IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 3, 2020, accepted October 7, 2020, date of publication October 15, 2020, date of current version October 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031477

Gaining Insight Into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools

MURAT KUZLU[1], (Senior Member, IEEE), UMIT CALI[2], (Member, IEEE), VINAYAK SHARMA[3], AND ÖZGÜR GÜLER[4]

[1]Department of Engineering Technology, Old Dominion University, Norfolk, VA 23529, USA
[2]Department of Electric Power Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[3]Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223, USA
[4]eKare Inc., Fairfax, VA 22031, USA

Corresponding author: Murat Kuzlu (mkuzlu@odu.edu)

This work was supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber research and development, innovation and workforce development in Virginia.

ABSTRACT Over the last two decades, Artificial Intelligence (AI) approaches have been applied to various applications of the smart grid, such as demand response, predictive maintenance, and load forecasting. However, AI is still considered to be a ''black-box'' due to its lack of explainability and transparency, especially for something like solar photovoltaic (PV) forecasts that involves many parameters. Explainable Artificial Intelligence (XAI) has become an emerging research field in the smart grid domain since it addresses this gap and helps understand why the AI system made a forecast decision. This article presents several use cases of solar PV energy forecasting using XAI tools, such as LIME, SHAP, and ELI5, which can contribute to adopting XAI tools for smart grid applications. Understanding the inner workings of a prediction model based on AI can give insights into the application field. Such insight can provide improvements to the solar PV forecasting models and point out relevant parameters.

INDEX TERMS Explainable artificial intelligence (XAI), solar PV power generation forecasting, explainability and transparency.

NOMENCLATURE

$\hat{y}$	The predicted value of y
$\mathcal{L}$	Fidelity function
Ω	Complexity measures
ϕ_j	Feature attribution for a feature j
π_x	Proximity measure
ξ	LIME explanation model
e	Explanation for the model
$ELI5$	Explain Like I'm 5
f	The model being explained
G	A set of interpretable model $g \in G$
$HOUR$	The hour of the day
HUM	Relative humidity
k	Feature
$LIME$	Local Interpretable Model-agnostic
M	The number of input features
N	The number of sample
P_{acc}	The accumulated value
P_{ave}	The average value
PV	Photovoltaic
RFR	Random Forest regressor
$RMSE$	Root-mean Square Error
S	A set of non-zero indexes in z'
$SHAP$	SHapley Additive exPlanations
SP	Surface pressure
$SSRD$	Surface solar rad down
$STRD$	Surface thermal rad down
TCC	Total cloud cover
$TCIW$	Total column ice water
$TCLW$	Total column liquid water
$TEMP$	2 metre temperature
TP	Total precipitation
TSR	Top net solar rad
U	10 metre U wind component
V	10 metre V wind component
XAI	Explainable Artificial Intelligence
$XGBoost$	eXtreme Gradient Boosting

The associate editor coordinating the review of this manuscript and approving it for publication was Long Wang.

y The actual value
Z A set of all input features
z' The coalition vector

I. INTRODUCTION

In recent decades, the world's energy consumption has been on the rise. This has led to a global concern regarding future energy demand as well as shifting to more sustainable sources of energy to meet this growing need [1]. Steps have been taken to tackle the concerns of the modern electric grid and to increase the efficiency and reliability of the electric grid. One such step is to use Artificial Intelligence (AI) methods in smart grid applications. AI methods have been used in electric network operation and control [2], energy management and control [3], demand response [4], predictive maintenance [5], energy generation [6] and load forecasting [7]. AI methods have been critical in the modernization of the electrical grid and making it a ''smart'' grid. Nevertheless, AI is still considered as a black-box method because of the absence of a simple understanding of the inner workings of the fundamental models. Numerous utility engineers in the energy industry are hesitant to deploy AI-based techniques considering their lack of insight and explainability, which can help understand their dynamic decision-making procedure. However, Explainable AI (XAI) addresses this concern by increasing the explainability and transparency of the AI models and thus opening the black-box. An extensive review of XAI was provided in [8] as far as concepts, taxonomies, opportunities, challenges, and adopting XAI tools. The Defense Advanced Research Projects Agency (DARPA), in 2017, introduced an XAI initiative with the aim to deliver AI techniques with more explainable models in order to understand, trust and adequately deal with rising AI applications [9]. Numerous applications of AI in the smart grid can be found in the literature. The authors in [10] proposed a high-precision deep neural network model, i.e., PVPNet, utilizing meteorological information, such as temperature, solar radiation, and historical PV system output data, for day-ahead solar PV generation forecasting. The purpose of this study is to focus on solar PV forecasting using XAI tools. There are three reasons for focusing on this application. To start with, there has been a remarkable rise in the adoption of solar PV in the U.S., with more than 1 million solar installations, totalling to 71.3 GW in capacity [11]. This increasing installation capacity of solar PV has led to a need to reassess traditional forecasting algorithms, which do not consider weather conditions, such as cloud cover, irradiance, etc. Those can drastically affect the accuracy of PV forecasts [12]. Secondly, not many studies have explored XAI in energy forecasting, and third, XAI has mostly been explored for text and image data and not so much with time-series data.

XAI has been explored more in areas where explainability and transparency of the model's working is critical, such as healthcare-stroke detection [13], cybersecurity-Intrusion Detection Systems (IDS) [14], military-target classification [15], and finance-risk management [16]. XAI based models have also been used in smart grid applications. In [17], the authors present a short-term electricity load forecasting study with generalized additive models to enable the integration of both a regressive part with explanatory variables (weather, calendar variables, and global trends) and an auto-regressive part with lagged loads. Authors in [18] propose an agent-based deep reinforcement learning algorithm using an XAI approach to control an energy storage system. The learning progression of an agent was explained, i.e., an efficient dispatch algorithm for the energy storage device under variable tariff-structures. In [19], the authors applied XAI techniques to interpret the load forecasting output of a gradient boosting algorithm (XGBoost) [20] and show the analysis in SHAP (SHapley Additive exPlanations) [21]. In this article, we present multiple forecasting methods to predict solar PV energy generation using three XAI tools, namely, LIME, SHAP, and ELI5. The XAI tools help explain how much an input feature contributes to the forecast. The use of feature engineering is employed in the preprocessing stage. XAI brings more insights and explanations in other stages of the process, such as during model operation and after (post-modelling). A joint approach, explainable feature engineering, provides the additional potential to manage the dimensions and effectiveness of the input parameters by considering the AI/ML model and domain-specific details.

