energies

MDPI

Review

Review on Interpretable Machine Learning in Smart Grid

Chongchong Xu [1], Zhicheng Liao [1], Chaojie Li [2,*], Xiaojun Zhou [1] and Renyou Xie [2]

[1] School of Automation, Central South University, Changsha 410083, China; chongxu@csu.edu.cn (C.X.); liaozhch@163.com (Z.L.); michael.x.zhou@csu.edu.cn (X.Z.)

[2] Department of Electrical Engineering and Telecommunications, University of New South Wales, Kensington, NSW 2052, Australia; xierenyou21@163.com

[*] Correspondence: chaojie.li@unsw.edu.au

Abstract: In recent years, machine learning, especially deep learning, has developed rapidly and has shown remarkable performance in many tasks of the smart grid field. The representation ability of machine learning algorithms is greatly improved, but with the increase of model complexity, the interpretability of machine learning algorithms is worse. The smart grid is a critical infrastructure area, so machine learning models involving it must be interpretable in order to increase user trust and improve system reliability. Unfortunately, the black-box nature of most machine learning models remains unresolved, and many decisions of intelligent systems still lack explanation. In this paper, we elaborate on the definition, motivations, properties, and classification of interpretability. In addition, we review the relevant literature addressing interpretability for smart grid applications. Finally, we discuss the future research directions of interpretable machine learning in the smart grid.

Keywords: interpretable machine learning; explainable artificial intelligence; machine learning; deep learning; smart grid

check for updates

Citation: Xu, C.; Liao Z.; Li, C.; Zhou, X.; Xie, R. Review on Interpretable Machine Learning in Smart Grid. *Energies* **2022**, *15*, 4427. https://doi.org/10.3390/en15124427

Academic Editor: Marco Pau

Received: 1 May 2022
Accepted: 15 June 2022
Published: 17 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The smart grid greatly improves the traditional power grid with advanced measurement and sensing, information and communication technologies, simulation analysis and control decision-making systems [1–4]. Compared with the traditional power grid, the smart grid has more advantages in self-healing, renewable energy consumption, situational awareness, information interaction and stability [5,6]. With the access to intermittent and distributed generation and the development of electricity markets, the complexity and uncertainty of the operation of the power system have greatly increased. The smart grid is gradually becoming a power cyber-physical system that closely integrates measurement, communication and various external systems (such as weather, market, etc.) [7,8]. It continues to generate high-dimensional, multi-source heterogeneous data. The emergence of massive data can provide data support for the study of smart grid problems, but also bring a new challenge to smart grid management. How to efficiently and pertinently analyze massive data with complex sources and extract valuable information from it to assist smart grid decision-making has become an important topic [9].

Artificial intelligence (AI) technology, which can improve the efficiency and accuracy of decision-making, is an important means to support the smart grid [10]. Machine learning (ML) is a branch of AI, the key technology and core creativity of AI development, and plays a major role in promoting the development of AI technology. The application of ML technology in the smart grid is regarded as one of the important technologies in the development of the power industry. ML algorithms use few assumptions and a lot of computing power to mine complex relationships of history data [11]. The use of ML algorithms can form input–output relationship mapping for complex mechanisms in the smart grid, thereby breaking through the limitations of existing physical knowledge, so it is very suitable for dealing with the challenges of the smart grid. Commonly used ML algorithms include linear regression (LR) [12], support vector machine (SVM) [13], K-nearest neighbors

(KNN) [14], clustering algorithms [15], decision tree (DT) [16], ensemble learning [17], multi-layer perceptron (MLP) [18], etc. They are currently used to address many related issues in the smart grid, such as rapid diagnosis of fault information [19], accurate prediction of distributed energy resources [20], and stability analysis of complex power grids [21]. In recent years, with the development of computing power, deep learning (DL) which is a special kind of ML is emerging. DL is a neural network with multiple hidden layers. Its basic idea is to combine low-level features through multiple layers of network structures and nonlinear transformations to form abstract high-level representations to discover complex patterns in data [22]. In recent years, in order to improve the effect of the deep neural network (DNN) and adapt to different forms of data and problems, some unique DL algorithms have been proposed successively, such as stacked autoencoder (SAE) [23], convolutional neural network (CNN) [24], recurrent Neural Network (RNN) [25], etc. DL algorithms are more complex in structure, which is considered more suitable for processing the massive and complex data in the smart grid. DL can also provide better accuracy than other ML algorithms [20].

Although the potential of ML for smart grid applications has been recognized, obstacles to further deployment of ML models remain. An important factor is the black box nature of most ML algorithms. With the development of ML algorithms, especially the emergence of DL, its representation ability is gradually improving. However, with the increase in model complexity, the interpretability of ML algorithms has deteriorated. Previous works in the smart grid domain mainly pursued accurate ML models while ignoring interpretability [26]. The power sector is highly regulated, and the analysis as well as decision-making of the power system must be reliable and transparent [27]. ML techniques often make critical decisions, especially in use cases regarding power outages [28], and professionals are hesitant to deploy such models because a model error may induce a very large impact. Furthermore, electricity is intrinsically complex and dangerous. If there is a problem with the model, the ML algorithm is likely to affect the personal safety of field employees [29]. Therefore, for some smart grid control problems that are too risky, ML is still not trusted at present. Furthermore, power grid professionals are more interested in understanding how decision outcomes are actually produced. However, most ML models are so complex that it is impossible for anyone to understand the reasoning process that makes the predictions. The input may undergo a series of complex nonlinear transformations, interact with numerous neurons, and then produce predictions. Such a black-box model cannot help us gain insight beyond the predicted outcome, understand the key drivers of the model and the role of its different input features [30]. Therefore, interpretability is an inherent requirement for applications in the power domain. In order to make better use of ML algorithms to promote the development of the smart grid, it is urgent to develop interpretable ML.

Interpretable ML is a hot topic in AI, and it allows professionals to understand, audit and even improve ML systems. ML algorithms are traditionally considered to have a tradeoff between performance and interpretability. ML models such as linear models and DTs are interpretable, but their fitting ability and prediction performance are often poor [31]. These models tie interpretability to model complexity, especially sparsity. Sparsity is considered an important aspect of interpretability. For the same model, the sparser the parameters or structure, the more interpretable the model is. Recently, researchers have proposed interpretation methods for complex ML algorithms (especially DL) that aim to enhance model interpretability without sacrificing model complexity, such as local interpretable model-agnostic explanation (LIME) [32] and shapley additive explanations (SHAP) [33]. After the model is built and trained, these methods use the approximate model, feature contribution, sensitivity, or other statistics to explain the black-box prediction process. At present, interpretable ML has been applied in various fields, such as healthcare, autonomous driving, finance and other fields, some studies can refer to [34–37].

Although interpretability has been noted in the smart grid and some related work has emerged, to the best of the authors' knowledge, there is no related review of interpretable ML in the smart grid. In general, the limited degree of human understanding of the interpretability of ML limits the upper bounds of ML applications in the smart grid. Therefore, we believe that a review of interpretable ML research in the smart grid is warranted. In this paper, we reveal the development of interpretable ML by reviewing the descriptions of interpretability and the classification of interpretable ML in recent papers. Further, we review existing interpretable ML methods in the smart grid, explore new possibilities for interpretable ML in the smart grid, and propose future research directions.

The rest of the paper is organized as follows. In Section 2, definitions, motivations, and properties of interpretability are given. Section 3 explains the classification of interpretable ML. Next, Section 4 discusses the application of explainable ML in the smart grid. Some future research directions are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Description of Interpretable Machine Learning

2.1. *Definition*

At present, the interpretability of ML has not been clearly defined, and there are subjective differences in the understanding of interpretability by different researchers [38–41]. One of the broadest definitions is given by [38]: ability to explain or to present in understandable terms to a human. In a nutshell, interpretability means providing simple and clear terms to explain the decision-making mechanism of a model and enabling users to understand and trust the decision.

Furthermore, there are two terminologies that are often confused: interpretability and explainability. Their concepts are difficult to define strictly. Some researchers have pointed out that interpretability mainly refers to providing human beings with an understandable model operating mechanism [42]. In fact, Interpretable ML has a long history, dating back to the 1950s. It first appeared in expert systems based on context rules and logical models (DTs, Decision rules) [43–45]. Early Interpretable ML pursued intrinsic explanation, i.e., explanation is part of the model. The original model structure and decision progress are understandable to people. For example, a DT model is intrinsically interpretable, because it can provide a human-friendly explanation: *IF input x is smaller/large than threshold c AND . . . THEN the prediction is the average of the instances in leaf node l*. It is important to note that interpretability is subjective, as it requires statistical or domain knowledge to reasonably explain the model decision [46].

Explainability refers to giving abstract-level insights into how models work and make decisions without trying to reveal computational details [47]. Its main purpose is to introduce explanations for complex black-box models that are not interpretable so that they can be understood by humans. Explainability stems from function approximation in the 1990s, where a simple model approximates the model output to explain a black-box decision process [48], and it is widely researched after DARPA proposed explainable artificial intelligence (XAI) in 2016. In practice, explainability methods usually generate some key elements (such as statistics, visualizations, or a simple interpretable model) to construct approximate explanations.

Actually, interpretability is a broader concept than explainability. As described in [40]: systems are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation. This means that interpretability includes introspection (intrinsically interpretable) and explainability (producing explanations for black-box models). Therefore, we believe that interpretable ML has broadened the concept. It pursues not only intrinsic interpretability, but also interpreting/explaining black-box models and other techniques that make models more transparent.

2.2. *Motivations*

The motivation for interpretability has stimulated a great deal of discussion. A lot of papers have discussed the importance of interpretability and highlighted the potentially catastrophic consequences of a lack of interpretability [38,49,50]. Ref. [49] summarizes three key reasons: to audit, to validate, and to discover. We find that all three are relevant to the smart grid. Figure 1 shows the benefits of Interpretable ML for smart grid applications.

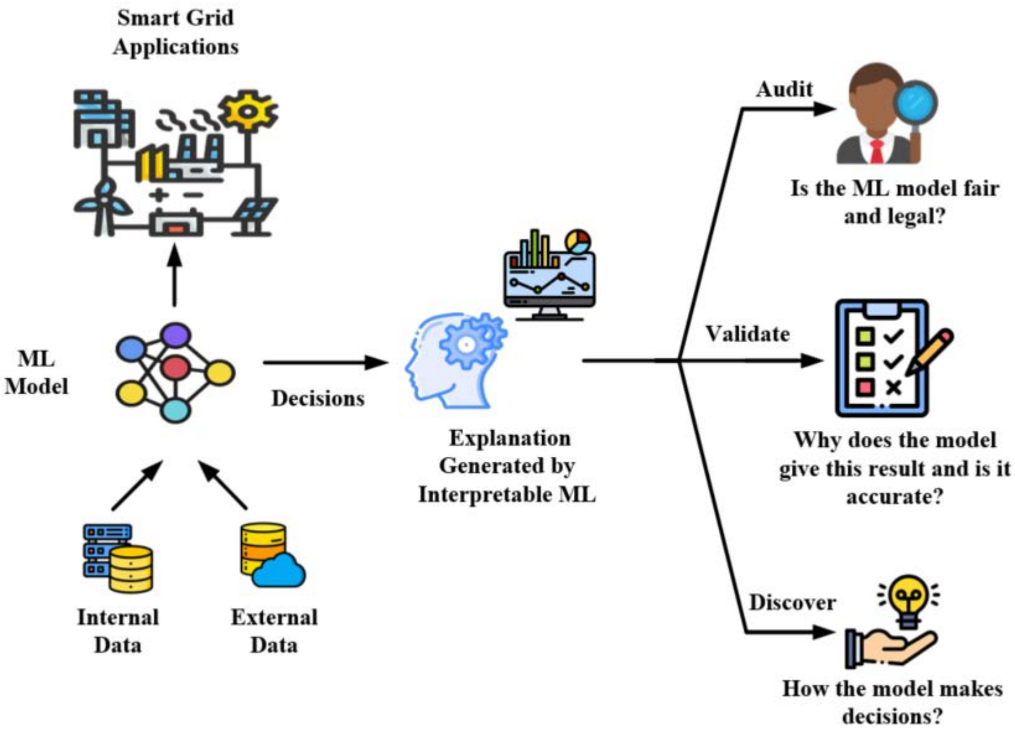
Smart Grid Applications
ML Model
Decisions
Internal Data
External Data
Explanation Generated by Interpretable ML
Audit
Is the ML model fair and legal?
Validate
Why does the model give this result and is it accurate?
Discover
How the model makes decisions?

