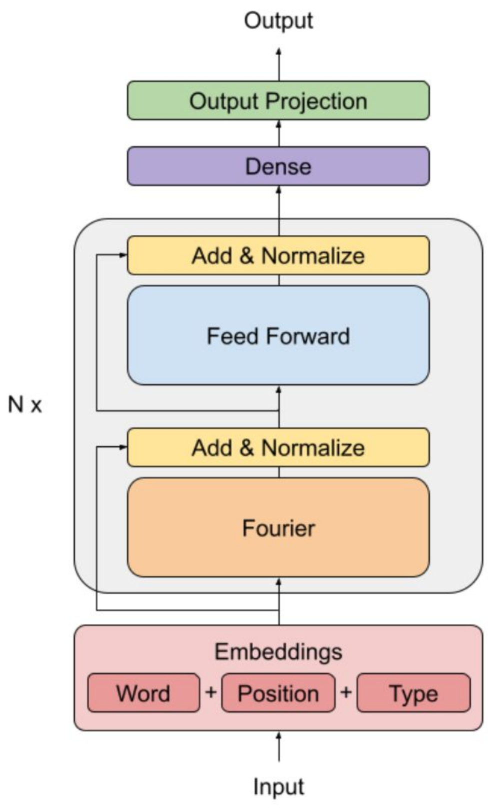


Figure 4. Encoder architecture with N encoder Fourier blocks(Lee-Thorp et al., 2021).

weather data problems) a linear model consisting in 6 layers was chosen for the PV prediction task in this project.

5. Experiments and Results

5.1. Frameworks for the Weather Missing Data Imputation

5.1.1. Super-Resolution Perception Convolutional Neural Network (SRPCNN)

We re-implemented the SRPCNN model described in Liu et al. (2021). To follow the structure proposed by authors, we created the input in the following way: as model takes d_f features (in our case - weather signals), at every batch creation we randomly skipped one feature. We could not change the number of missing features to have always a proper number of channels in models layers.
After that we conducted several experiments to define the best model configuration. The best accuracy was obtained with the following parameters: Convolution layer - $kernelsize = 3, padding = 1, m = 100$; Reconstruction part - $kernelsize = 3, padding = 1, alpha = 20$; Information Supplement part - $kernelsize = 5, padding = 2, layers = 64$. The training was conducted using Adam optimizer with learning rate $1e^{-3}$ and mean square error (MSE) as an optimization criterion. During training, 64 batch-size was updated per epoch, while one feature was randomly dropped. Validation and Test were carried out on 8 batches per epoch. For algorithm efficiency, normalization of data was performed where we used MinMax transformation tool.
Unlike the model described in paper, where the parameters were set of only one PV variable, we have adapted the method to work with multivariate inputs, in our case the weather data. As for the reconstruction block, which is not well explained in the paper, we adapted the full connected layer in order to return and predict the complete inputs necessary for the prediction part.
Finally, the test results present an efficient way to recover the missing signal, performing well reconstruction for every feature of the dataset (see Fig.5) We also can conclude that the Wind and Solar parameters are recovering in the best way.

However, during the process of the model implementation we revealed several disadvantages of this approach:

- The amount of missing input data can not be controlled in a proper way - in order to keep the number of d_f features for the model input we have to skip the whole signal of a random feature per batch. Thus, the amount of missing data is controlled only be the length of the batch.

- We find the approach of the missing data creation too synthetic, which is not similar to the real-world data.
- Non-missing data is distorted during the prediction process.

The possible reason for the non-missing distortion is that there is not enough correlation between the various features, unlike the original article, where the features are the previous values of the same variable and thus affect the summation of the convolution during prediction. So the non-missing data cannot keep the original value, and we receive a noisy data in the output.