AI has been used with the context of energy systems and the energy market domain for at least two decades. It is indicated that many practitioners, academicians, and researchers in the domain consider the AI-based system as a ''black-box,'' which might lead them to oversee many important and influencing parameters in their modelling framework. Solar forecasting can also be considered a good sample territory where XAI may bring additional insights and explanations regarding various variables, which will transform the ''black-box'' model to a type of ''grey-box'' model. According to the comprehensive literature review, this study is one of the first publicly available resources, which proposes XAI-based solar PV power generation forecasting.

II. MACHINE LEARNING MODELS AND EXPLAINABLE ARTIFICIAL INTELLIGENCE TOOLS

This section discusses the machine learning models and XAI tools used in this study.

A. MACHINE LEARNING MODELS

1) RANDOM FOREST REGRESSION (RFR)

Random forest is an ensemble machine learning technique used for supervised learning. It can be implemented for classification as well as regression problems. Multiple decision trees are trained using randomly sampled data from the input. The meta-estimator combines the predictions from the decision trees [22]. For this article, a random forest regressor is implemented to forecast PV power generation using the Scikit-learn library in Python [23]. Random forests are known to work well with tabular data. Random forest models can produce accurate results without having to aggressively fine-tune the model's hyperparameters [24].

B. EXPLAINABLE ARTIFICIAL INTELLIGENCE TOOLS

Due to the increasing use of artificial intelligence and machine learning and recent dependence on these technologies, XAI has become an emerging field of study. A variety of XAI tools have been developed by researchers across various fields to help understand AI-based black-box models. LIME (Local Interpretable Model-Agnostic Explanations) [25], uses an interpretable model to approximate any AI model. SHAP (Shapley Additive exPlanations) [26] helps to explain models, and the features that are important in building the model using Shapely values. ELI5 [27] helps to explain various regression and classification models and their implementations in Python. MLxtend (machine learning extensions) [28] offers a solution to better understand popular machine learning libraries in Python. Skater [29] provides a solution to understand the learning structure of various machine learning libraries in Python. InterpretML [29] is a tool developed by Microsoft that helps explaining machine learning models as well as provides a new model called Explainable Boosting Machine (EBM). TreeInterpreter [30] is a tool to understand tree-based ensemble models. Alibi [31] provides explanations and insights into machine learning models. In this work, we explore and implement LIME, SHAP, and ELI5 for solar PV power forecasting.

1) LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS

As the name suggests, Local Interpretable Model-agnostic Explanations (LIME) is a tool to understand and interpret the underlying machine learning model while remaining model-agnostic. LIME was introduced by [25], with the idea of approximating the machine learning model with a model that can be understood. This is done locally since it can be easier to understand and approximate complicated machine learning models globally [32]. The explanations provided by LIME will enable users to understand and interpret the model. LIME defines explanations in the following manner:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{1}$$

where, G is the set of interpretable models, $\Omega(g)$ defines the complexity of the explanation of all $g \in G$. The aim should be to have low $\Omega(g)$ so as to have a simple model that can be interpretable. The black-box model that is being explained is denoted by f. π_x is the proximity measure, which defines the size of the neighborhood around instance x, and $\mathcal{L}(f, g, \pi_x)$ is the measure of how close the explanation model g to the prediction of the original model f, i.e., fidelity. The final goal is to minimize $\mathcal{L}(f, g, \pi_x)$ and to get a interpretable approximation of the black-box model. As the name suggests, LIME tries to minimize $\mathcal{L}(f, g, \pi_x)$ while being model-agnostic. It presents local models that approximate the black-box model globally [25], [33].

2) SHapley ADDITIVE exPlanation

Introduced in [34], SHapley Additive exPlanation (SHAP) uses Shapely values to explain the contribution of each feature to the prediction [35]. SHAP uses the coalitional game theory defining how well each group (or coalition) of agents can do for itself. SHAP is defined as:

$$e\left(z'\right) = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{2}$$

where ϕ_j is the attribute of the feature j, z' denotes the coalition vector, i.e., if the feature is present ($z' = 1$) or absent ($z' = 0$). M denotes the number of input features. e gives the explanation for the model. SHAP, in order to compute shapely values, assumes only some features values are playing, i.e., present, and some are not, i.e., absent [35], [36]. By doing this, SHAP identifies how much each feature contributes to the prediction. To compute SHAP values for model f, with S denoting a subset of features, with Z denoting the set of all input features, with ($z' = 1$) and $E[f(x) \mid x_S]$ denoting expected value of the function conditioned on a subset S of the input features, SHAP values from game theory to attribute ϕ_j values to each feature can be defined as a value function of players in S [36]:

$$\phi_j = \sum_{S \subseteq Z \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[f_x(S \cup \{j\}) - f_x(S)\right] \tag{3}$$

SHAP has a Python implementation, which offers a visualization tool for each feature and its importance. It works with tree-based models from Scikit-learn package in Python as well.

3) ELI5

ELI5 is a Python package that aims to explain black-box machine learning models in Python. ELI5 gives the weights associated with each feature to depict the feature's importance in the machine learning model. ELI5 is implemented for most of the commonly used Python-based machine learning packages, such as Scikit-learn, Keras, and XGBoost [27]. Unlike LIME, ELI5 is not model-agnostic, and it has its own implementation of XGBoost [37].

III. DATA COLLECTION AND PREPROCESSING

A popular open-source benchmark dataset from the Global Energy Forecasting Competition (GEFCOM) held in 2014 [38] is used for this work. The reason for selecting an open-source dataset is to make the work easily reproducible. The data consists of hourly solar power generation data and corresponding numerical weather forecasts from April 1^{st}, 2012 to July 1^{st}, 2014. In this work, the data contains the following weather variables from the European Centre for Medium-Range Weather Forecasts (ECMWF):

1) *TCLW* (kg m**-2)
2) *TCIW* (kg m**-2)
3) *SP* (Pa)
4) *HUM* (%).