Figure 1. An illustration of interpretable ML in the smart grid.

2.2.1. To Audit

The ML technique has become an important part of many human-oriented applications, such as credit risk assessment [51], medical screening [52], etc., and has produced a great social impact. From this perspective, the fairness and ethics of the ML model are a bigger issue. There may be prior biases hidden in the data, such as racial discrimination, geographical discrimination, etc. [53]. A trained ML model may inherit biases in the training data and automate injustices. Interpretable ML methods can help quantify and reduce this ethical bias by introducing explanations. Notably, the need for interpretability has been written into laws and regulations. The European Union's General Data Protection Regulation (GDPR), which came into effect in May 2018, clearly stipulates that when a machine makes a decision about an individual, the decision must meet certain requirements for interpretability.

In the smart grid, we need to focus on this when it comes to applications that are more focused on the individual [29]. Load modeling or customer behavior modeling may draw conclusions from the electricity consumption profile of a household or a region. Unfair decisions can be made when deciding where to upgrade the grid and selecting potential customers. In addition, smart meter electricity theft detection systems may predict theft based on factors such as location. This could negatively impact those accused customers. Therefore, it is difficult to ensure fairness and morality if the reliance of the model on sensitive features is not transparent enough.

2.2.2. To Validate

When taken from an epistemological point of view, interpretability can help to verify the safety and reliability of algorithms, thus allowing ML systems to gain trust. The performance and interpretability of ML models in practical critical systems require rigorous and continuous verification of their safe use [54]. Verifying the behavior of ML systems is both important and difficult. The black-box nature of most ML models makes it nearly impossible to verify that they work as expected. A common cause of unreliable systems is overfitting. Since the model overreacted to tiny noises, it performed well on the training data, but failed to predict in practice [55]. Interpretation is an effective means of verifying network overfitting. Through model interpretation, researchers can gain some information about whether overfitting has occurred. This is mainly because overfitted models usually focus on non-informative features in the raw data, which are impossible to understand in most cases.

In the smart grid, rarely completely testable ML systems pose a challenge for some applications involving large risks. For example, the accurate and rapid voltage stability assessment of the power system is of great significance to maintain the voltage stability [56]. Without timely and reliable voltage stability assessment, voltage instability may occur after a power system is disturbed. In severe cases, it will lead to voltage collapse or even power outages, causing huge economic losses in multiple industries. Advanced ML techniques can assess the stability of power system voltages so that grid operators can take preventive measures in advance. Although the rich high-resolution system state data provided by the wide-area measurement system of the smart grid creates favorable conditions for this task. However, how to ensure that ML algorithms can extract the correct valuable information is still a huge challenge. Since traditional ML algorithms cannot prove the reliability of the assessment results, operators of smart grids may be reluctant to act in advance to correct voltage instability. Interpretable ML can provide the decision-making process or the contribution of the input variables. These interpretations can be compared with the actual operating laws of the power system to help decision makers verify the reliability of the prediction results. For example, if there is an interpretation that a too high or too low node voltage will have a negative impact on the stability prediction, it is in line with the actual operation law of the power system. The interpretation of the model allows grid operators to trust prediction results and take quick steps to maintain voltage stability. Figure 2 shows the general idea of interpretable ML for voltage stability assessment.

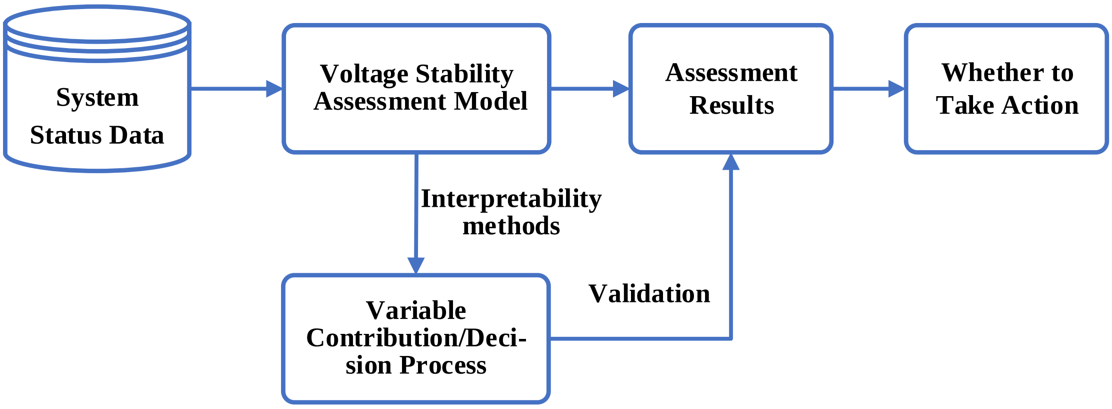

Figure 2. The general idea of interpretable ML for voltage stability assessment.

2.2.3. Discovery

An interpretable ML model can help us understand the reasons behind the output and discover correlations between various factors. This is important because it can provide meaningful knowledge and even facilitate the formation of new theories.

For smart grid applications, the knowledge generated by model interpretation can help grid operators solve unexpected problems, thereby guaranteeing system reliability. For example, grid fault diagnosis is an important application to realize the self-healing function of the smart grid [57]. When the power grid fails, the power grid fault diagnosis system needs to quickly analyze the fault-related data from the massive measurement data, find the cause of the fault, and assist the dispatching operators to analyze and deal with the accident in a timely manner, and quickly restore the power supply. In order to identify and resolve failures as quickly as possible, we must identify the most likely failures and their causes for further investigation. We can use system measurements, as well as other external factors (such as weather), to train a fault diagnosis model and generate a fault list that enables operators to take appropriate action immediately. In this case, interpretable

ML can help explain the types of failures that may occur so that maintenance personnel can fix them as quickly as possible. Going a step further, interpretable ML can discern the causal logic between input and output to discover the cause of equipment fault, so that they can find weak points in the system and take action to prevent the failure from recurring. Figure 3 shows a flowchart of interpretable ML methods for smart grid fault diagnosis.

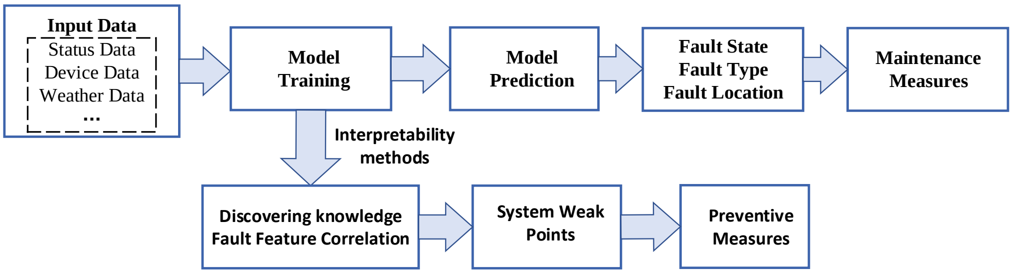

Figure 3. A flowchart of interpretable ML methods for smart grid fault diagnosis.

2.3. Properties

Below we give several properties of explanation methods, which are derived from the research of [58]. These properties can be used to judge the quality of interpretable methods or explanations, although it is still difficult to accurately quantify these properties.

- **Expressive Power**: The language or structure of explanation. Such as logic rules, linear models, statistics, natural language, etc.
- **Translucency**: Translucency describes how much the explanation method looks inside the ML model. For example, interpretable methods that rely on intrinsically interpretable models are highly translucent. Explanation methods that rely only on inputs and outputs and treat the model as a black box have zero translucency.
- **Portability**: Portability describes the range of ML models that can be interpreted using this method. Model-agnostic methods are more portable.
- **Algorithmic Complexity**: The computational complexity of the interpretable methods.

The **quality of explanation** is another important characteristic and usually has the following properties:

- **Accuracy**: The ability of an explanation for a decision to generalize to other unseen situations.
- **Fidelity**: The degree to which the explanation reflects the decision-making behavior of the model. Some explanations only provide local fidelity, such as LIME.
- Consistency: Consistency measures the degree to which models trained on the same task and producing similar predictions produce similar explanations.
- **Stability**: Stability is the similarity of explanations between similar instances. This criterion targets explanations generated from the same prediction model.
- **Comprehensibility**: The readability of the explanation (subjective) and the size of the explanation (such as the depth of the DT, the number of weights in the linear model, etc.).
- **Certainty**: Whether the explanation reflects the confidence of the predicted result.
- **Degree of Importance**: Does the explanation include the importance of its return component?
- **Novelty**: Does the explanation reflect that the instance to be explained comes from a new region far from the distribution of the training data? With high novelty, model decisions may be inaccurate.
- **Representativeness**: Representativeness is the extent to which the explanation can cover the instance. For example, the rule interpretation of a DT can cover the entire model, and the Shapely value only represents the interpretation of a single prediction.

3. Taxonomy of Interpretable Machine Learning

Interpretability methods can be classified according to different criteria [59]. It can be divided into *local* versus *global* according to whether it is for a specific sample/feature or the whole of the model. Another criterion is *model-specific* versus *model-agnostic*. Model-specific methods rely on the parameters or internal structure of the model. Model-agnostic methods only need to know the inputs and outputs of ML models, so it is suitable for interpreting any ML model. It can also be divided into *pre-model*, *in-model* and *post-model* according to the stage of explanation generation. In this paper, we adopt this classification to introduce different interpretability methods, as illustrated in Figure 4.

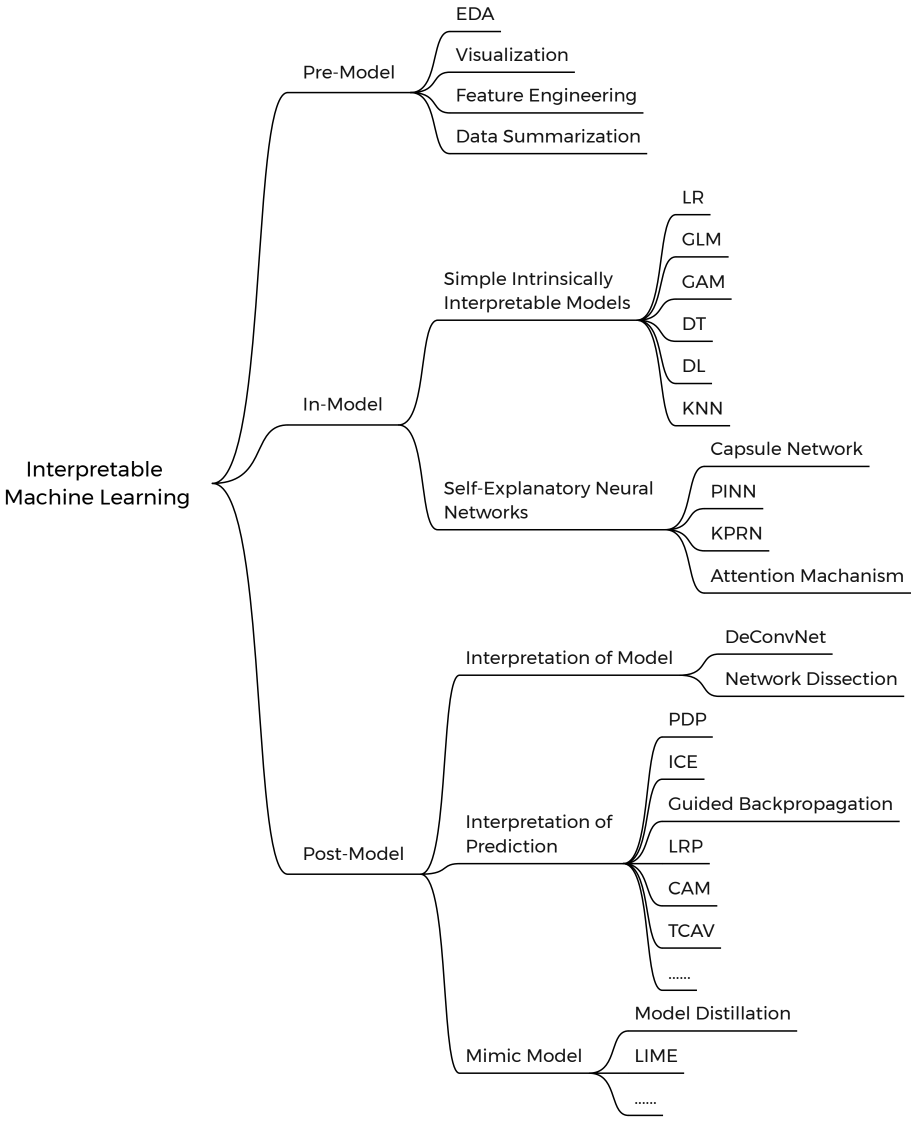

Figure 4. Taxonomy of interpretable ML.