Although we have an acceptable result for the missing data created for this task, the proposed model is almost impossible be applied to the real data (the number of missing features should be always the same). Considering this and other listed disadvantages, we attempted to make the model more flexible and adaptive to the real-world input data. Instead of skipping the feature as authors of the original paper, we implemented the mask to indicate the missing and real data points. This way, the number of features is always d_c (complete). The advantage of the masking method is the opportunity to change the missing data rate in an intuitive way.
However, with all the experiments with the model tuning, such a method does not provide a reliable data imputation: the final values were too over-scaled, and the training process experienced overfitting (see example in Fig.6).

Finally, we concluded that it is hard to estimate its robustness for data imputation problems for this model as authors have hidden too many essential details of its configuration and provided suspiciously too good results for their particular task. These outcomes forced us to define a more reliable solution to our problem.

5.1.2. Generative Adversarial Imputation Net (GAIN)

The idea behind the GAIN model was to re-implement the original model from Yoon et al. (2018) to reconstruct our missing weather dataset and make sure the non-missing inputs retain their original values during prediction.
The Generator and Discriminator of the GAIN model have the same 3 hidden fully connected layers with the ReLU activation function. In both cases, the sigmoid activation function was used as an output, where, in the case of the generator, as the normalized output, and in the case of the discriminator, as the output of the probability. For both models, we used Adam's optimization algorithm with a

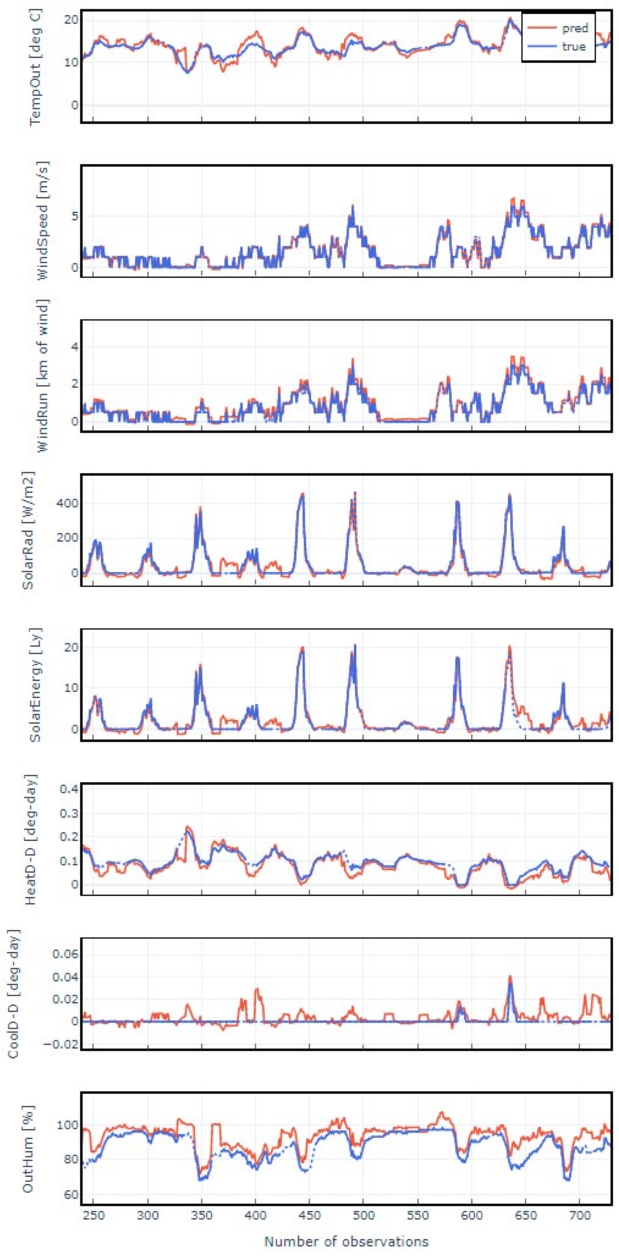


Figure 5. Results of the filling the missing data points with the SRPCNN model. Blue color: ground truth data with missing data points by dash. Red color: restored signal.