5) *TCC*
6) *U* (m s**-1)
7) *V* (m s**-1)
8) *TEMP* (K)
9) *SSRD* (J m-2)
10) *STRD* (J m-2)
11) *TSR* (J m-2)
12) *TP* (m)
13) *HOUR* (h)

A. DATA PREPROCESSING

SSRD, STRD, TSR, and TP are accumulated fields, i.e., the values are accumulated hourly throughout the day. The values need to be preprocessed in order to get average values. The following formula is used to get average values:

$$P_{ave}(k) = \frac{P_{acc}(k+1) - P_{acc}(k)}{3600} \tag{4}$$

where, $P_{ave}(k)$ is the average value and P_{acc} denotes the accumulated value [39].

B. TRAIN AND TEST DATASET AND VALIDATION

In order to select the best parameters for the model and prevent overfitting, the dataset was split into two primary sets, i.e., training and testing set. We extracted out 30% of our data as test data, while the remaining 70% was used as training data. The test data was not used during the training phase, except for final performance evaluation (error) of the applied models. Root Mean Squared Error (RMSE) is used as the error metric for the experiments. It is defined as:

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(y(n) - \hat{y}(n))^2}{N}} \tag{5}$$

where $y(n)$ denotes the actual solar power generation at time-step n and $\hat{y}(n)$ denotes the solar power forecast value at time n while N is the number of samples.

IV. IMPLEMENTATION OF XAI ON SOLAR PV GENERATION FORECASTING AND VALIDATION

The objective of this article is to apply XAI methods on solar PV power generation forecasting and to interpret "black-box" machine learning models so that it can be used in smart grid applications with a proper acceptance. The Random Forest Regressor (RFR) is considered as the base black-box model for this article. The RFR model is trained using the training data explained in the previous section. The hyper-paremeters of the RFR model are tuned to get maximum accuracy. The final RFR model is trained with 50 estimators. The base RFR model gives an RMSE of 7.23%, which is a decent result for this model. Figure 1 shows the plot of actual test data points versus the forecast. However, the main objective of this article is to use XAI tools to understand the underlying model and the impact of each feature on the forecasting results rather than the forecast itself. Below, we present the following XAI techniques, LIME, SHAP,

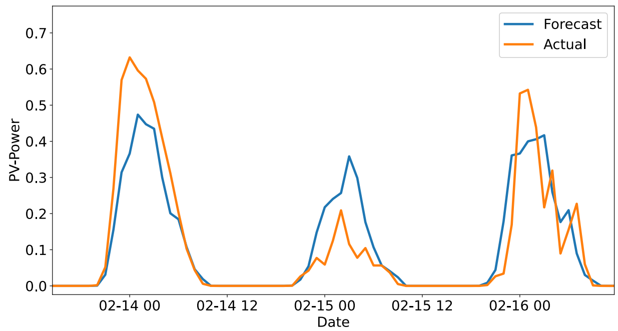


FIGURE 1. Actual solar PV power generation vs predicted solar PV power generation.

and ELI5, which can be employed for model interpretations and make the machine learning models understandable. Figure 2 shows the methodology of the work in the form of a flowchart. It includes three steps: (1) Solar PV data collection and preprocessing, (2) Black-box AI-based forecasting models, and (3) Applying XAI tools, i.e., LIME, SHAP, and ELI5.

A. APPLYING LIME XAI TOOL TO SOLAR PV FORECASTING

The LIME tool helps to identify an interpretable model over the interpretable representation locally. When we apply LIME for an explanation of individual predictions, it shows the solar PV power output forecasting results with each feature, as shown in Figure 3. Please note that LIME provides the explainability locally, i.e., explanation in the neighborhood of the prediction. *SSRD*, *HOUR*, and *TSR* are the most important features, while *TCWL*, *U*, and *TP* are the lowest in terms of numerical contribution. The contribution of each feature, either positive or negative, can be seen in the explanations, e.g., *SSRD* has a positive effect, while *TCIW* has a negative effect on predictions. Please note that the results obtained from LIME can be slightly different when we train the data again due to the stochastic nature of machine learning.

LIME is also capable of local interpretability of the models. Figures 4-6 show the local explanations for 3 hours of a day, i.e., 6th, 7th, and 8th. LIME results consist of three parts: (1-Left) Prediction probabilities of solar PV power generation forecasting, (2-Middle) The LIME explanation of selected features, and (3-Right) The original feature values. According to the results, the solar PV forecasted values are 0.731, 0.647, and 0.686, while the actual values are 0.774, 0.758, and 0.712, respectively. It provides a pretty good result for this instance. It can also be extended with more instances. For example, as shown in Figure 5, the solar PV power generation is predicted as 0.647. *SSRD*, *HOUR*, *TSR*, and *TCIW* in orange have positive impacts, i.e., increasing the prediction, while *STRD*, *U*, *TEMP*, and *V* in blue have a negative impact, i.e., decreasing the prediction, for this instance.

B. APPLYING SHAP XAI TOOL TO SOLAR PV FORECASTING

SHAP computes the global feature importance by taking an average of the magnitude of the SHAP values across the

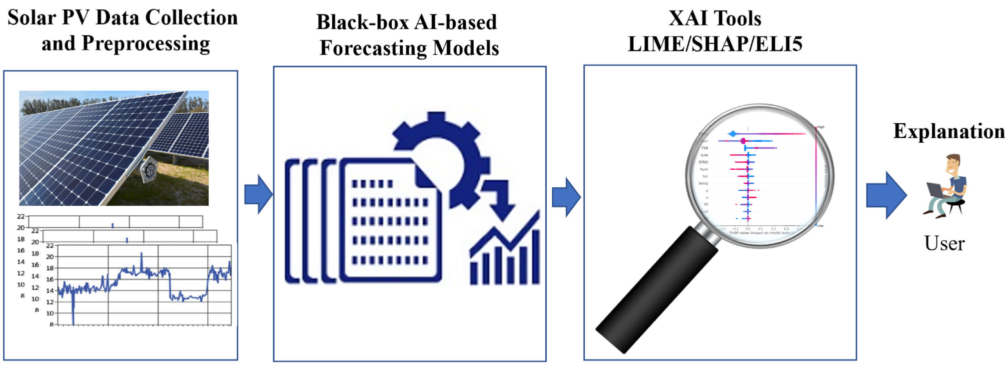


FIGURE 2. Flowchart showing the methodology applied in the work.