3.1. Pre-Model

Pre-model interpretability is applied before ML model selection and training. Pre-model interpretability is related to data interpretability, whose goal is to understand the dataset used for ML models as much as possible. Pre-model interpretability is mainly achieved through exploratory data analysis (EDA) [59]. EDA is a collection of data analysis methods used to explore the structure and regularity of data [60]. EDA can help us better understand patterns in data, find outliers, and find correlations between features. The most basic EDA method is descriptive statistics, including calculating the mean, standard deviation, and quantiles. Other methods are visualization, feature engineering, data summarization, etc. [61].

Visualization is an important means of exploratory data analysis. Visual analysis transforms data into graphical representations, which can enhance human cognitive ability to data. Visualization is also widely used to improve data quality and assist data processing [62]. To support users in visually identifying patterns in high-dimensional data, dimensionality reduction methods are usually used to visualize high-dimensional datasets. Commonly used dimensionality reduction methods are principal component analysis (PCA) [63] and t-distributed stochastic neighbor embedding (t-SNE) [64].

Feature engineering can extract useful features and discover feature relationships. Representative sparse features help understand and interpret data. Feature extraction is a critical step in interpretable feature engineering, as the future implementation of ML algorithms heavily depends on the selected features [61]. Feature correlation analysis can be used to find implicit relationships between variables from large-scale data sets. It can also help us verify subjective judgments and improve data interpretability. The most commonly used feature correlation analysis methods are Pearson correlation [65] and Spearman's rank-order correlation [66].

The target of data summarization is outputting smaller subsets of samples that reflect the overall characteristics of the dataset [67]. Prototype selection is an implementation of data summarization [68]. Prototype selection usually selects the sample prototype that is most representative of the data according to the inherent distribution and structure of the target set. Classical prototype selection algorithms include K-Medoid [69] and Affinity Propagation Clustering [70], which select the prototype set that meets the requirements by minimizing the global dissimilarity between the target set and the prototype set. The prototype reflects the main distribution of the data set, but does not reflect all distributions of the data. Ref. [71] proposed model criticism, i.e., the data points with the largest similarity deviation between the dataset and the prototype. Criticism represents data points that are not well explained by the prototype and it gives new insights into the data.

3.2. In-Model

In-model interpretability aims to create ML models that are intrinsically interpretable [59]. The explanation is contained within the model and is part of the prediction process, allowing model decisions to be understood without additional post-processing. We generally create intrinsically interpretable ML models through mediations and constraints such as linearization, rules, examples, sparsity or causality.

3.2.1. Simple Intrinsically Interpretable Models

The easiest way to achieve in-model interpretability is to use simple ML models. These models are inherently transparent, decomposable and simulatable. Some classical linear models, such as LR, generalized linear model (GLM), generalized additive model (GAM) have simple structure and strong statistical basis. For LR, the model weights reflect the relationship between features, giving an easy-to-understand explanation. GLM is a generalization of LR model [72]. On the one hand, GLM does not require a linear relationship between features and prediction. On the other hand, GLM does not require the predictions to obey a normal distribution. GLM has the following form:

$$g(y) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i, \tag{1}$$

where x_i is the ith feature, g is a link function and β_i represents model weight. Logistic regression model is a GLM that assumes a Bernoulli distribution and uses the Logit function as the link function. GAM is a further extension of GLM that allows the use of arbitrary functions to model the effect of each feature on prediction [73]. The general form of GAM is:

$$g(y) = \beta_0 + \sum_{i=1}^{n} f_i(x_i), \tag{2}$$

where f_i represents a univariate function which is possibly nonlinear. GAM is more accurate because it captures the nonlinear relationship between each feature and the final prediction.

Further, pairwise interactions can be added to the GAM to form GA^{2}M [74], which has the following form:

$$g(y) = \beta_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{i \neq j} f_{i,j}(x_i, x_j), \tag{3}$$

The link function of a GAM can be a very complex nonlinear function, even a DT or a neural network.

The rule-based methods give the model decision-making process in symbolic form, which can describe and explain the model mechanism [75]. The most widely used rule model is DT. A DT consists of leaf nodes representing categories and internal nodes representing features or attributes. Every path from the root node to the leaf node in the DT can be transformed into a rule in the form of *if-then*, forming a traceable reasoning process [16]. Some other rule-based methods are decision list, decision set and fuzzy system etc. Decision list or decision set are assembled from a set of pre-mined rules by association classification methods [76]. Decision list greedily adds rules to the model one by one. Decision set scores each rule individually according to the scoring function and simply adds all the "highest scoring" rules to the model. In addition, the fuzzy rule-based system also provides interpretability and is able to effectively utilize quantitative information and qualitative knowledge to deal with uncertainty [77].

KNN is the most classic nearest neighbor-based model. KNN finds k training instances with the smallest distance from the test instance and uses their average as the prediction. Finding a suitable distance metric to quantify the difference between input instances is very important for KNN models [78]. It is important to note that the nearest neighbor models require a lot of distance computation. Moreover, the nearest neighbors may not be representative, leading to poor interpretability.

3.2.2. Self-Explanatory Neural Networks

In addition to the existing simple models, there are some ways to generate in-model interpretability by making deep models more transparent. These complex models often have meaningful features or structures within neural networks from which useful information can be extracted to explain prediction. There are roughly two types of self-explanatory neural networks. One is to the neural network imposes physical, semantic, or causal constraints to make its structure more interpretable. The other is to include the explanation generation module in the model.

Many methods have emerged to make the structure of neural networks more interpretable. Ref. [79] proposed capsule network, which improved the traditional CNN. Capsule network replaces the neurons of a traditional neural network with a vector (called capsule), which can detect a specific pattern. Capsule network can be regarded as a specific semantic network structure. The weighted routing relationship between capsule nodes can explain the spatial relationship between detected objects, reflecting the causal correlation interpretation. Ref. [80] designed a physical information neural network (PINN) to incorporate physical prior knowledge for deep learning. PINN approximates the solution of a set of partial differential equations with initial and boundary conditions. The loss of PINN includes errors in initial and boundary conditions, as well as errors in partial differential equations. PINN enhances interpretability through the action of automatic differentiation. The knowledge graph regards each element in the dataset as an entity, and there is a path between entities. Knowledge graph reveals relationships between adjacent entities by encoding contextual information, intrinsically supporting reasoning and causality. Ref. [81] combined knowledge graph and long short-term memory (LSTM) network to propose knowledge path recurrent network (KPRN). KPRN can directly exploit the entity relations on the path for interpretation.

Some neural network models incorporate explanation generation modules into network training. While completing the prediction task, it can also generate feature summary explanations or human-understandable visual or natural language explanations. The attention mechanism is a strategy that enables neural network to output feature summaries explanation [82]. Attention mechanism can be added to most neural networks and endow the model with the ability to distinguish key important information. The attention mechanism is currently widely used in image processing, natural language processing, time series prediction and other fields. With attention weights, the attention mechanism can well interpret the alignment relationship between input and output. In addition to feature importance, there are also important feature subset explanations. Ref. [83] introduced a self-explanatory neural network in which an explanation generation module generates a subset of important features for each sample. This sample makes predictions only based on important features. On the other hand, it is also possible to generate an explanation that is directly understandable to humans. Ref. [84] proposed a framework for deep visual interpretation using natural language, combining classification and interpretation models to visually explain the basis for the predicted label given by the image. Ref. [85] proposed a method for generating multimodal explanations that include both visual and textual explanations. Multimodality can promote each other to improve the quality of interpretation. Ref. [86] introduced the teaching explanations for decisions (TED) framework for generating local explanations that satisfy human mental models. TED utilizes explanation production components to generate domain-specific explanations that reflect the reasoning process of human experts in a particular domain.

3.3. Post-Model

Post-model interpretability attempts to explain the trained ML model. Due to the increasing complexity of ML models, post-model interpretability has become the main direction of current interpretable ML research and is mainly focused on the field of deep learning [87]. According to the different interpretation objects, post-model interpretability is mainly divided into three types: interpretation of model, interpretation of prediction and mimic model. Table 1 summarizes the post-model interpretability methods, giving common methods and their interpretation forms.

Table 1. Summary of post-model interpretability methods.

Type	Medium	Representative Method	Interpretation Form
Interpretation of model	Hidden layer analysis	DeConvNet [88], Network dissection [89,90]	Visualization of internal pattern
	Activation maximization	Ref. [91]	example
Interpretation of prediction	Sensitivity analysis	PDP [92], ICE [47], ALE plot [93], Influence function [94], Ref. [95], MASK [96]	Feature summary
	Gradient backpropagation	Gradients [97], Guided backpropagation [98], Integrated Gradients [99], VarGrad [100]	Feature summary
	Relevance propagation	LRP [101], DeepLIFT [102]	Feature summary
	Shapley Values	KernelSHAP [103], TreeSHAP [104]	Feature summary
	Activation map	CAM [105], Grad-CAM [106] Grad-CAM++ [107]	Feature summary (saliency map)
	Conceptual attribution	TCAV [108], ACE [109]	Conceptual feature summary
	Counterfactual explanation	Ref. [110]	example

Table 1. *Cont.*

Type	Medium	Representative Method	Interpretation Form
Mimic model	Global mimic model	Model distillation [111], Tree regularization [112]	Intrinsically intepretable model for all samples
	Local mimic model	LIME [113], DLIME [114], Anchor [115]	Intrinsically Intepretable model for local area

3.3.1. Interpretation of Model

The main purpose of interpretation of model is to understand the inner working mechanism of the neural network and the learned meaning of the hidden layers. Common interpretation methods of model are hidden layer analysis and activation maximization.

The main purpose of hidden layer analysis is to analyze and visualize the semantics learned by the hidden layers in the neural network. This approach can help people generate deep insights into the internal structure of deep networks and build an interactive system. Ref. [88] visualized the features of each hidden layer of CNN using the deconvolution network (DeConvNet). The features learned by each convolutional layer are visually presented. The first few layers of CNN mainly learn background information, and the higher the number of layers, the more abstract the learned features. Going a step further, we can analyze abstract concepts learned by individual neurons. Ref. [89] proposed a framework for network dissection. They quantified the semantics learned by individual neurons in neural networks used in the image domain by analyzing network changes when neurons were activated or deactivated. Ref. [90] analyzed the role of individual neurons of neural networks used in the field of natural language processing. They studied their linguistic meaning by visualizing the saliency maps of the neurons that had the greatest impact on output.