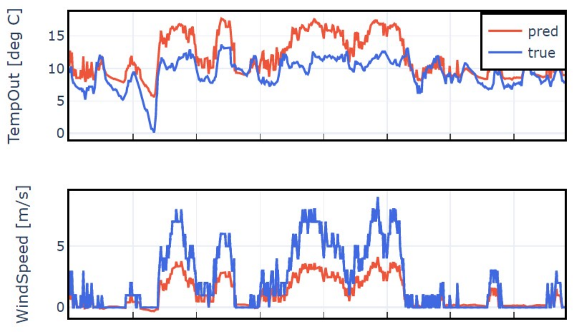


Figure 6. Results of the filling the missing data points with the second adaptation of the SRPCNN model for two weather data signals. Blue color: ground truth data. Red color: restored signal.

learning rate of 1e3 and mean square error (MSE) and cross entropy as an optimization 1 criterion. The best test results are shown at Fig. 7.

All these parameters were selected experimentally and perform the best in comparison with the rest of the experiments. In addition to the model introduced Hui (2018), we also added batch normalization to the generator and discriminator. This allowed us to increase the rate of the error convergence, but still the overall test result was worse. Adding a dropout also did not significantly improve the result. In our experiments the best results were obtained using the 16 - 64 - 32 - 8 architecture, while the authors of the original paper used bigger numbers, but they also had much more features.

In Yoon et al. (2018) the generator error is considered as the least squares method. In doing so, the authors compare the non-omitted values. Since we have the original full dataset, we decided to check what will happen if we compare not the unknown values, but those which were skipped and then restored. In general, the results turned out to be quite logical, with a not large percentage of lost data, the model shows the best result when calculating the error using the proposed equation(3), in the opposite case - MSE with not missing data(4).

$$MSE_loss = \frac{(M * X - M * \bar{X})^2}{mean(M)} \tag{3}$$

$$MSE_loss = \frac{((1 - M) * X - (1 - M) * \bar{X})^2}{mean(1 - M)} \tag{4}$$

The error achieved in the initial position varied from 0.05 to 0.14 when passing through different datasets, in our case the error is 0.09.

The significant advantage of the proposed model is that non-missing data can completely retain its values during prediction as they are stored in the input mask. This advantage makes it to be more promising approach for working with the real-world data than SRPCNN model for multidimensional inputs. Also, this model can control and learn the different amount of missing data simultaneously. One can see in Fig.7 that missing data forecast does not always recover very well, especially for wind data where some samples have huge picks.

5.2. Frameworks for the PV Generation Prediction

5.2.1. Long short-term memory (LSTM)

In the following research the model consists of two LTSM layers, each followed by a dropout layer to avoid overfitting and a FC layer for prediction. It can be used for both one time-step forward and multiple time-steps predictions, depending on the set parameters. 60 past values were passed to the model to predict for 1 or 10 steps for every feature.

In total, 4 LSTM configurations were tested: one-step and multiple-step predictors with or without the bidirectional LSTM layer. The higher number of recurrent layers resulted in higher loss.

The best results according to the Mean Square Error (MSE) metric were achieved with unidirectional layers only (0.029 for unidirectional vs 0.0529 for bidirectional model, 80 epochs of training). Therefore, unidirectional model was chosen as the baseline.

Model training was completed on the real weather dataset. The trained model was tested on both real data and data recovered with GAIN. For the MSE loss for 10 steps prediction is 0.034, and 0.023 for one step prediction.

However, the multiple time-step LSTM almost doesn't show the trend of future values, and predicts the almost constant averaged values for each feature. The example on normalized feature prediction for 10 steps is shown on 8.

One-step predicting LSTM trends to understate the predicted value in comparison to the true one and also smoothen the peaks. The forecasted features examples are on 9 For concistency, one-step predictions were used in future analysis.