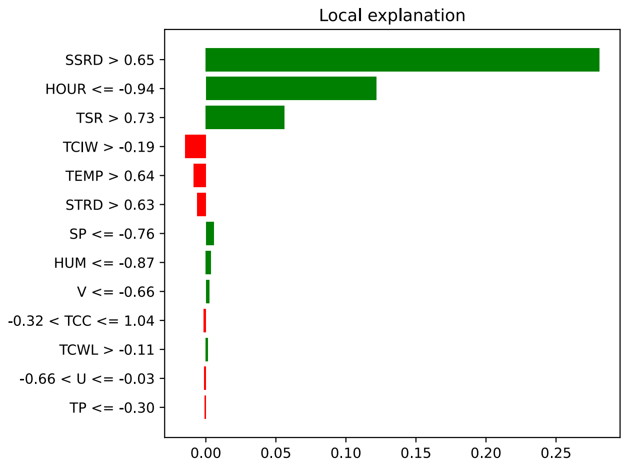


FIGURE 3. LIME feature importance results.

dataset. In this study, each SHAP value provides information about the contribution of each feature, either positively or negatively, towards predicting solar PV power. Figure 7 shows the SHAP values for the Random Forest regression model. It represents each feature's importance while remaining visually concise. The higher the SHAP value of a feature the higher is the impact on the model output, either negatively or positively. As per the SHAP values, *SSRD* has the highest impact on the model output, around 0.5. Each forecast is run through the model, and a dot is created for each feature attribute value. Therefore, one result gets one dot on each feature's line. This reveals, for example, that a rise in the *SSRD* increases the solar PV output. Dots are colored by the feature's value for that forecast and pile up vertically to show density.

The other advantage of the SHAP XAI tool is to provide a partial dependence plot. It helps to understand how the marginal effect of one or two features have on the predicted outcome of the model. We can plot the SHAP value of the target feature to explore the relationship with other features, i.e., linear, monotonic, or more complex. For instance, vertical dispersion at a single value of *SSRD* represents interaction effects with *HOUR*, as shown in Figure 8. The plot shows that there is a positive and approximately linear correlation relationship between target variable, i.e., *SSRD*, and *HOUR* interacting frequently.

SHAP is also capable of local interpretability of the models. The results in Figures 9-11 are generated by applying the SHAP algorithm on the same 3 hours of a day as done in the previous section with LIME. We predicted three instances as 0.720, 0.680, and 0.690, while the actual values are 0.774, 0.758, and 0.712, respectively. According to three predictions, all features show similar trends for all selected data points. Feature values causing increased predictions are in red, and their visual size shows the magnitude of the feature's effect. Feature values decreasing the prediction are in blue. The biggest impact comes from *SSRD*, *HOUR*, and *TSR* for the prediction. For example in Figure 10, we predicted the solar PV power generation as 0.680, here *SSRD*, *HOUR*, and *TSR* in red have positive impacts, i.e., increasing the prediction, while *STRD* in blue has a negative impact, i.e., decreasing the prediction, for this instance.

C. APPLYING ELI5 XAI TOOL TO SOLAR PV FORECASTING

ELI5 is a Python library that allows to visualize and debug various ML models, such as Scikit-learn, XGBoost, LightGBM, CatBoost, Sklearn-crfsuite, and Keras. ELI5 uses an approach based on interpreting-random-forest feature weights. These weights are calculated by following the decision paths in trees of an ensemble. Each node of the tree has an output score, and the contribution of a feature on the decision path is how much the score changes from parent to child. Table 1 shows the weight for each feature in the model. *SSRD* has the highest weight, i.e., it is the most important feature. The prediction can be described as the sum of the feature contributions + the "bias", i.e., the mean given by the top most region that covers the entire training set. According to the results, *SSRD* has the highest contribution to final solar PV predictions. *HOUR* also has a significant contribution along with *TSR* and *TCIW*. The intercept (often labeled the constant) is the expected mean value of the predicted results when all features are zero.

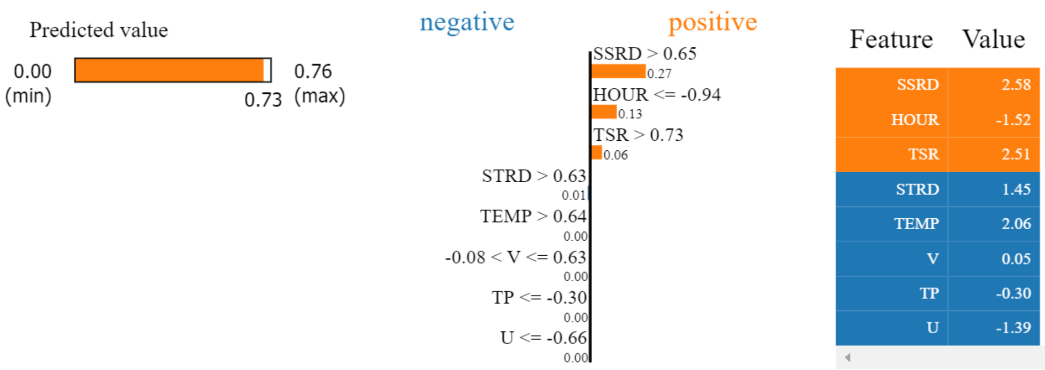


FIGURE 4. LIME explanation for hour 6 of a day where Actual:0.77461539, Predicted:0.73156154.