The goal of activation maximization is to find an input pattern that maximizes activation for a given neuron. The input pattern to which a neuron responds maximally may be a good first-order representation of what a neuron is doing [91]. This is an optimization problem that can be defined as:

$$x^* = \arg\max(f_l(x) - \lambda\Omega(x)), \tag{4}$$

where $f_l(x)$ is the activation of a neuron in the lth layer of neural network under the input x, Ω is an optional regularizer. Analyzing the generated prototype sample x^* can help us understand what the neuron learned. When we analyze the maximum activation of the output neuron, we can find a prototype corresponding to a certain class. However, the activation maximization method can only be used to optimize continuous data, and it is difficult to directly use it in natural language processing models.

3.3.2. Interpretation of Prediction

Interpretability methods of prediction mainly study the sensitivities or contributions of features (including user-defined advanced features) to predictions. It includes methods such as sensitivity analysis, gradient backpropagation, relevance propagation, shapley Values, activation map, conceptual attribution, counterfactual explanation, etc.

Sensitivity analysis refers to a method to study the degree of influence of input changes on output [116]. Sensitivity analysis gives explanations in the form of a feature summary, which can be global or local. Classical global sensitivity analysis methods include partial dependence plot (PDP) [92], individual conditional expectation (ICE) [47], accumulated local effects (ALE) plot [93], etc. PDP can show the global impact of specific features on the prediction results of the model. PDP can be obtained by calculating the average of the predictions of the original model for each sample set. ICE characterizes the relationship between individual prediction and a single feature. An ALE plot can describe the average influence of features on predictions. ALE is more practical as it gets rid of the constraints of feature independence. Local sensitivity analysis studies the impact of a specific sample change on its prediction. Ref. [94] evaluated the importance of a training sample through the influence function, which is defined as the derivative of the parameter change to the small change of the sample. Some local sensitivity analysis methods treat the model to be explained as a black box, and only need to know the output of the model for a certain input. Ref. [95] determined the sensitivity of the feature to the prediction by the change of the prediction before and after deleting a feature. Ref. [96] proposed an image sensitivity analysis method based on MASK by perturbing different regions of the image to be explained, the most significant part of its predicted value is found as a saliency explanation.

Gradient backpropagation-based methods exploit the back-propagation of gradients in neural networks to understand the impact of changes in the input on the output. Gradients [97] is a classic gradient attribution method, which uses the gradient of the input layer as the importance of pixels to generate saliency maps. Guided backpropagation [98] combines the deconvolutional nets [88] with Gradients and corrects the gradient of the ReLU by discarding negative values during backpropagation. Ref. [99] proposed Integrated Gradients, which effectively addresses misleading interpretations due to vanishing gradients by integrating relative gradients instead of a single gradient. In addition, the saliency map generated by the gradient backpropagation method usually has more noise. VarGrad [100] produces higher quality saliency maps by averaging the interpretations of multiple noisy copies of the image.

Layer-wise relevance propagation (LRP) is an interpretability method based on deep Taylor decomposition [101,117]. LRP distributes prediction scores backwards up to the input layer through specialized correlation propagation rules and ultimately determines the contribution of individual features to predictions. Each neuron in each layer of the LRP corresponds to a correlation score. According to the propagation rule, the assignment of each neuron to the lower layers is conserved with the correlation score received from the higher layers. LRP can be applied to various data types as well as various neural networks. Ref. [102] proposed DeepLIFT to improve the LRP method, which improves the quality of saliency maps by defining reference points in the input space and propagating the correlation scores proportionally with reference to the gradient information of neuron activations.

SHAP [103] is a game theory inspired method that attributes the output value to the shapely value of each feature. SHAP has a solid theoretical foundation in game theory, and as such, its explanation has good properties. SHAP quantifies the contribution to the prediction by computing Shapely values for each feature. SHAP explanation has the following form:

$$g(z') = \phi_0 + \sum_{i=1}^{N} \phi_j z'_i, \tag{5}$$

where g is the interpretable model, N is the number of input features, z' represents the presence or absence of the corresponding feature (1 or 0), ϕ_i is the Shapley value, and ϕ_0 is a constant. For a certain feature x_i, the shapley value needs to be calculated by all possible feature combinations, and then weighted and summed, that is:

$$\phi_i(val) = \sum_{S \subseteq \{x_1, \ldots x_N\} \setminus \{x_i\}} \frac{|S|!(N - |S| - 1)!}{p!} (val(S \cup \{x_i\}) - val(S)), \tag{6}$$

where S is the subset of features used for the model, x_i is the ith feature of the sample to be explained, $val(S)$ refers to the model output value under the feature combination S. However, a practical issue is exponential computational complexity, more seriously, the training cost before each call of $val(S)$. To solve this problem, KernelSHAP was proposed to approximate the actual Shapley value in [103]. The workflow of KernelSHAP is shown in the Figure 5. The calculation of the kernel to estimate the SHAP value is as follows:

$$\pi_x(z') = \frac{(N-1)}{(N \text{ choose } |z'|)|z'|(N-|z'|)}, \tag{7}$$

where $|z'|$ represents the number of non-zero features of z'. The loss function used to train the weighted linear model is defined by:

$$L(f, g, \pi_x) = \sum_{z'_k \in Z} \left[f(h_x(z'_k)) - g(z'_k) \right]^2 \pi_x(z'_k), \tag{8}$$

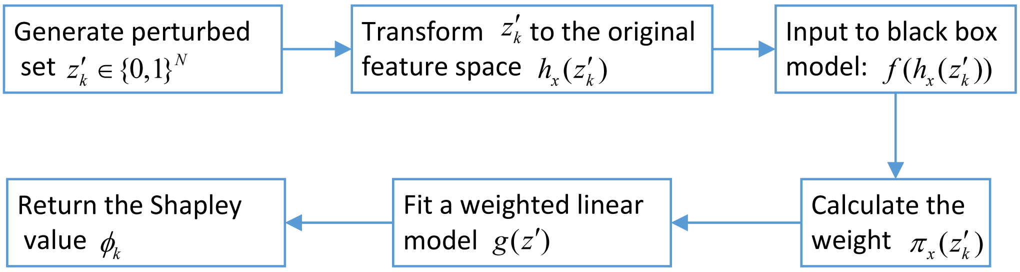
Generate perturbed set $z'_k \in \{0,1\}^N$
Transform z'_k to the original feature space $h_x(z'_k)$
Input to black box model: $f(h_x(z'_k))$
Calculate the weight $\pi_x(z'_k)$
Fit a weighted linear model $g(z')$
Return the Shapley value ϕ_k

Figure 5. The workflow of KernelSHAP.

In 2018, [104] further proposed TreeSHAP for tree-based ML models. TreeSHAP is faster than KernelSHAP and can accurately estimate Shapley values.

Activation map-based interpretability methods are mainly used for interpreting CNN models. They generate pixel-level feature summary in the form of a saliency map by a weighted combination of feature activation maps. Class activation map (CAM) [105] introduces a global average pooling layer instead of a fully connected layer, and then obtains the mean value of each feature map in the last convolutional layer, which is then weighted and summed. Grad-CAM [106] uses a weighted combination of gradients using the activation maps of the last convolutional layer as weights to obtain saliency maps. Grad-CAM++ [107] extends object localization to multiple object instances in a single image, while using the mean of the positive partial derivatives of the last convolutional layer as weights to generate saliency maps.

Attribution methods do not necessarily focus only on raw features, but also on user-defined concepts. Ref. [108] proposed a method called quantitative testing with concept activation vectors (TCAV) to judge the importance of a concept for prediction. They used directional derivatives to quantify the sensitivity of concepts. Automatic concept interpretation (ACE) [109] was proposed to address the subjectivity of manual selection of concepts. ACE starts by segmenting a given image using multiple resolutions. Similar fragments are then grouped as instances of the same concept. Finally, TCAV provides an importance score for a concept.

Counterfactual refers to an instance whose prediction is different from the original instance. A counterfactual explanation aims to obtain a new instance with a different output result by making minimal changes to the input features of the original instance [118]. Counterfactual explanations describe the effects of changing model inputs in specific ways. Reference [110] presents a survey on counterfactual explanations, including properties, generation algorithms, evaluation criteria, etc.

3.3.3. Mimic Model

We can approximate the decisions of the complex original model by training a simple interpretable model which is called mimic model, or surrogate model. Generally, the mimic model is obtained by training with the predictions of the original model (instead of the labels), which can be as faithful as possible to the original model. The mimic model can be global or local.

The core of the global mimic model is to train a simple model to learn the output results of the black-box model, and the interpretation of the model prediction results is realized by understanding the simple model. Figure 6 shows the general principle of the global mimic model. Model distillation [111] is a way to acquire global mimic model which use a simple student model to simulate a complex teacher model. We can train a mimic model by minimizing the error between teacher and student. When using model distillation as a global interpretation method, student models are usually implemented by models that can fit complex functional relationships and are interpretable, such as DTs [112,119,120] and generalized additive models [121]. However, since the mimic model cannot be too complicated and can only approximate the teacher model, it sometimes cannot fully explain the behavior of the original model. For DTs as mimic models, the interpretability becomes worse as the depth of the DT increases. Therefore, we need to comprehensively consider the fit of the DT and the complexity of the DT. Ref. [112] proposed tree regularization with the goal of approximating the model well using shallower DT.

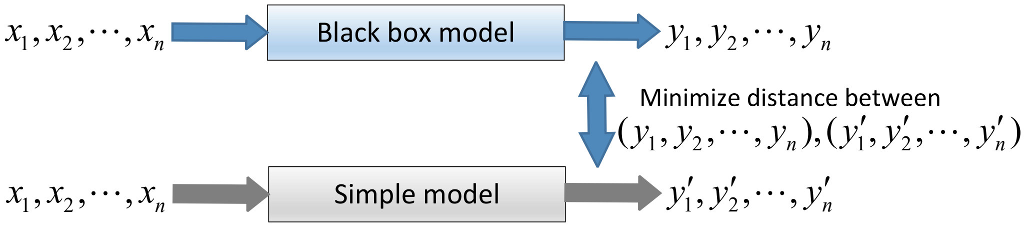
$x_1, x_2, \cdots, x_n$
Black box model
$y_1, y_2, \cdots, y_n$
Minimize distance between
$(y_1, y_2, \cdots, y_n), (y'_1, y'_2, \cdots, y'_n)$
$x_1, x_2, \cdots, x_n$
Simple model
$y'_1, y'_2, \cdots, y'_n$

Figure 6. The general principle of the global mimic model.

In practice, the global mimic model is often difficult to fully explain the original model. So we can consider using a local mimic model. The local mimic model focuses on local area of the samples or an individual sample. The output is interpreted in the form of an interpretable model within the neighborhood of that sample. A typical local mimic model is LIME [113]. Figure 7 shows the process of building a local mimic model using LIME. This method obtains a set of neighbor samples of the target instance by sampling. These samples are then used to train a simple and interpretable model to locally approximate the complex model. The interpretation generated by LIME can be described as:

$$g(x) = \underset{g \in G}{\text{argmin}}\, L(f, g, \pi_x) + \Omega(g), \tag{9}$$

where G is the set of interpretable models, π_x represents the distance measured between sample x and neighboring samples. $L(f, g, \pi_x)$ represents the loss of approximating original model f with g under the weight π_x, $\Omega(g)$ is the complexity of model g. LIME is suitable for interpreting any black-box model.Sampling from a Gaussian distribution can make the interpretation generated by LIME unstable. Ref. [114] proposed a deterministic LIME (DLIME) method to solve this problem. DLIME uses Hierarchical Clustering to group the training data into clusters, and then uses KNN to select the samples closest to the samples to be explained for training. The explanations generated by this sampling method are more stable. Anchor [115] is another local mimic model that uses *if-then* rules (called anchors) as local explanations. The authors define anchors as rules that can adequately make predictions on a local scale, addressing the inability of linear models in LIME to determine coverage.