5.2.2. Transformer

We run 2 Transformer models - Transformer-based forecasting model and FFT based model.We tried different hyperparameters - number of encoder/decoder layer from 1 to 5, number of heads [2, ..., 4], lerning rate [$1e - 9$, ..., $1e - 2$], optimizer [Adam, SGD, Adagrad], number pf epochs [30, ..., 100]. According our experiments optimal parameters for both models are the same and they are the following:

- Num of encoder layers - 3
- Num of decoder layers - 3

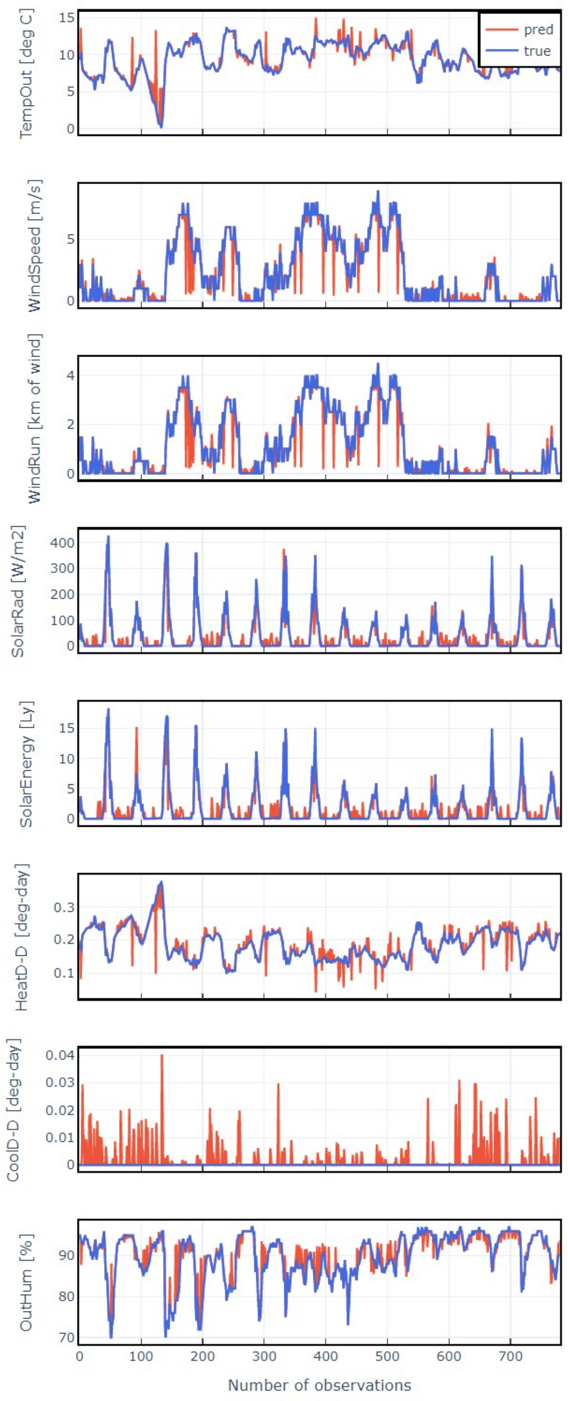


Figure 7. Results of the filling the missing data points with the GAIN model. Blue color: ground truth data. Red color: restored signal.

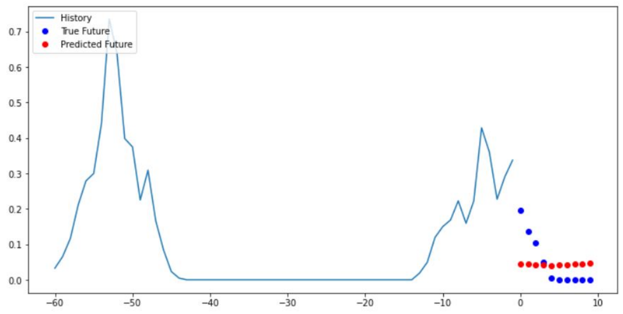


Figure 8. Results of the LSTM prediction on the Solar Radiation normalized feature.