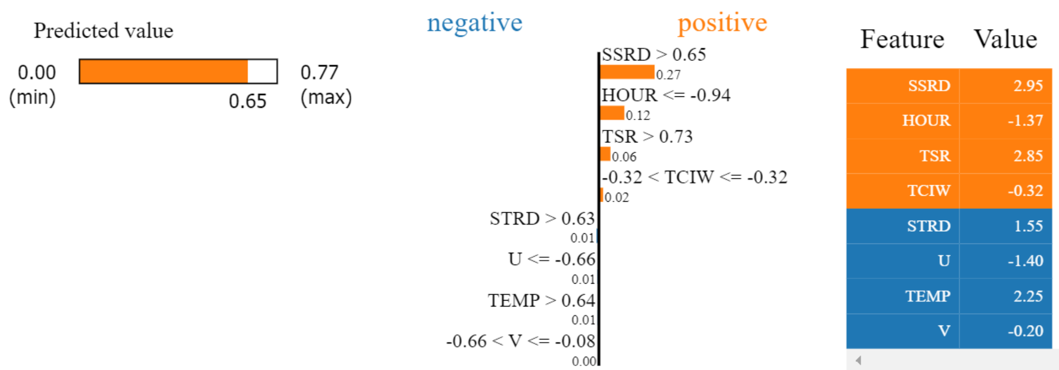


FIGURE 5. LIME explanation for hour 7 of a day where Actual:0.75833333, Predicted:0.64756026.

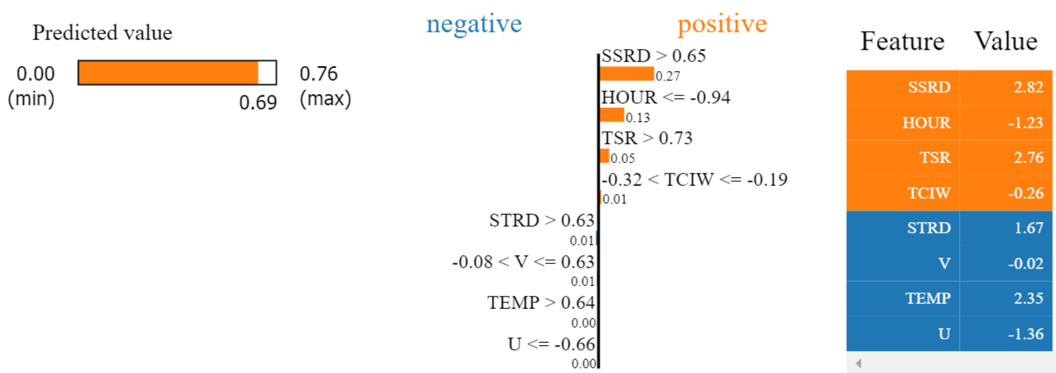


FIGURE 6. LIME explanation for hour 8 of a day where Actual:0.71262821, Predicted:0.6868576.

To understand how the model works, individual predictions are examined in ELI5. We examine this for three instances of each model, i.e., three consecutive hours of a day shown in Tables 2-4. In terms of accuracy, the solar PV predicted power generation values are 0.717, 0.677, and 0.689, while the actual values are 0.774, 0.758, and 0.712, respectively. Tables show that *SSRD* has the highest positive effect, while *STRD* has the highest negative effect. For example, as given in Table 3, contributions of *SSRD*, and *STRD* to the predicted value, i.e., 0.677, are 0.474, and −0.051, respectively.

D. MODELS WITH SUBSET OF FEATURES USING THE ANALYSIS FROM ELI5, LIME AND SHAP

XAI tools help us understand complex black-box machine learning models. With the knowledge that we get about the inner working of the ML models, our goal is to improve the models performance by providing it with better inputs and take away the input features that negatively impact the model's performance. In this section, we look at the feature importance provided by LIME, SHAP and ELI5, and eliminate the two least important features according to each of the

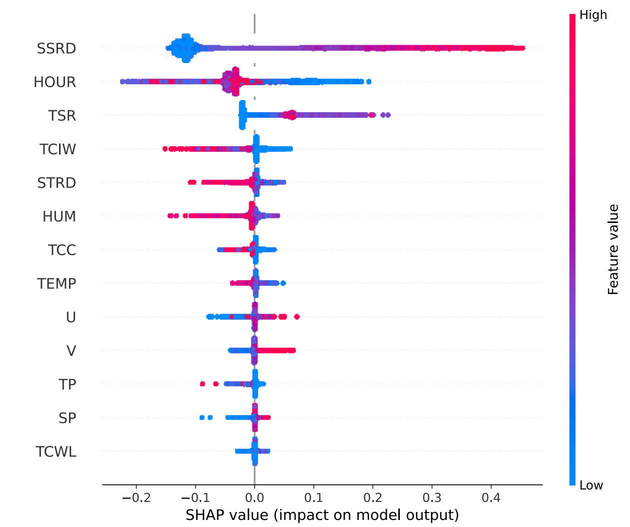


FIGURE 7. SHAP values with impact on model.

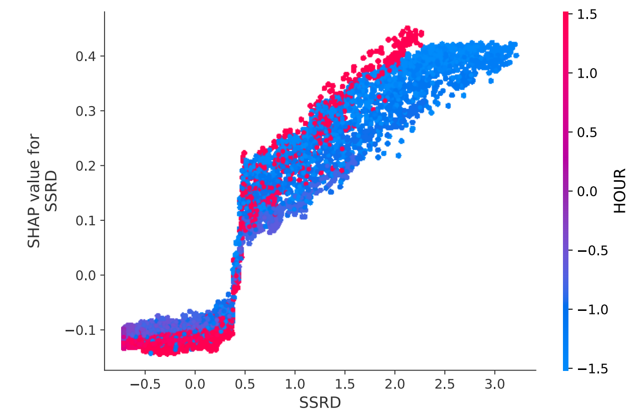


FIGURE 8. Dependence plot of SSRD.

TABLE 1. ELI5 explanations.