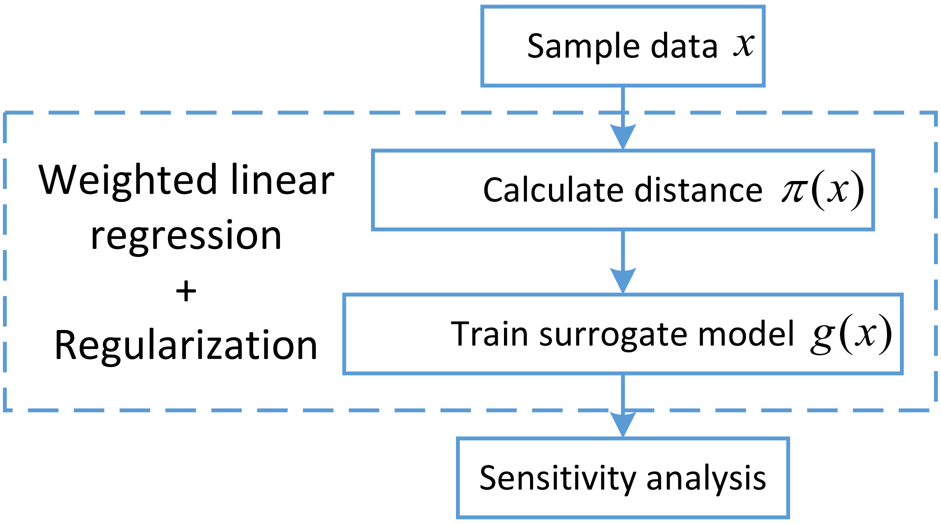
Sample data x
Weighted linear regression + Regularization
Calculate distance $\pi(x)$
Train surrogate model $g(x)$
Sensitivity analysis

Figure 7. The process of building a local mimic model using LIME.

4. Interpretable Machine Learning in Smart Grid

This paper reviews and discusses relevant literature on interpretable ML for smart grid applications. We mainly use Google Scholar for literature collection and focus on publications in the last 5 years. Keywords include smart grid, interpretable/explainable machine learning, explainable deep learning, explainable artificial intelligence, etc. are used for combinatorial search. According to the literature search results, the current interpretable ML applications in the smart grid mainly focus on fault diagnosis, energy forecasting, security and stability analysis, etc.

4.1. Fault Diagnosis

Fault diagnosis plays an important role in power system accident analysis and rapid restoration of power supply. Fault diagnosis includes detection, classification and localization of fault signals [122]. With the increasing complexity and uncertainty of the power grid, the fault characteristics of the power system are no longer obvious, and the traditional mechanism modeling becomes increasingly difficult. At present, ML has been widely used in fault diagnosis [123].

Interpretable ML makes people understand the decisions of ML models and be able to track and locate the cause of fault, which help grid management and reduce losses. In [124], Grad-CAM was used to generate saliency maps of the spectrogram of the three-phase voltage signal. According to the saliency maps, the regions that have the greatest impact on fault classification can be found, which helps in fault localization. Ref. [125] studied the application of LRP to a fault diagnosis model for nuclear power plant reactors and gained insights into feature correlations. Ref. [126] constructed a heterogeneous graph attention network (HGAT) model for multi-source heterogeneous power equipment faults. Interpretation based on graph attention weights improves the confidence of the model. Ref. [127] used graph convolutional network (GCN), which can efficiently exploit power system topology, to construct a search model for critical cascading faults. They explain the diagnostic results by LRP and give the contribution of different fault features. Ref. [128] used a random forest (RF) classifier to identify the fault types of photovoltaic grid-connected systems. SHAP is used to give an explanation of feature importance and identify factors that lead to failures. Ref. [129] proposed a two-layer power transformer fault diagnosis model composed of a binary unbalanced classification model and a multi-classification model. In order to achieve the interpretability of the model, they used TreeSHAP to analyze the correlation between the input features and the diagnosis results. Table 2 summarizes the application of interpretable ML for fault diagnosis.

Table 2. Summary of interpretable ML for fault diagnosis.

Ref.	Year	Application	ML Model	Interpretability Method	Stage	Scope	Model-Specific/Model-Agnostic	Discussion
[124]	2021	Distribution system	CNN	Grad-CAM	Post-model	Local	Model-specific	Visual interpretations produced by Grad-CAM are not fine-grained enough.
[125]	2021	Nuclear power plant	DNN	LRP	Post-model	Local	Model-specific	LRP is more robust than gradients, but less sensitive to target.
[126]	2021	Secondary equipment	HGAT	Attention mechanism	In-model	Local	Model-specific	HGAT contains importance explanations from features to nodes to paths, but attention increases model complexity.
[127]	2021	Cascading failures	GCN	LRP	Post-model	Local	Model-specific	The model is likely to assign contributions to factors that are unrelated to the prediction, and the explanation will lack reliability.
[128]	2021	Grid-connected photovoltaic system	RF	TreeSHAP	Post-model	Local	Model-specific	SHAP is more consistent, and easier to approximate the global interpretation.
[129]	2021	Oil-immersed transformer	XGBoost	TreeSHAP	Post-model	Local	Model-specific	SHAP has a solid theoretical foundation, but the calculation is more complicated.

4.2. Security and Stability Analysis

With the continuous expansion of the scale of the power system and the deepening of the reform of the power market, the security and stability of the power grid has received more and more attention. The process mechanism of power system safety and stability analysis is complex, and the number of influencing factors is huge. ML has advantages in solving complex problems with multiple factors and unknown mechanisms. Therefore, the application of ML technology to power system security and stability analysis has become a research hotspot.

Interpretability is critical for ML-based system security and stability analysis, providing assurance and insights for subsequent system control. To evaluate short-term voltage stability, [130] proposed a shapelet-based spatiotemporal feature learning method to extract key features. Shapelet is a sample-based time series classification method with good interpretability. Ref. [26] used DT to implement a dynamic safety assessment for the power system. They developed two optimization-based tree learning algorithms through disjunctive programming, capable of training high-performance DT while maintaining the interpretability of safety rules. Ref. [131] proposed an improved deep belief network (DBN) for evaluating the transient stability, and proposed a local mimic model-local linear interpretation (LLI) to explain the DBN. Experiments show that LLI can reasonably explain the relationship between input features and system instability. In addition, they also visualized the internal state of the DBN by t-SNE to help operators understand the prediction results.

Ref. [132] developed a fuzzy rule-based classifier for decentral smart grid control (DSGC) stability prediction and achieved an interpretability-accuracy trade-off through a multi-objective optimization algorithm. Ref. [133] used SHAP to analyze the deterministic frequency deviation and its relationship to external characteristics in detail. Ref. [134] used SHAP to identify key characteristics and risk factors for frequency stability in power systems. Ref. [30] constructed a DT-based global mimic model for gated recurrent unit (GRU) model used for transient stability assessment. A new tree regularization is proposed to achieve interpretability. Ref. [135] proposed a neighborhood deep model for total transfer capability evaluation. Quasi-steady state sensitivity analysis method considering the correlation of input variables is proposed to analyze the interpretability. Table 3 summarizes the application of interpretable ML for security and stability analysis.

Table 3. Summary of interpretable ML for security and stability analysis.

Ref.	Year	Application	ML Model	Interpretability Method	Stage of Explanation Generation	Scope	Model-Specific/ Model-Agnostic	Discussion
[130]	2018	Short-Term voltage stability assessment	Shapelet+DT	-	In-model	Global	Model-specific	The model Provides insights into voltage stability assessment from a spatiotemporal perspective, but the explanation is not clear enough.
[26]	2019	Dynamic safety assessment	DT	-	In-model	Global	Model-specific	The accuracy of the DTs is not as high as that of the neural networks.
[131]	2019	Transient stability assessment	DBN	Local mimic model (LLI)	Post-model	Local	Model-agnostic	The interpretation of the local mimic model is not stable enough, and the sampling neighborhood is not easy to determine
[132]	2020	DSGC stability prediction	Fuzzy rule-based classifier	-	In-model	Global	Model-specific	The accuracy of the model is not high, and it is not suitable for high-dimensional large data.
[133]	2021	Deterministic frequency deviations analysis	Boosting model	KernelSHAP	Post-model	Local	Model-agnostic	KernelSHAP is computationally expensive and ignores feature correlations
[134]	2021	Frequency stability assessment	XGBoost	TreeSHAP	Post-model	Local	Model-specific	
[30]	2021	Transient stability assessment	GRU	Global mimic model (DT)	Post-model	Global	Model-agnostic	The global mimic model may not be suitable for all samples, it may be better to divide the area to build multiple mimic tree models
[135]	2021	total transfer capability evaluation	DNN	Quasi-steady state sensitivity analysis	Post-model	Local	Model-agnostic	Sensitivity analysis generally does not take into account the dependencies of variables

4.3. Energy Forecasting

Energy forecasting can provide important information for grid management and electricity market transactions [136]. Energy forecasting generally includes load forecasting, electricity price forecasting, and renewable energy generation forecasting. The use of ML learning techniques has dramatically improved the accuracy of energy forecasts. However, since some energy-related decisions have very high impact, the black-box nature of ML hinders the application of energy prediction models.

Currently, a large portion of interpretable ML in the smart grid is focused on the field of energy forecasting. Ref. [55] proposed an IoT-based deep learning system and a two-step prediction scheme for daily total consumption forecasting problems. They determined the contribution of input features by perturbing the input and presented a good interpretation by generating an impact analysis heatmap. Ref. [137] proposed a reasoning mechanism that can explain individual prediction based on LIME, which breaks the trade-off between model complexity and model interpretability. At the same time, a new performance evaluation index-trust is given to quantitatively evaluate the validity of each prediction. The method is applied to the prediction of building energy performance. Ref. [138] proposed a CNN-LSTM neural network to simultaneously extract spatial and temporal features to effectively predict residential load. By further visualizing key variables using CAM, they determined that heaters and air conditioners had the greatest impact on load. Ref. [139] proposed an autoencoder model to predict load in different situations. They used t-SNE to visualize the hidden states of the model so that they could explain the prediction results. Ref. [140] developed an interactive system based on KNN algorithm for short-term load forecasting. Reference [141] studied Solar power generation forecasting using post-hoc interpretability methods, LIME, SHAP, and ELI5. This paper analyzed the advantages and disadvantages of several post-hoc interpretability algorithms from different aspects.

Ref. [142] proposed a binary classification neural network and a regression neural network for solar power generation prediction. In order to achieve interpretability, they adopted three feature attribution methods, Integrated Gradients, Expected Gradients, and DeepLIFT to evaluate the contribution of features. Ref. [143] introduced a symbolic regression model- QLattice to predict annual building load. Qlattice has a simple and transparent structure, and can directly derive the interaction of different input variables, which is intrinsically interpretable. Table 4 summarizes the application of interpretable ML for energy forecasting. Ref. [144] developed an interpretable memristive (IM) LSTM model for residential load forecasting. This model uses mixture attention mechanism to extract variable and temporal importance, improving the interpretability of time series model for load forecasting.

Table 4. Summary of interpretable ML for energy forecasting.