- Num of heads - 3 (for fft based model is not applicable due to architecture)
- Size of attention - 64
- Hidden size - 10
- Learning rate - $1e-3$
- Optimizer - Adam with weight decay=1e-3
- Num of epochs - 50
- Criterion - MSELoss

	T	**T-FFT**
MSE	0.0045	0.006
Train time	57.63	49.2

Table 1. Transformer models performance on Real Weather. 'T'- Transformer model, 'T-FFT' - Transformer with FFT model. Training time measured in seconds.

	T	**T-FFT**
MSE	0.003	0.005
Train time	49.32	43.70

Table 2. Transformer models performance on Restored Weather. 'T'- Transformer model, 'T-FFT' - Transformer with FFT model. Training time measured in seconds. Blue color: ground truth data. Red color: predicted signal.

Transformer models show quite good behaviour in time series prediction task. We can observe quite good prediction power (Fig. 10 and Fig. 11). Besides, FFT based network trains faster with less loss of quality (Table. 1 and 2). During conducting experiments we observe till 2 times speed-up by FFT based network. Due to our project dataset consist of thousands, and not billions points, training time differ is not stunning, but still during experiments we observe only less training time.

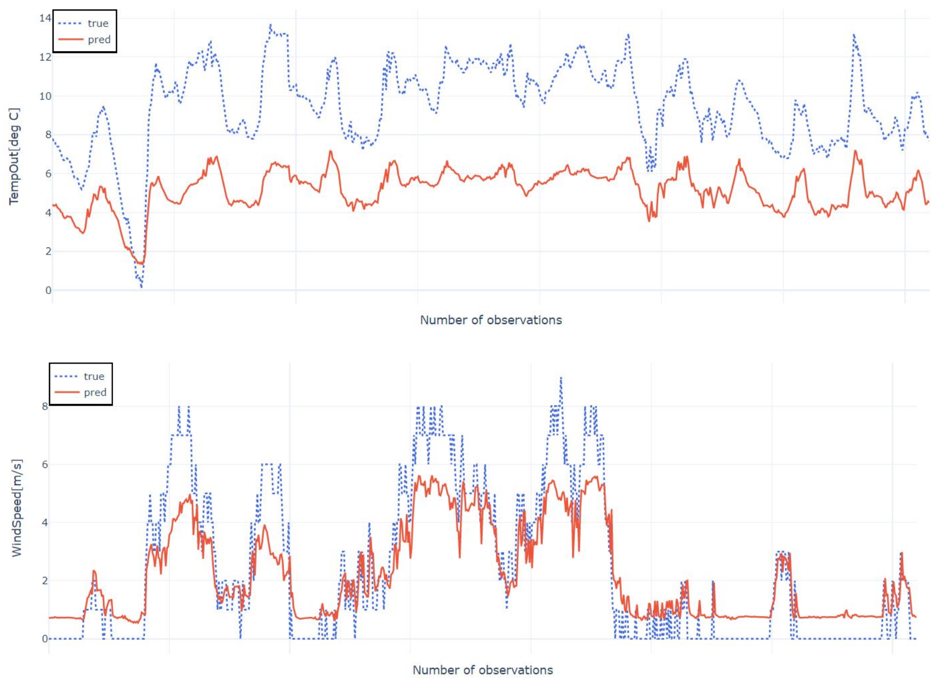


Figure 9. Results of one-step forward weather forecasting with LSTM model. Blue color: ground truth data. Red color: predicted signal.