Feature	Weight	Std
SSRD	0.778945	0.102629
HOUR	0.075221	0.018073
TSR	0.053688	0.103161
TCIW	0.015678	0.003519
STRD	0.013434	0.003318
TCC	0.013019	0.006075
HUM	0.011854	0.002545
U	0.008606	0.002205
TEMP	0.006973	0.002058
V	0.006887	0.00189
SP	0.006249	0.001437
TCWL	0.005549	0.001944
TP	0.003896	0.001722

XAI tool. Table 5 shows RMSE results associated with all features and removing the two least important features as per the three XAI models. SHAP XAI model provides a better performance in terms of RMSE, i.e., from 7.236 to 7.216.

E. OBSERVATIONS

This study gives a few pointers on how to implement an XAI tool, i.e., LIME, SHAP, and ELI5. Each XAI tool/package has its own strengths and limitations, in terms of computing cost, explanation locally/globally, feature weight, etc.

TABLE 2. ELI5 explanation for the 6th hour of a day.

Actual	0.774615	Predicted	0.717422
Target	**Feature**	**Weight**	**Value**
PV Power	SSRD	0.468908	2.575425
PV Power	<BIAS>	0.178025	1
PV Power	HOUR	0.085265	-1.51686
PV Power	TCC	0.029704	-1.10524
PV Power	TCIW	0.012997	-0.32074
PV Power	HUM	0.007081	-1.61374
PV Power	TSR	0.004176	2.508821
PV Power	SP	0.00322	-1.08073
PV Power	TCWL	0.001611	-0.34857
PV Power	TP	0.000852	-0.29901
PV Power	V	-0.00583	0.050971
PV Power	TEMP	-0.01019	2.061648
PV Power	U	-0.01932	-1.39247
PV Power	STRD	-0.03908	1.445211

TABLE 3. ELI5 explanation for the 7th hour of a day.

Actual	0.758333	Predicted	0.677499
Target	**Feature**	**Weight**	**Value**
PV Power	SSRD	0.474194	2.953836
PV Power	<BIAS>	0.178025	1
PV Power	HOUR	0.083735	-1.3724
PV Power	TCC	0.01109	-1.09629
PV Power	HUM	0.006946	-1.69948
PV Power	TSR	0.001394	2.851593
PV Power	TCWL	0.001183	-0.34819
PV Power	V	0.001053	-0.19898
PV Power	TP	0.000796	-0.29901
PV Power	SP	-0.00049	-1.16186
PV Power	TCIW	-0.0049	-0.31923
PV Power	U	-0.00848	-1.40341
PV Power	TEMP	-0.01566	2.251754
PV Power	STRD	-0.05138	1.551317

TABLE 4. ELI5 explanation for the 8th hour of a day.

Actual	0.71262821	Predicted	0.6890705127
Target	**Feature**	**Weight**	**Value**
PV Power	SSRD	0.475717	2.823923
PV Power	<BIAS>	0.178025	1
PV Power	HOUR	0.048827	-1.22794
PV Power	TCC	0.037107	-0.82766
PV Power	HUM	0.005721	-1.70496
PV Power	TSR	0.002227	2.759754
PV Power	TP	0.001239	-0.29901
PV Power	TCIW	-0.00034	-0.26052
PV Power	TEMP	-0.00165	2.346532
PV Power	V	-0.00314	-0.02194
PV Power	SP	-0.00469	-1.26207
PV Power	U	-0.00492	-1.3612
PV Power	TCWL	-0.00921	-0.30173
PV Power	STRD	-0.03585	1.673153

TABLE 5. RMSE results from each model.

Model	Features removed	RMSE (%)
w/ all features	-	7.236
LIME	TCWL, TCC	7.228
SHAP	SP, TCWL	7.216
ELI5	TP, TCWL	7.235

Table 6 shows the execution time for LIME, SHAP and ELI5. The LIME XAI model provides the best efficiency in terms of execution time, i.e., 34.3 milliseconds. The following

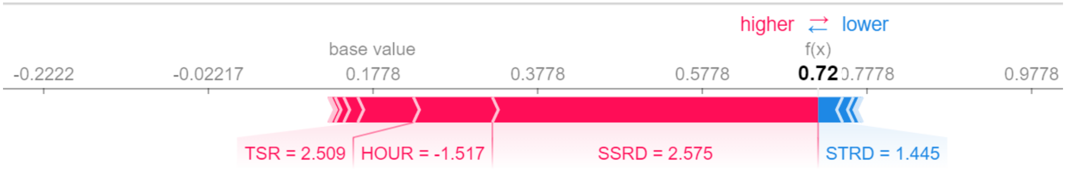


FIGURE 9. SHAP explanation for the 6th hour of a day.

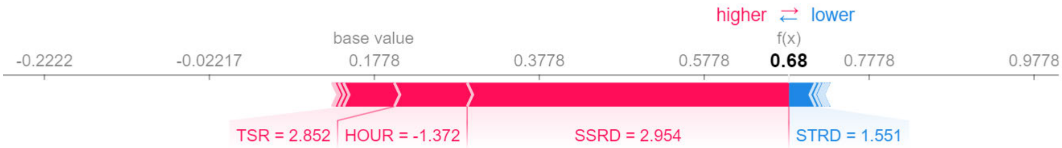


FIGURE 10. SHAP explanation for the 7th hour of a day.

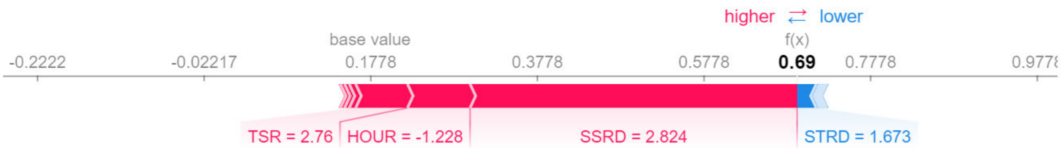


FIGURE 11. SHAP explanation for the 8th hour of a day.

TABLE 6. Time taken to run LIME, SHAP and ELI5.