Ref.	Year	Application	ML Model	Interpretability Method	Stage of Explanation Generation	Scope	Model-Specific/ Model-Agnostic	Discussion
[55]	2017	Daily total consumption forecasting	DNN	Sensitivity analysis	Post-model	Local	Model-agnostic	Authors visualizes the impact by changing a training data, but does not explain the overall decision-making of model.
[137]	2019	Building energy performance forecasting	GLM, MLP, SVM, RF, XGBoost	LIME	Post-model	Local	Model-agnostic	For similar samples, the interpretation of LIME may be less stable.
[138]	2019	Residential load forecasting	CNN-LSTM	CAM	Post-model	Local	Model-specifi	CAM requires the model to have a global average pooling layer, which is inconvenient to use.
[139]	2020	Residential load forecasting	Autoencoder	hidden states visualization (t-SNE)	In-model	Global	Model-specific	Latent variable analysis does not visually show the effect of the input.
[140]	2020	Short-term load forecasting	KNN	-	In-model	Global	Model-specific	The KNN model is not accurate and has low interpretability for time series data.
[141]	2020	Solar power generation forecasting	RF	LIME, SHAP, ELI5	Post-model	Local	Model-agnostic	

Table 4. *Cont.*

Ref.	Year	Application	ML Model	Interpretability Method	Stage of Explanation Generation	Scope	Model-Specific/ Model-Agnostic	Discussion
[142]	2021	Solar power generation forecasting	DNN	Gradients, Expected Gradients, DeepLIFT	Post-model	Local	Model-specific	Explanations produced by gradient-based feature attribution often contain noise.
[143]	2022	Annual building load forecasting	Symbolic regression (Qlattice)	-	In-model	Global	Model-specific	Symbolic regression may not be accurate enough, and interpretability needs to be traded off with sparsity.
[144]	2022	Residential load forecasting	IM-LSTM	Attention mechanism	In-model	Local	Model-specific	The computational complexity of the model is high.

4.4. Power System Flexibility

With the massive access of new energy sources and active loads, the power system needs sufficient adjustment capacity to cope with the imbalance of supply and demand caused by various changes. Based on this, the concept of flexibility of power system is proposed. Flexibility refers to the ability of a power system to reliably maintain power during transients and imbalances [145]. In general, the primary approach to achieving power system flexibility is to integrate rapid supply, demand-side management, demand response, and energy storage systems [146]. ML is an important means to provide flexibility, which can be used in demand-side load and renewable energy generation forecasting, optimal dispatch and control of flexible load, flexible load identification, and user energy consumption pattern analysis [147]. Table 5 summarizes some application of interpretable ML for power system flexibility.

Residential customers can provide considerable flexibility as their energy consumption typically accounts for 70% of total consumption [148]. Generally, power system flexibility is increased through load shifting or load shedding based on demand response signal. Residential load shifting is realized through the control and dispatch of the home energy management system (HEMS). Reinforcement learning (RL) is a type of ML learning method that is often used for scheduling and control of HEMS in the residential sector [149]. RL can learn from interactions and act accordingly to maximize its rewards based on consumer preferences. Therefore, RL has stronger online self-learning ability than other ML methods. Ref. [150] established an interpretable RL model to control the operation of home energy storage devices, which can improve demand-side flexibility and save electricity costs. They interpret the learning process of the agent and the learning strategy based on storage capacity.

Residential building load forecasting is an important basis for realizing load transfer. We can also achieve flexibility estimates by forecasting residential building loads and household renewable energy generation. There are already interpretable ML techniques for estimating household loads, as detailed in the previous subsection. The flexible loads of residential buildings include Air conditioner, water heater, electric vehicle and other controllable household appliances, etc. It is worth noting that we can also forecast for a single flexible load demand. Ref. [151] proposed a building cooling load prediction model based on attention mechanism and RNN. Attention vectors are used to visualize the impact of the input on the predictions, which helps users understand how the model makes predictions. On the other hand, the load monitoring of residential customers can analyze the user's energy consumption habits and power consumption composition, so as to evaluate the flexibility of the power grid and provide a theoretical basis for dispatching. Non-intrusive load monitoring (NILM) only needs to monitor the total voltage and total current at the power inlet and decompose them to obtain the operating status of each subload [152]. This method can not only protect the privacy of customers, but also save a lot of monitoring equipment. There are already interpretable ML techniques for NILM. Ref. [153] combined time-frequency analysis and CNN to solve NILM, and used LRP method to explain what CNN learned. Ref. [154] interpreted deep autoencoder-based NILM models by visualizing activation.

Table 5. Summary of interpretable ML for power system flexibility.

Ref.	**Year**	**Application**	**ML Model**	**Interpretability Method**	**Stage of Explanation Generation**	**Scope**	**Model-Specific/ Model-Agnostic**	**Discussion**
[150]	2019	Control of household energy storage system	RL	-	In-model	Global	Model-specific	The interpretation of the model is not intuitive enough.
[151]	2021	Building cooling load forecasting	RNN	Attention mechanism	In-model	Local	Model-specific	The Model cannot analyze the effect of each feature on predictions.
[153]	2020	NILM	CNN	LRP	Post-model	Local	Model-specific	
[154]	2020	NILM	Autoencoder	visualizing activation	Post-model	Local	Model-specific	This method visualizes the features learned by the hidden layer, but does not explain the overall decision-making of the model.

4.5. Others

With the increasing application of ML in the smart grid, the black-box nature of ML models is gradually being paid attention to. In addition to the above three main aspects, interpretable ML has the following applications in our findings. Ref. [155] used different ML algorithms to establish a prediction model for the diversity factor of the distribution feeder that comprehensively considers various features. The contributions of different features were quantified using SHAP. Ref. [156] developed an interpretable cyber-physical energy system (CPES) based on a knowledge graph, which can integrate multi-source heterogeneous data in the smart grid to generate causal-based explanations. In addition, they demonstrated a demand response-oriented application scenario. In [157], a model based on the attention mechanism and encoder-decoder structure were proposed for area control error prediction in a renewable energy-dominated power system. The variable selection module was designed to provide insights into the relative importance of features. Then, a specially designed attention mechanism can help to better capture temporal dependencies and give temporal importance insights. In [158], a SHAP-based back-propagation deep explanation method was proposed to provide reasonable feature importance explanations for emergency control of power systems based on deep reinforcement learning. Ref. [159] explained the output of a power quality disturbances classifier using occlusion-based sensitivity analysis, Grad-CAM and LIME. They also give a definition of explainability and propose an evaluation process to measure the explainability scores of explainability methods and classifiers. Ref. [160] proposed a nonlinear autoregressive exogenous (NARX) model for anomaly mitigation control models in smart inverter-based microgrids. They employed PDP to account for the effect of features on network output. Table 6 summarizes other applications of interpretable ML in smart grid.

4.6. Case: Interpretable LSTM Model for Residential Load Forecasting

Residential load forecasting is very important to improve the energy efficiency of HEMS. Time series deep learning models, such as LSTM, can significantly improve forecasting accuracy. However, General LSTM networks are a complex model with low interpretability, which is not conducive for customers to further understand the prediction results and respond quickly. Ref. [144] proposed IM-LSTM to solve the problem of residential load forecasting, aiming to improve the interpretability of LSTM-based neuromorphic computing architecture.

Table 6. Summary of interpretable ML for other smart grid applications.

Ref.	Year	Application	ML Model	Interpretability Method	Stage of Explanation Generation	Scope	Model-Specific / Model-Agnostic	Discussion
[155]	2020	Distribution feeder diversity factor prediction	DNN, Gradient boosting tree, RF	KernelSHAP	Post-model	Local	Model-specific	
[156]	2021	CPES modeling	Knowledge graph	-	In-model	Global	Model-specific	The reasoning ability of knowledge graph is insufficient.
[157]	2021	Area control error prediction	Encoder-decoder model	Attention mechanism	In-model	Local	Model-specific	The model does not discern the temporal importance of each feature.
[158]	2021	Emergency control of power system	Deep reinforcement learning	SHAP	Post-model	Local	Model-agnostic	The computational complexity of the model is high and have poor real-time performance.
[159]	2021	Power quality disturbances prediction	CNN	Occlusion-based sensitivity analysis, Grad-CAM, LIME	Post-model	Local	Model-agnostic	For these methods, feature interactions are difficult to consider.
[160]	2022	Anomaly mitigation	NARX	PDP	Post-model	Global	Model-agnostic	The maximum number of features for PDP is 2, and feature dependencies are not considered.

The standard LSTM network represents all input variables as one hidden state. However, the effects of dynamic evolution of different input features on model predictions are indistinguishable due to the common pass through multiple activations. To address this issue, a multivariate LSTM is applied to this architecture to characterize the dynamics of different input variables. The update of the hidden state of the multivariate LSTM is shown in Figure 8. Next, a mixture attention mechanism is used to extract feature importance and feature-wise temporal importance , enabling model-level interpretability. To provide more robust prediction results, the probabilistic prediction based on pinball loss function is built after the mixture attention mechanism. Finally, the authors deployed their proposed interpretable LSTM model on memristors, which improve memory capacity and data transfer bandwidth. The implementation process of the IM-LSTM network is shown in Figure 9.

The task of Ref. [144] is to predict the net load for the next time step including total consumption and solar power generation. Among them, the total consumption consists of lighting, air conditioning (AC), and two other meters. Other input variables include weather, historical statistics of net load, and time-related variables. The global feature importance scores computed by IM-LSTM are given in Figure 10. It can be seen that besides net load, AC and solar power are the two features that contribute the most. This is because AC consumes the most power, while solar power provides a considerable amount of electricity for the home. Figure 11 shows the feature-wise temporal importance in IM-LSTM. It can be seen that the closer the time step, the greater the temporal importance scores. Moreover, the effect of AC varies widely, possibly due to the instantaneous behavior of switching the AC on and off. The knowledge about feature importance and temporal importance produced by IM-LSTM is consistent with the knowledge of experts in the energy domain. To sum up, the introduction of interpretable ML improves the reliability of load forecasting results in smart grids and enhances the trust of people.

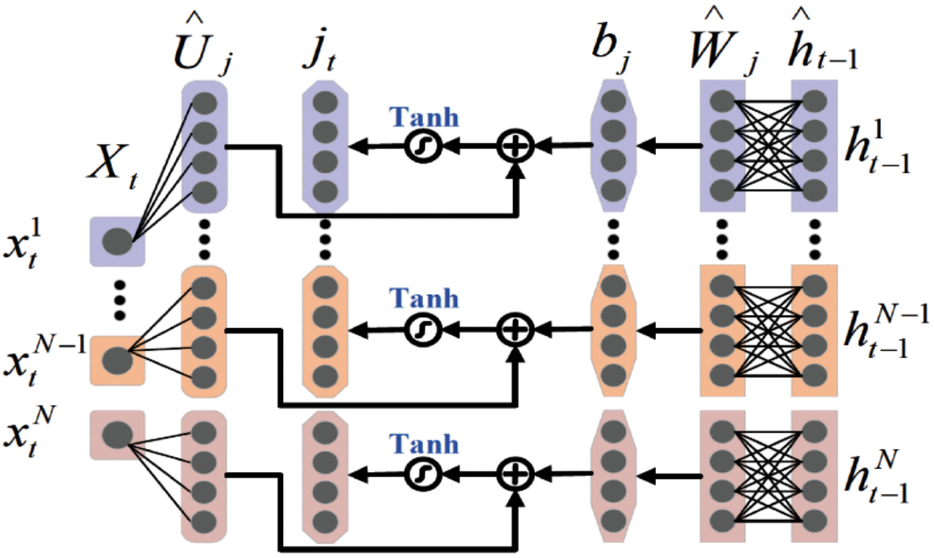

Figure 8. The update of the hidden state of the multivariate LSTM [144].

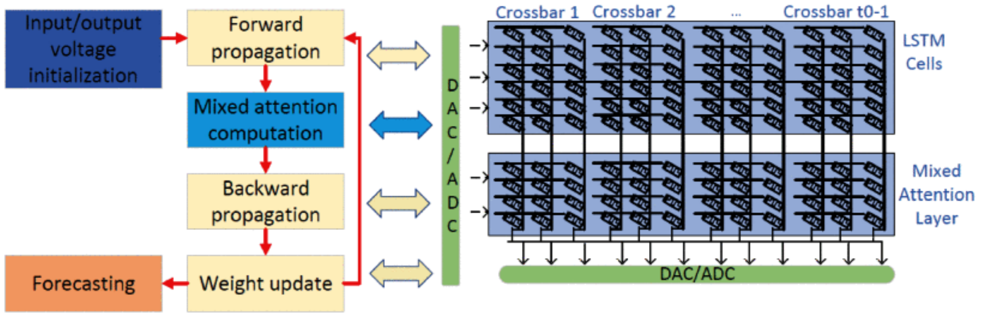

Figure 9. The implementation process of the IM-LSTM network [144].