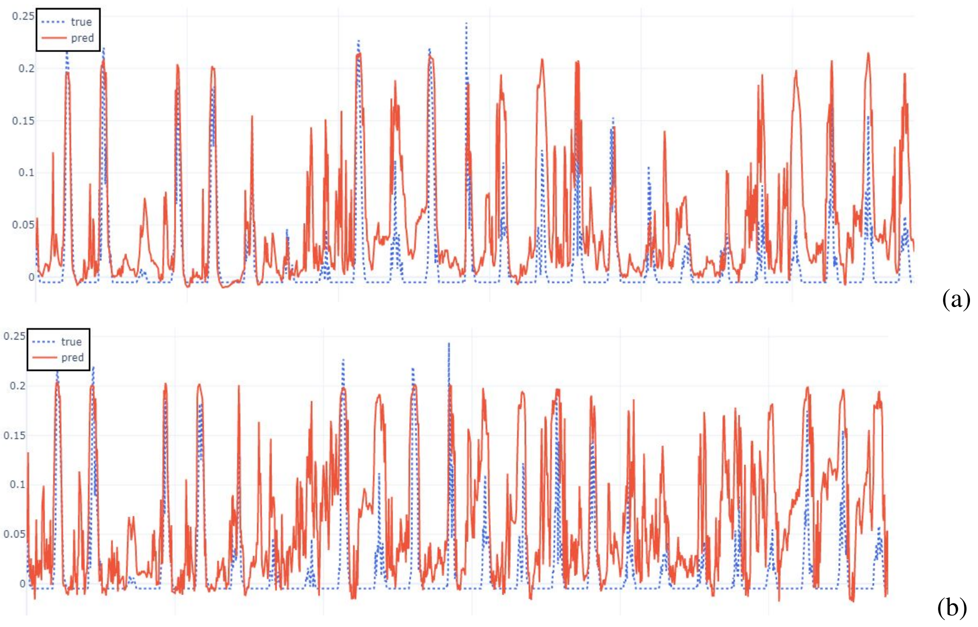
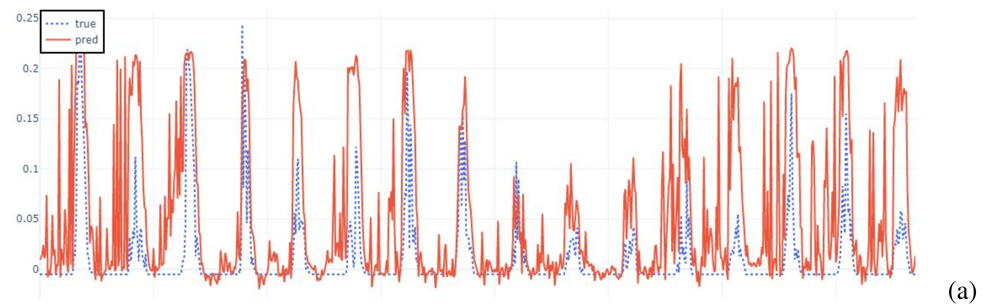


Figure 10. (a) PV prediction based on Real Weather by Transformer. (b) PV prediction by Transformer with FFT blocks based on Real Weather. Blue color: ground truth data. Red color: predicted signal.



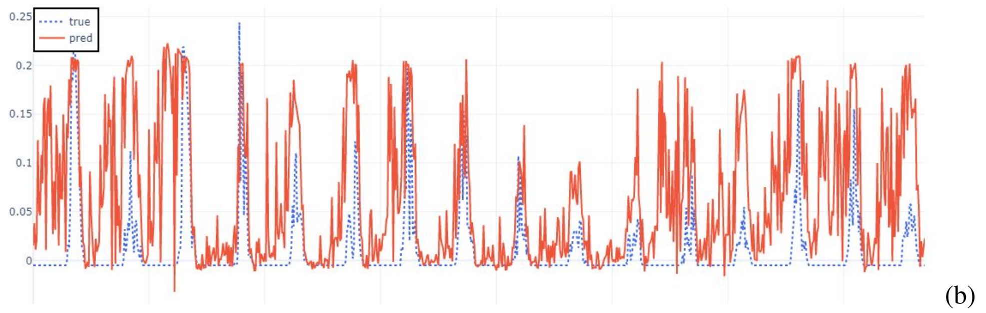


Figure 11. (a) PV prediction based on Restored Weather by Transformer. (b) PV prediction by Transformer with FFT blocks based on Restored Weather.