Model	Time
LIME	34.3 milliseconds
SHAP	9.4 minutes
ELI5	47.32 milliseconds

observations regarding LIME, SHAP, and ELI5 tools can be made from results.

- Observation 1: The key limitation of all tools is that they need to run many evaluations of the original model.
- Observation 2: All tools support regression and classification models (in this study, we focus on regression models).
- Observation 3: LIME is a locally surrogated model, which explains the prediction at local boundaries.
- Observation 4: LIME is model agnostic, which means that it can be applied to any machine learning model.
- Observation 5: LIME does not guarantee a perfect distribution of the effects.
- Observation 6: The SHAP value is the only method to deliver a full explanation and considers all possible predictions, for instance, using all possible combinations of inputs.
- Observation 7: LIME is faster than SHAP since the calculation of SHAP values is very time-consuming as it checks all the possible combinations.
- Observation 8: The SHAP value is the contribution of a variable to the difference between the actual prediction and the mean prediction.
- Observation 9: SHAP can guarantee properties like consistency and local accuracy.
- Observation 10: SHAP provides more detailed information and results, such as visualizing, explaining multiple predictions, dependence and summary plots, and feature importance with SHAP values.
- Observation 11: ELI5 is the simplest of the three XAI tools.
- Observation 12: ELI5 does not support true model-agnostic interpretations and support for models.
- Observation 13: ELI5 is mostly limited to tree-based and other parametric-linear models.
- Observation 14: A prediction by ELI5 can be described simply, i.e., the sum of the feature contributions + the "bias".
- Observation 15: ELI5 provides weights for each feature depicting how influential it might have been in contributing to the final prediction decision across all trees as well as individual data-point predictions.

F. TEST ENVIRONMENT

The experiments presented in this study were implemented on Python version 3.8. The workstation used for the work runs an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz processor with 16 GB RAM, and NVIDIA GeForce GTX 1070 GPU with 8 GB memory.

V. CONCLUSION

This article presents the application of a random forest forecasting model to predict solar PV power generation. Furthermore, XAI tools, such as LIME, SHAP and ELI5, are applied to the random forest AI model to understand and explain the reasons for a particular prediction as well as to contribute to the adoption of XAI tools to smart grid applications. The data used for this article is a public dataset from GEFCOM 2014. According to the results, XAI tools can provide detailed information to interpret the model features and results as well as improvement of the model's results through explainability and transparency. This study has given a few pointers on how to choose an XAI tool, such as LIME, SHAP and ELI5. Each XAI tool/package has its own strengths and limitations, in terms of the computing cost, explanation locally/globally, feature weights, etc.

The utilities are willing to create next generation control centers with visualization technologies and business analytics tolls, which support emerging technologies, such as AI and mixed reality, but at the same time they want to simplify the usability for employees who have less expertise in these technologies. XAI based PV solar forecasting systems and similar tools provide a very productive play-ground for the utilities. It is expected that this study can benefit utility engineers and researchers working on power generation and load forecasting by providing an insight into the XAI potentials and availability in smart grid applications.

REFERENCES

[1] M.-J. Santofimia-Romero, X. del Toro-García, and J.-C. López-López, "Artificial intelligence techniques for smart grid applications," in *Green ICT: Trends Challenges*. 2011, pp. 41–44.

[2] J. A. Momoh, "Smart grid design for efficient and flexible power networks operation and control," in *Proc. IEEE/PES Power Syst. Conf. Expo.*, Mar. 2009, pp. 1–8.

[3] H. Zhou, M. Rao, and K. T. Chuang, "Artificial intelligence approach to energy management and control in the HVAC process: An evaluation, development and discussion," *Develop. Chem. Eng. Mineral Process.*, vol. 1, no. 1, pp. 42–51, May 2008.

[4] A. K. Pathak, D. S. Chatterji, and M. S. Narkhede, "Artificial intelligence based optimization algorithm for demand response management of residential load in smart grid," *Int. J. Eng. Innov. Technol.*, vol. 2, no. 4, pp. 136–141, 2012.

[5] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, and A. Vasilakos, "Anomaly detection and predictive maintenance for photovoltaic systems," *Neurocomputing*, vol. 310, pp. 59–68, Oct. 2018.

[6] A. Mellit and S. A. Kalogirou, "Artificial intelligence techniques for photovoltaic applications: A review," *Prog. Energy Combustion Sci.*, vol. 34, no. 5, pp. 574–632, Oct. 2008.

[7] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[8] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[9] D. Gunning, "Explainable artificial intelligence (XAI)," *Defense Adv. Res. Projects Agency (DARPA), nd Web*, vol. 2, p. 2, Nov. 2017.

[10] C.-J. Huang and P.-H. Kuo, "Multiple-input deep convolutional neural network model for short-term photovoltaic power forecasting," *IEEE Access*, vol. 7, pp. 74822–74834, 2019.

[11] *U.S. Solar Market and 15 States See Best Quarter Ever for Residential Solar*. [Online]. Available: https://www.seia.org/news/us-solar-market-and-15-states-see-best-quarter-ever-residential-solar

[12] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, "Data-driven probabilistic net load forecasting with high penetration of Behind-the-Meter PV," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3255–3264, May 2018.

[13] N. Prentzas, A. Nicolaides, E. Kyriacou, A. Kakas, and C. Pattichis, "Integrating machine learning with symbolic reasoning to build an explainable AI model for stroke prediction," in *Proc. IEEE 19th Int. Conf. Bioinf. Bioeng. (BIBE)*, Oct. 2019, pp. 817–821.

[14] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable AI in intrusion detection systems," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 3237–3243.

[15] Mandeep, H. S. Pannu, and A. Malhi, "Deep learning-based explainable target classification for synthetic aperture radar images," in *Proc. 13th Int. Conf. Human Syst. Interact. (HSI)*, Jun. 2020, pp. 34–39.

[16] J. Adams and H. Hagras, "A Type-2 fuzzy logic approach to explainable AI for regulatory compliance, fair customer outcomes and market stability in the global financial sector," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–8.

[17] A. Pierrot and Y. Goude, "Short-term electricity load forecasting with generalized additive models," in *Proc. ISAP Power*, 2011, pp. 1–6.