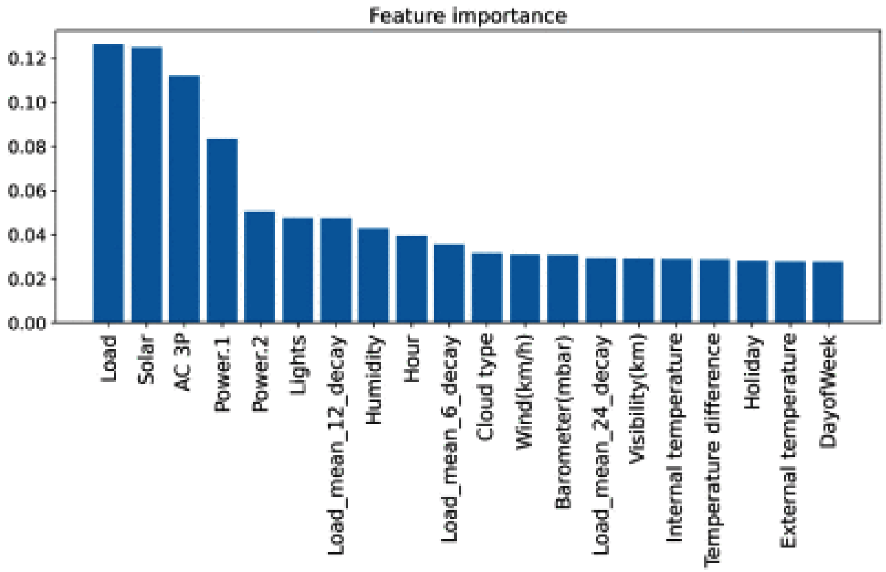

Figure 10. The global feature importance scores in IM-LSTM [144].

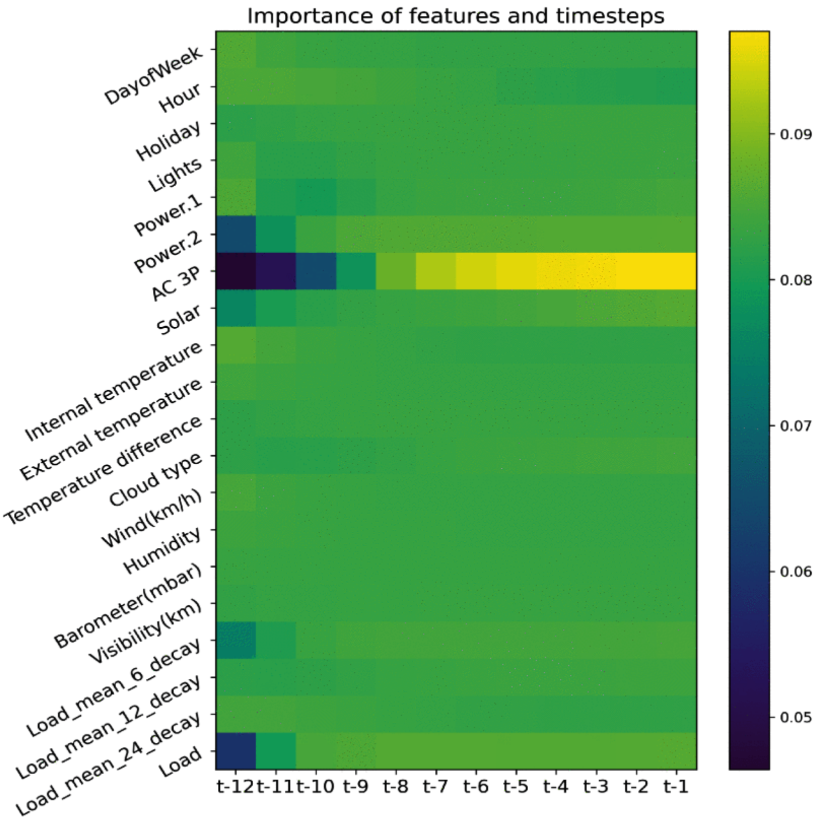

Figure 11. The feature-wise temporal importance in IM-LSTM [144].

5. Future Research Directions

Various ML techniques have achieved notable success in the smart grid. However, the current academic research is still at a very early stage in explaining why and how the model works. From the current research status, researchers generally realized the importance of ML interpretability, and have carried out many very meaningful research studies. Nonetheless, the expansion and application of interpretable ML in the smart grid are still limited. Based on the analysis and understanding of the current research, we believe that the future research on interpretable ML in the smart grid can proceed from the following aspects:

- Interpreting data: The smart grid field uses data from a variety of different sources, including various signals collected in real time from the power system, user information, device data, weather data, and more. Most of the published research focuses on the performance and interpretation of prediction models, ignoring the exploration and understanding of the data. Knowing what is behind the data can help you choose and explore a more suitable model later.
- Embedding domain knowledge: Most ML models in the smart grid provide prediction results using a data-driven approach. Domain knowledge may only be used to validate model decisions rather than being incorporated into models to participate in decision inference. If we embed domain knowledge into model inference, we can obtain more informative explanations. Therefore, it is a promising research direction to combine human knowledge, such as in the form of a knowledge graph, with ML technology to build interpretable ML models.
- Developing more in-model interpretability methods: Benefiting from the excellent characterization performance, complex DL models have been applied to different areas of the smart grid. It is advisable to verify the reliability of the model through post-hoc analysis of feature contributions. However, there is still an open question on how to build intrinsically interpretable deep neural networks without degrading

model performance. In fact, post-model interpretability methods are always difficult to explain the model directly from the internal logic. They are only approximate interpretations of the model and may not be consistent with how the model actually predicted. Therefore, the use of these models in key decision-making areas requires careful consideration. Future work should develop more complex DL models with in-model interpretability.

- Generating human-centered interpretation: An ideal interpretability method should be able to make different interpretations according to the audience's background knowledge and interpretation needs. At the same time, this interpretation should be the logical reasoning process behind the model while giving the decisions. Therefore, extensive research is required to establish appropriate methods to provide personalized interpretations based on the expected user's expertise and abilities.
- Develop interpretable time series models: Most studies of interpretability methods are on images. However, in smart grid applications, much information such as current, load, etc., exists in the form of time series. Therefore, we urgently need to study interpretable ML applied to time series models.
- Applying interpretable ML to more critical areas: In addition to the applications mentioned in the paper, we believe that power dispatch and control, power safety operations, and other user-oriented fields require more support for interpretability.

6. Conclusions

Applying interpretable ML in the smart grid is a promising research direction. Due to the need for transparent and reliable AI systems, this paper reviews interpretable ML and its applications in the smart grid. First, we clarify the definition, motivation, and several properties of interpretability. Next, we detail three types of ML interpretability methods, pre-model, in-model and post-model. Pre-model interpretability methods can help understand the data. In-model interpretability methods are more faithful to the model. Post-model interpretability methods can interpret more complex deep models in different forms. We then review the relevant literature on the application of interpretable ML in key areas of the smart grid, all of which are explicitly motivated by interpretability. We observed that post-model interpretability methods are the primary means of these papers. Finally, we point out some future research directions of interpretable ML committed to realizing a transparent and reliable smart grid. These research directions mainly include interpreting data, establishing more in-model interpretability methods, and realizing human-centered interpretation, etc. In conclusion, with the continuous deepening of research, interpretable ML is bound to play an important role in the smart grid field. We hope this survey can help scholars accelerate research in this area.

Author Contributions: Conceptualization, methodology, validation, writing—original draft preparation, C.X.; writing—review and editing, Z.L.; visualization, R.X.; supervision, X.Z.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Fundamental Research Funds for the Central Universities of Central South University under 2019zzts563.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Air conditioning
ACE	Automatic concept interpretation
AI	Artificial intelligence
CAM	Class activation map
CNN	Convolution neural network
CPES	Cyber-physical energy system

DBN	Deep belief network
DeConvNet	Deconvolution network
DL	Deep learning
DNN	Deep neural network
DSGC	Decentral smart grid control
DT	Decision tree
EDA	Exploratory data analysis
GAM	Generalized additive model
GCN	Graph convolutional network
GDPR	General Data Protection Regulation
GLM	Generalized linear model
GRU	Gated recurrent unit
HEMS	Home energy management system
HGAT	Heterogeneous graph attention network
ICE	Individual conditional expectation
IM-LSTM	Interpretable memristive LSTM
KNN	K-nearest neighbors
KPRN	Konwledge path recurrent network
LIME	Local interpretable model-agnostic explanation
LLI	Local mimic model-local linear interpretation
LR	Linear regression
LRP	Layer-wise relevance propagation
LSTM	Long short-term memory
ML	Machine learning
MLP	Multi-layer perceptron
NARX	Nonlinear autoregressive exogenous
NILM	Non-intrusive load monitoring
PCA	Principal component analysis
PDP	partial dependence plot
PINN	Physical information neural network
ReLU	Rectified linear unit
RF	Random forest
RL	Reinforcement learning
SAE	Stacked autoencoder
SHAP	Shapley additive explanations
SVM	Support vector machine
TCAV	Quantitative testing with concept activation vectors
TED	Teaching explanation for decisions
t-SNE	t-distributed stochastic neighbor embedding
XAI	Explainable artificial intelligence

References

1. Dileep, G. A survey on smart grid technologies and applications. *Renew. Energy* **2020**, *146*, 2589–2625. [CrossRef]
2. Paul, S.; Rabbani, M.S.; Kundu, R.K.; Zaman, S.M.R. A review of smart technology (Smart Grid) and its features. In Proceedings of the 2014 1st International Conference on Non Conventional Energy (ICONCE 2014), Kalyani, India, 16–17 January 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 200–203.
3. Mollah, M.B.; Zhao, J.; Niyato, D.; Lam, K.Y.; Zhang, X.; Ghias, A.M.; Koh, L.H.; Yang, L. Blockchain for future smart grid: A comprehensive survey. *IEEE Internet Things J.* **2020**, *8*, 18–43. [CrossRef]
4. Syed, D.; Zainab, A.; Ghrayeb, A.; Refaat, S.S.; Abu-Rub, H.; Bouhali, O. Smart grid big data analytics: Survey of technologies, techniques, and applications. *IEEE Access* **2020**, *9*, 59564–59585. [CrossRef]
5. Hossain, E.; Khan, I.; Un-Noor, F.; Sikander, S.S.; Sunny, M.S.H. Application of big data and machine learning in smart grid, and associated security concerns: A review. *IEEE Access* **2019**, *7*, 13960–13988. [CrossRef]
6. Azad, S.; Sabrina, F.; Wasimi, S. Transformation of smart grid using machine learning. In Proceedings of the 2019 29th Australasian Universities Power Engineering Conference (AUPEC), Nadi, Fiji, 26–29 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
7. Sun, C.C.; Liu, C.C.; Xie, J. Cyber-physical system security of a power grid: State-of-the-art. *Electronics* **2016**, *5*, 40. [CrossRef]
8. Yohanandhan, R.V.; Elavarasan, R.M.; Manoharan, P.; Mihet-Popa, L. Cyber-physical power system (CPPS): A review on modeling, simulation, and analysis with cyber security applications. *IEEE Access* **2020**, *8*, 151019–151064. [CrossRef]
9. Ibrahim, M.S.; Dong, W.; Yang, Q. Machine learning driven smart electric power systems: Current trends and new perspectives. *Appl. Energy* **2020**, *272*, 115237. [CrossRef]