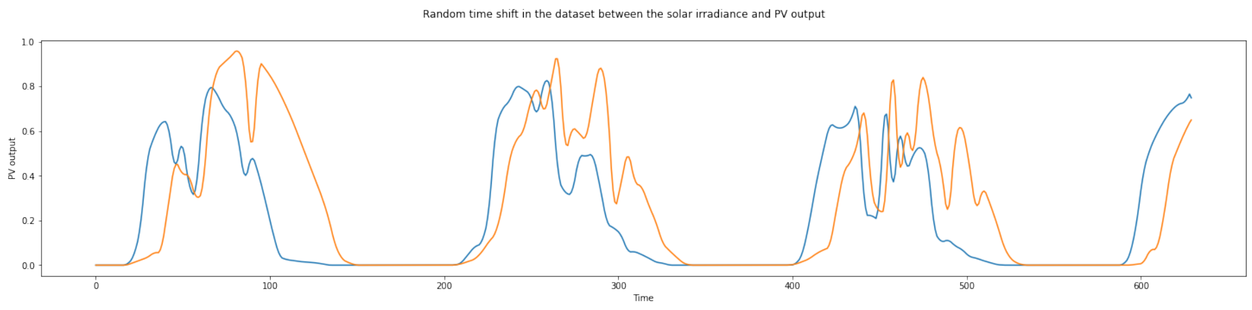


Figure 12. Arbitrary time shift in the dataset between the solar irradiance (yellow) and PV output (blue)

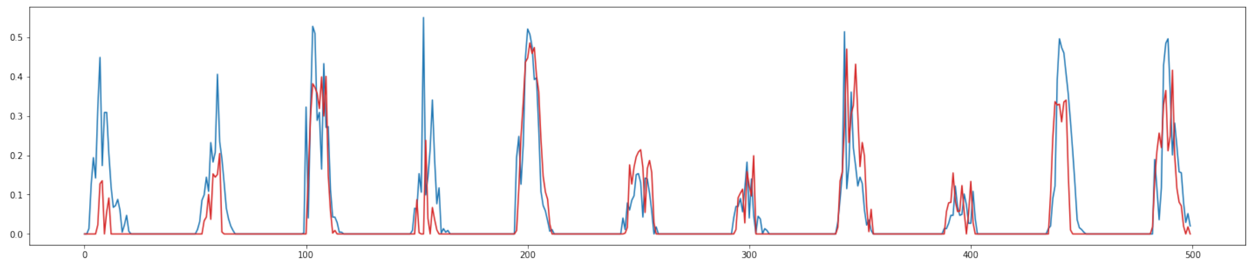


Figure 13. PV prediction (red) and PV ground truth (blue)

5.2.3. PV power prediction

The results of the model implementation are shown on Fig.12. The main problem here is an arbitrary time shift in the dataset between the solar irradiance and PV output. It is remarkable here that a PV power appears before the solar irradiance, that is not possible in the reality, thus the causal relationship was broken and it affects on the ability of any machine learning model here to correctly predict the PV output based on the dataset used. Moreover, the time step between the measurements is too big (30 minutes) and as a consequence the input signal, containing in 8 features, is too rough and chopped. In order to solved this problem, different architectures of model and training conditions were tested, including 1-d convolution. All of them demonstrated the same result and mismatching between the predicted and real PV output.

Additionally, the linear interpolation of the features signal was applied to make a signal smoother, but results improved not significantly. Finally, the 4 times interpolated signal was used for the model training while tests were conducted on a raw data. Rectified Linear Unit activation function was used at the output of the model to prevent the negative values of the PV power prediction. Resulting validation Mean Square Error (MSE) loss depends on the batch size. The best training performance for the interpolated train data was achieved with the train batch size equaled to 4.

