NAME : Christian Merz
COUNTRY : Germany / Belgium
REGISTRATION NUMBER : DLG6351
GROUP REF. : SC D2
PREF. SUBJECT : PS1
QUESTION N° : 1.2

Build up trust in time series prediction by using Explainable AI to comply with upcoming EU AI regulation.

Time series forecasting plays a crucial role to operate the transmission grid safely and in a cost-effective manner. As forecasts are never 100% accurate, an operator needs to know to which extent a forecast can be trusted. To build up trust, it is key to provide transparency on the model by offering easy-to-understand insights on its functioning and its output.
Therefore, often simple models (e.g. regression models) are accepted in system critical infrastructures as the model and its parameters can be understood easily. Beside the fact that these models are reliable and have themselves proven in production, the complexity in the grid as well as the need to make more accurate predictions have been on the rise. More complex models might increase the accuracy but often are a black-box and lack an easy-to-understand interpretability of the results. This makes it difficult to build trust to the operators.
Up to now, the decision which kind of models are used in a system critical environment depends on the internal risk assessment or possibly on local regulatory authorities.
This might change soon as the European Commission published a framework on AI that would make it obligatory to create transparency on forecast models that operate in a high-risk environment.[1]

In the following, the paper examines the requirements on transparency that come with the new EU AI regulation and proposes an easy-to-understand approach for various ML models that enables non-technical users to understand the impact of each input feature on a time series prediction.

The proposal on a European AI Framework

On April 4th, 2021 the European Commission published a proposal on the first-ever legal framework on AI to build trust and mitigate the risks of AI applications.[2] The framework aims at ensuring that AI systems are safe, trustworthy and lawful as well as setting a regulatory framework and governance.[3] The framework is based on a risk-based approach that differentiates between (i) unacceptable risk, (ii) a high risk, and (iii) low or minimal risk. Applications that run in a system critical infrastructure like energy transmission grid operations fall under (ii) and therefore legal requirements in regards to data, documentation, logging, transparency, provision of information to users, robustness, accuracy, security and human oversight apply.[4]
Each provider of high-risk AI systems needs to ensure that its systems are compliant with all requirements[5] stated in Chapter 2 which, focusing on transparency and interpretability, include

[1] EC – COM(2021) 206
[2] Regulatory framework proposal on artificial intelligence | Shaping Europe's digital future (europa.eu). The requirements are based on the recommendation of the high-level expert group on artificial intelligence – Ethics guidelines for trustworthy AI
[3] EC – COM(2021) 206 p 3
[4] EC – COM(2021) 206 p. 13
[5] EC – COM(2021) Chapter 3, Article 16

the *responsibility for transparency*[6], providing *human oversight*[7] as well as the *obligation to inform the user that they are interacting with an AI system*.[8]

High-risk AI systems shall ensure that the system's output is sufficiently transparent to enable users to interpret the results (*responsibility for transparency*)[9] as well as providing an appropriate human-machine interface tool to oversee the AI system during its operations (*human oversight*).[10]
As most of these legal requirements are additional steps to put in place, an open-source, easy-to-understand interface to provide transparency and explicability to the users for time series forecasting is still lacking.

Using Explainable AI to provide insights in time series predictions

In the following, one approach is shown how to fulfil the requirements on transparency and explicability by using existing open-source explainable AI libraries.
There are several open-source frameworks that enable interpretability of ML models like "Shapley values" and "lime" with explainer methods for tree-based, linear and neural networks.[11] These frameworks offer great interpretability and visualization for various use-cases like classification problems and image recognition but lack in visualization for time series prediction.
With this approach, the goal is to explain the impact of each input feature on each time step on the final prediction to the operator by using the existing SHAP framework. This aims at understanding the impact of each feature on the overall forecast and also building trust as it can confirm patterns that a human knows by experience.[12]

Applied Method to provide interpretability to time series forecasts

In this approach, the SHAP library, implemented in Python, has been selected as it provides local and global interpretability for various ML models.[13] For the following method to work, it is required that a multi-variate time series ML model has been trained and has made its predictions.[14]
As an example, a multi-variate grid load forecast on the Belgium grid with XGboost is chosen. Additional features include solar radiation, wind speed, temperature as well as periodicity. The original data, the model as well as the prediction are passed to SHAP-script which generates a validation and a prediction plot with SHAP values.
In the validation plot shown in Figure 1, each feature and its respective positive or negative impact on the forecast is plotted as a stacked bar plot. All SHAP values of single time step added together equal to the predicted value for the respective time step. The base line is the