[18] H. Kumar, P. Mary Mammen, and K. Ramamritham, "Explainable AI: Deep reinforcement learning agents for residential demand side cost savings in smart grids," 2019, *arXiv:1910.08719*. [Online]. Available: http://arxiv.org/abs/1910.08719

[19] Y.-G. Lee, J.-Y. Oh, and G. Kim, "Interretation of load forecasting using explainable artificial intelligence techniques," *Trans. Korean Inst. Elect. Eng.*, vol. 69, no. 3, pp. 480–485, 2020.

[20] *XGBoost Documentationq*. [Online]. Available: https://xgboost.readthedocs.io/en/latest/

[21] *Explainersq*. [Online]. Available: https://shap.readthedocs.io/en/latest/

[22] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[24] J. Moon, K.-H. Kim, Y. Kim, and E. Hwang, "A short-term electric load forecasting scheme using 2-Stage predictive analytics," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 219–226.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[26] K. Sriwong, T. School of Computer EngineeringSuranaree University of Technology (SUT)111 University AvenueMuangNakhon Ratchasima 30000, K. Kerdprasop, and N. Kerdprasop, "Post-operative life expectancy of lung cancer patients predicted by Bayesian network model," *Int. J. Mach. Learn. Comput.*, vol. 8, no. 3, pp. 280–285, Jun. 2018.

[27] TeamHG-Memex. *Teamhg-Memex/eli5*. [Online]. Available: https://github.com/TeamHG-Memex/eli5

[28] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack," *J. Open Source Softw.*, vol. 3, no. 24, p. 638, 2018.

[29] D. Dataman. (Mar. 2020). *Explain Your Model With Microsoft's Interpretml*. [Online]. Available: https://medium.com/analytics-vidhya/explain-your-model-with-microsofts-interpretml-5daab1d693b4

[30] F. Revert. (Feb. 2019). *Interpreting Random Forest and Other Black Box Models Like XGboost*. [Online]. Available: https://towardsdatascience.com/interpreting-random-forest-and-other-black-box-models-like-xgboost-80f9cc4a3c38

[31] *Getting Startedq*. [Online]. Available: https://docs.seldon.io/projects/alibi/en/latest/overview/getting_started.html

[32] Sameer Singh Marco Tulio Ribeiro. (Aug. 2016). *Local Interpretable Model-Agnostic Explanations (Lime): An Introduction*. [Online]. Available: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/

[33] T. Peltola, "Local interpretable model-agnostic explanations of Bayesian predictive models via kullback-leibler projections," 2018, *arXiv:1810.02678*. [Online]. Available: http://arxiv.org/abs/1810.02678

[34] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.

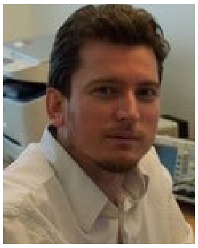
[35] C. Molnar, "Interpretable machine learning," in *A Guide for Making Black Box Models Explainable*. 2019. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[36] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," 2018, *arXiv:1802.03888*. [Online]. Available: http://arxiv.org/abs/1802.03888

[37] Posted By: OnClick360 and OnClick360. (Feb. 2020). *Interpretable Machine Learning With Lime Eli5 Shap Interpretml*. [Online]. Available: https://www.onclick360.com/interpretable-machine-learning-with-lime-eli5-shap-interpret-ml/

[38] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *Int. J. Forecasting*, vol. 32, no. 3, pp. 896–913, Jul. 2016.

[39] M. Abuella and B. Chowdhury, "Solar power probabilistic forecasting by using multiple linear regression analysis," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–5.



MURAT KUZLU (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electronics and telecommunications engineering, in 2001, 2004, and 2010, respectively. He joined the Department of Engineering Technology, Old Dominion University (ODU), as an Assistant Professor, in 2018. In 2006, he joined the TUBITAK MAM Energy Institute (Scientific and Technological Research Council of Turkey-Marmara Research Center), where he worked as a Senior Researcher. Before joining ODU, he worked as a Research Assistant Professor with the Virginia Tech's Advanced Research Institute. His research interests include smart grid, demand response, the Internet of Things (IoT), machine learning with a particular applications on energy savings, and wireless communication.

UMIT CALI (Member, IEEE) received the B.E. degree in electrical engineering from Yildiz Technical University, Istanbul, Turkey, in 2000, and the M.Sc. degree in electrical communication engineering and the Ph.D. degree in electrical engineering and computer science from the University of Kassel, Germany, in 2005 and 2010, respectively. He joined the Department of Electric Power Engineering, Norwegian University of Science and Technology, Norway, in 2020, as an Associate Professor. He worked with the University of Wisconsin–Platteville and the University of North Carolina at Charlotte as an Assistant Professor, in 2013 and 2020, respectively. His current research interests include energy informatics, artificial intelligence, blockchain technology, renewable energy systems, and energy economics. He is serving as the Active Vice Chair for the IEEE Blockchain in Energy Standards WG (P2418.5).

VINAYAK SHARMA received the M.S. degree in applied energy and electromechanical systems with a thesis on deterministic and probabilistic forecasting for renewable energy. He is currently pursuing the Ph.D. degree in electrical engineering with the University of North Carolina at Charlotte. His research interests include applying statistical and machine learning techniques for load, wind, solar, and natural gas forecasting, power systems planning, and optimization of energy storage.

ÖZGÜR GÜLER received the B.S. degree in computer science and the M.S. degree in computer science with a focus on image-guided surgery from the University of Innsbruck, Innsbruck, Austria, and the Ph.D. degree from the Medical University of Innsbruck, Austria, with a focus on image-guided diagnosis and therapy. He is currently an Imaging Scientist and AI Researcher specialized in 3-D chronic wound imaging and computer vision. Prior to joining eKare Inc., he was a Researcher with the Sheikh Zayed Institute (SZI) for Pediatric Surgical Innovation Center, Washington, DC, where he developed the segmentation and classification algorithms that laid the groundwork of the eKare inSight system.