Eventually, developed and trained linear model for the PV output prediction demonstrates visually adequate results and correctly operates on signals from the all models were used at the previous step in out pipeline (LSTM, Transformer). The best resulting MSE error was 0.0107 in the validation test with a batch size equals to 30 and real weather input data (not restored). Comparison between the PV ground truth and prediction value for the raw real weather data is shown in the Fig.13.

6. Conclusion

Our project was devoted to the reconstruction of the missing weather data and PV generation forecasting.

We conducted a number of experiments and modified the original models as well as implemented and developed several different architectures and their adaptations in order to achieve the best results for the dataset used (UKPowerNetworks). During the process, we revealed the advantages and disadvantages of every model implementation. For the missing data the most appropriate model is GAIN. Although data recovery in the SRPCNN model is better, GAIN model allows us to work with conditions close to the real life. As further improvements to this model, we plan to also try a convolution layer.

As a baseline, different topologies os LSTM-based models were tested as a predictor. Although they do predict the future values, the results are not satisfactory enough for the precise generation prediction

Transformer models achieved good prediction power in our time series forecasting task. Moreover, we applied recently presented method related to Transformer model based on Fast Fourier Transform.

References

Hui, J. Gan — ways to improve gan performance, 2018.

Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. Fnet: Mixing tokens with fourier transforms, 2021.

Liu, W., Ren, C., and Xu, Y. Pv generation forecasting with missing input data: A super-resolution perception approach. *IEEE Transactions on Sustainable Energy*, 12 (2):1493–1496, 2021. doi: 10.1109/TSTE.2020.3029731.

Lo Brano, V., Ciulla, G., and Di Falco, M. Artificial neural networks to predict the power output of a pv panel. *International Journal of Photoenergy*, 2014, 2014.

UKPowerNetworks. Photovoltaic (PV) Solar Panel Energy Generation data. London data link. Published: 2014; Accessed: May 11, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Wang, D. and Li, M. Stochastic configuration networks: Fundamentals and algorithms. *CoRR*, abs/1702.03180, 2017.

Wu, N., Green, B., Ben, X., and O'Banion, S. Deep transformer models for time series forecasting: The influenza prevalence case. *CoRR*, abs/2001.08317, 2020. URL `https://arxiv.org/abs/2001.08317`.

Yoon, J., Jordon, J., and van der Schaar, M. GAIN: missing data imputation using generative adversarial nets. *CoRR*, abs/1806.02920, 2018.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Olga Klyagina

Made the LSTM models; interpreted and explained the dataset and participated in its preparation; assisted on the implementation of SRPCNN model; coordinated the team.

Ekaterina Dorzhieva

Implemented the GAIN model accordance with the article Yoon et al. (2018). Conducted various experiments to adapt the original model to the dataset (UKPowerNetworks). Worked with development of SRPCNN model.

Mile Mitrovic

Worked on the implementation of SRPCNN Liu et al. (2021) and GAIN model Yoon et al. (2018) from scratch, and set up their experiments. Participated in dataset preparation (UKPowerNetworks)

Mikhail Kirillov

Worked on weather prediction architectures, conducted Transformer related experiments. Performed the final presentation video.

Galina Chikunova

Implemented SRPCNN model from scratch in accordance with related paper Liu et al. (2021), tuned the model parameters in a series of experiments. Worked on the modification of the proposed approach. Assisted in the implementation of GAIN model.

Anastasia Sozykina

Implementation of Transformer-based forecasting model and Transformer FFT-based model. Conducting Transformer related experiments. Participating in first-stage Missing Data Imputation.

Aleksei Shcherbak

Preparing, processing and time merging of datasets, implementation and training of the linear PV forecasting model, analyzing the dataset problems (time shift), testing the results from the previous pipline models (LSTM, Transformer).

B. 3rd party code list

- time series with lstm
- `https://habr.com/ru/post/495884/`
- `https://pytorch.org/tutorials/beginner/transformer_tutorial.html`
- `https://github.com/jaketae/fnet`