[6] EC – COM(2021) Chapter 2, Article 13
[7] EC – COM(2021) Chapter 2, Article 14
[8] EC – COM(2021) Title IV, Article 52
[9] EC – COM(2021), Chapter 2 Article 13 No. 1
[10] EC – COM(2021), Chapter 2 Article 14 No. 1
[11] An introduction to explainable AI with Shapley values — SHAP latest documentation ; [1602.04938] "Why Should I Trust You?": Explaining the Predictions of Any Classifier (arxiv.org) ; GitHub - marcotcr/lime: Lime: Explaining the predictions of any machine learning classifier
[12] E.g. seasonal patterns: consumption on week days is higher than on weekends
[13] SHAP stands for SHapley Additive exPlanations and is based on a game theoretic approach to explain the output of a machine learning model A Unified Approach to Interpreting Model Predictions (neurips.cc)
[14] A list of ML models that are supported by SHAP can be found here: Tabular examples — SHAP latest documentation

extracted expected SHAP value of the forecast. The same can be seen in Figure 2 (excluding the actuals).
The advantage with this visualization is the option for a user to understand how the model made the prediction for each time stamp and confirm patterns that are known by experience. If the forecast deviates from the actuals, it can then easily be understood which feature impacted the results. In a next step, it can be analyzed whether the given input features were not selected correctly or if additional parameters could be included to possibly improve the forecast.

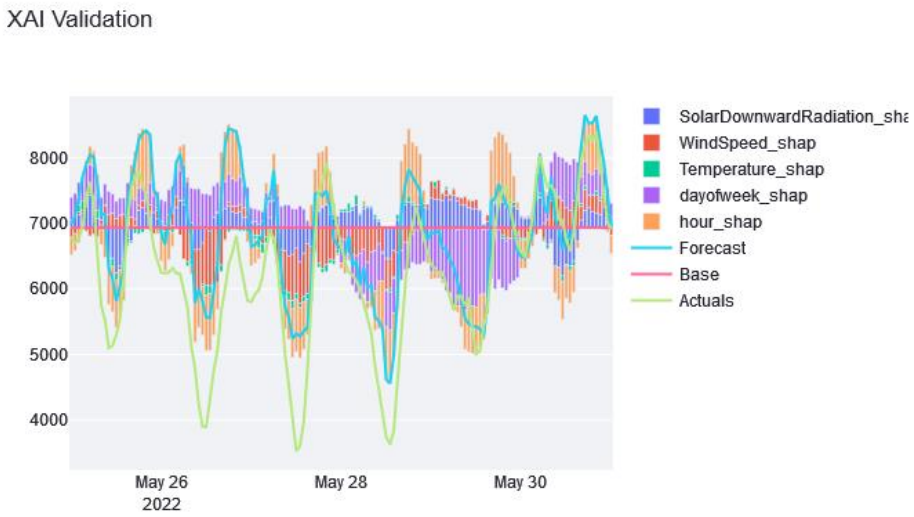

Figure 1: SHAP Values calculated and visualized on the validation data set.

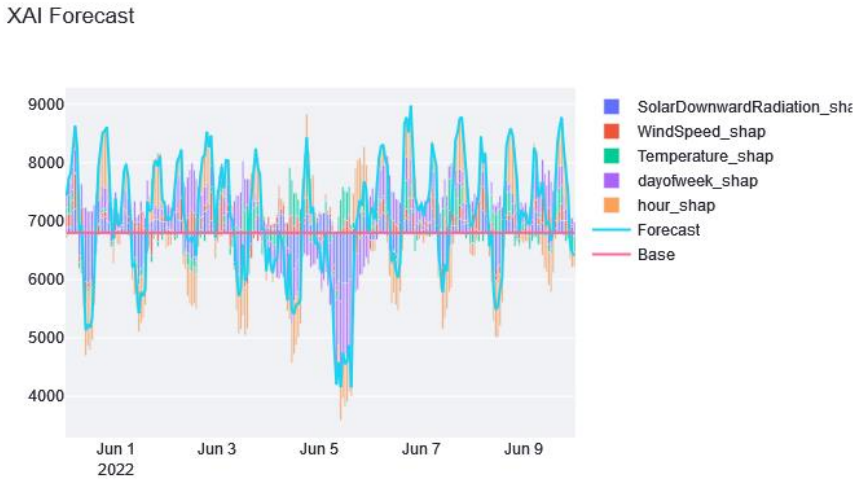

Figure 2 SHAP Values calculated and visualized on the prediction

Summary & Outlook

The explained approach can be one approach to build trust in time series predictions as well as to comply with transparency requirements stated in the proposal of the EU AI regulations.
Next steps include further refining the graphical user interface, by adding accuracy and error metrics[15], testing different supported ML models as well as receiving more feedback from users.

[15] Additional parameters, independent from SHAP, like accuracy metrics (e.g. MAE, MAPE) and other validation methods (e.g. backtesting on older time data) could be added. Furthermore, both feature importance of SHAP as well as permutation feature importance of the model can be integrated.