10. Omitaomu, O.A.; Niu, H. Artificial intelligence techniques in smart grid: A survey. *Smart Cities* **2021**, *4*, 548–568. [CrossRef]
11. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
12. Dobson, A.J.; Barnett, A.G. *An Introduction to Generalized Linear Models*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018.
13. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.
14. Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. *Neurocomputing* **2016**, *195*, 143–148. [CrossRef]
15. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [CrossRef]
16. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An introduction to decision tree modeling. *J. Chemom. A J. Chemom. Soc.* **2004**, *18*, 275–285. [CrossRef]
17. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, 1249. [CrossRef]
18. Gurney, K. *An Introduction to Neural Networks*; CRC Press: Boca Raton, FL, USA, 2018.
19. Doshi, D.; Khedkar, K.; Raut, N.; Kharde, S. Real Time Fault Failure Detection in Power Distribution Line using Power Line Communication. *Int. J. Eng. Sci.* **2016**, *4834*.
20. Gu, C.; Li, H. Review on Deep Learning Research and Applications in Wind and Wave Energy. *Energies* **2022**, *15*, 1510. [CrossRef]
21. You, S.; Zhao, Y.; Mandich, M.; Cui, Y.; Li, H.; Xiao, H.; Fabus, S.; Su, Y.; Liu, Y.; Yuan, H.; et al. A review on artificial intelligence for grid stability assessment. In Proceedings of the 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Tempe, AZ, USA, 11–13 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
23. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, Bellevue, WA, USA, 27 June 2012; pp. 37–49.
24. Aloysius, N.; Geetha, M. A review on deep convolutional neural networks. In Proceedings of the 2017 international Conference on Communication and Signal Processing (ICCSP), Chennai, India, 6–8 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 588–592.
25. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
26. Cremer, J.L.; Konstantelos, I.; Strbac, G. From optimization-based machine learning to interpretable security rules for operation. *IEEE Trans. Power Syst.* **2019**, *34*, 3826–3836. [CrossRef]
27. IqtiyaniIlham, N.; Hasanuzzaman, M.; Hosenuzzaman, M. European smart grid prospects, policies, and challenges. *Renew. Sustain. Energy Rev.* **2017**, *67*, 776–790. [CrossRef]
28. Eskandarpour, R.; Khodaei, A. Machine learning based power grid outage prediction in response to extreme events. *IEEE Trans. Power Syst.* **2016**, *32*, 3315–3316. [CrossRef]
29. Lundberg, J.; Lundborg, A. Using Opaque AI for Smart Grids. Bachelor's Thesis, Department of Informatics, Lund University, Lund, Sweden, 2020.
30. Ren, C.; Xu, Y.; Zhang, R. An Interpretable Deep Learning Method for Power System Dynamic Security Assessment via Tree Regularization. *IEEE Trans. Power Syst.* **2021**. [CrossRef]
31. Ahmad, M.A.; Eckert, C.; Teredesai, A. Interpretable machine learning in healthcare. In Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Washington, DC, USA, 29 August–1 September 2018; pp. 559–560.
32. Garreau, D.; Luxburg, U. Explaining the explainer: A first theoretical analysis of LIME. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Online, 26–28 August 2020; pp. 1287–1296.
33. Mokhtari, K.E.; Higdon, B.P.; Başar, A. Interpreting financial time series with SHAP values. In Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, Markham, ON, Canada, 4–6 November 2019; pp. 166–172.
34. Tjoa, E.; Guan, C. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4793–4813. [CrossRef]
35. Watson, D.S. Interpretable machine learning for genomics. *Hum. Genet.* **2021**. [CrossRef]
36. Rutkowski, T. *Explainable Artificial Intelligence Based on Neuro-Fuzzy Modeling with Applications in Finance*; Springer Nature: Berlin, Germany, 2021; Volume 964.
37. Omeiza, D.; Webb, H.; Jirotka, M.; Kunze, L. Explanations in autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
38. Doshi-Velez, F.; Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv* **2017**, arXiv:1702.08608.
39. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 80–89.
40. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
41. Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Stat. Surv.* **2022**, *16*, 1–85. [CrossRef]

42. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A survey of methods for explaining black box models. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–42. [CrossRef]
43. Shortliffe, E.H.; Buchanan, B.G. A model of inexact reasoning in medicine. *Math. Biosci.* **1975**, *23*, 351–379. [CrossRef]
44. Laurent, H.; Rivest, R.L. Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.* **1976**, *5*, 15–17.
45. Rivest, R.L. Learning decision lists. *Mach. Learn.* **1987**, *2*, 229–246. [CrossRef]
46. Petch, J.; Di, S.; Nelson, W. Opening the black box: The promise and limitations of explainable machine learning in cardiology. *Can. J. Cardiol.* **2021**, *38*, 204–213. [CrossRef]
47. Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **2015**, *24*, 44–65. [CrossRef]
48. Craven, M.; Shavlik, J. Extracting tree-structured representations of trained networks. *Adv. Neural Inf. Process. Syst.* **1995**, *8*, 24–30.
49. Watson, D.S.; Floridi, L. The explanation game: A formal framework for interpretable machine learning. In *Ethics, Governance, and Policies in Artificial Intelligence*; Berlin/Heidelberg, Germany, 2021; pp. 185–219.
50. Zhang, Y.; Tiňo, P.; Leonardis, A.; Tang, K. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**. [CrossRef]
51. Chen, N.; Ribeiro, B.; Chen, A. Financial credit risk assessment: A recent review. *Artif. Intell. Rev.* **2016**, *45*, 1–23. [CrossRef]
52. Obermeyer, Z.; Powers, B.; Vogeli, C.; Mullainathan, S. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* **2019**, *366*, 447–453. [CrossRef]
53. Tsamados, A.; Aggarwal, N.; Cowls, J.; Morley, J.; Roberts, H.; Taddeo, M.; Floridi, L. The ethics of algorithms: Key problems and solutions. *AI Soc.* **2022**, *37*, 215–230. [CrossRef]
54. Zhao, X.; Banks, A.; Sharp, J.; Robu, V.; Flynn, D.; Fisher, M.; Huang, X. A safety framework for critical systems utilising deep neural networks. In I*nternational Conference on Computer Safety, Reliability, and Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 244–259.
55. Li, L.; Ota, K.; Dong, M. When Weather Matters: IoT-Based Electrical Load Forecasting for Smart Grid. *IEEE Commun. Mag.* **2017**, *55*, 46–51. [CrossRef]
56. Van Cutsem, T.; Vournas, C. *Voltage Stability of Electric Power Systems*; Springer Science & Business Media: New York, NY, USA, 2007.
57. Furse, C.M.; Kafal, M.; Razzaghi, R.; Shin, Y.J. Fault diagnosis for electrical systems and power networks: A review. *IEEE Sens. J.* **2020**, *21*, 888–906. [CrossRef]
58. Robnik-Šikonja, M.; Bohanec, M. Perturbation-based explanations of prediction models. In *Human and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 159–175.
59. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine learning interpretability: A survey on methods and metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
60. Tukey, J.W. *Exploratory Data Analysis*; Pearson: Reading, MA, USA, 1977; Volume 2.
61. Liu, H.; Zhong, C.; Alnusair, A.; Islam, S.R. FAIXID: A framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. *J. Netw. Syst. Manag.* **2021**, *29*, 1–30. [CrossRef]
62. Kandel, S.; Paepcke, A.; Hellerstein, J.M.; Heer, J. Enterprise data analysis and visualization: An interview study. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 2917–2926. [CrossRef] [PubMed]
63. Artac, M.; Jogan, M.; Leonardis, A. Incremental PCA for on-line visual learning and recognition. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 781–784.
64. Wattenberg, M.; Viégas, F.; Johnson, I. How to use t-SNE effectively. *Distill* **2016**, *1*, e2. [CrossRef]
65. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
66. Ramsey, P.H. Critical values for Spearman's rank order correlation. *J. Educ. Stat.* **1989**, *14*, 245–253.
67. Ahmed, M. Data summarization: A survey. *Knowl. Inf. Syst.* **2019**, *58*, 249–273. [CrossRef]
68. Kleindessner, M.; Awasthi, P.; Morgenstern, J. Fair k-center clustering for data summarization. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 10–15 June 2019; pp. 3448–3457.
69. Hadi, Y.; Essannouni, F.; Thami, R.O.H. Video summarization by k-medoid clustering. In Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, 23–27 April 2006; pp. 1400–1401.
70. Wang, K.; Zhang, J.; Li, D.; Zhang, X.; Guo, T. Adaptive affinity propagation clustering. *arXiv* **2008**, arXiv:0805.1096.
71. Kim, B.; Khanna, R.; Koyejo, O.O. Examples are not enough, learn to criticize! criticism for interpretability. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2288–2296.
72. Nelder, J.A.; Wedderburn, R.W. Generalized linear models. *J. R. Stat. Soc. Ser. A* **1972**, *135*, 370–384. [CrossRef]
73. Hastie, T.J.; Tibshirani, R.J. *Generalized Additive Models*; Routledge: London, UK, 2017.
74. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 623–631.
75. Sun, R. Robust reasoning: Integrating rule-based and similarity-based reasoning. *Artif. Intell.* **1995**, *75*, 241–295. [CrossRef]

76. Liu, B.; Hsu, W.; Ma, Y. Integrating Classification and Association Rule Mining. In Proceedings of the KDD'98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 27–31 August 1998; Volume 98, pp. 80–86.
77. Gacto, M.J.; Alcalá, R.; Herrera, F. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Inf. Sci.* **2011**, *181*, 4340–4360. [CrossRef]
78. Weinberger, K.Q.; Blitzer, J.; Saul, L. Distance metric learning for large margin nearest neighbor classification. *Adv. Neural Inf. Process. Syst.* **2005**, *18*.
79. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
80. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
81. Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; Chua, T.S. Explainable reasoning over knowledge graphs for recommendation. In Proceedings of the AAAI conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5329–5336.
82. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 25–29 October 2014.
83. Lei, T.; Barzilay, R.; Jaakkola, T. Rationalizing neural predictions. *arXiv* **2016**, arXiv:1606.04155.
84. Hendricks, L.A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; Darrell, T. Generating visual explanations. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–19.
85. Park, D.H.; Hendricks, L.A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; Rohrbach, M. Multimodal explanations: Justifying decisions and pointing to the evidence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 8779–8788.
86. Hind, M.; Wei, D.; Campbell, M.; Codella, N.C.; Dhurandhar, A.; Mojsilović, A.; Natesan Ramamurthy, K.; Varshney, K.R. TED: Teaching AI to explain its decisions. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, Honolulu, HI, USA, 27–28 January 2019; pp. 123–129.
87. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable ai: A review of machine learning interpretability methods. *Entropy* **2020**, *23*, 18. [CrossRef] [PubMed]
88. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
89. Bau, D.; Zhu, J.Y.; Strobelt, H.; Lapedriza, A.; Zhou, B.; Torralba, A. Understanding the Role of Individual Units in a Deep Neural Network. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 30071–30078. [CrossRef]
90. Dalvi, F.; Durrani, N.; Sajjad, H.; Belinkov, Y.; Bau, A.; Glass, J. What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 6309–6317. [CrossRef]
91. Montavon, G.; Samek, W.; Müller, K.R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **2018**, *73*, 1–15. [CrossRef]
92. Cafri, G.; Bailey, B.A. Understanding variable effects from black box prediction: Quantifying effects in tree ensembles using partial dependence. *J. Data Sci.* **2016**, *14*, 67–95. [CrossRef]
93. Molnar, C. *Interpretable Machine Learning*; Lulu: Morrisville, NC, USA, 2020.
94. Koh, P.W.; Liang, P. Understanding black-box predictions via influence functions. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 1885–1894.
95. Robnik-Šikonja, M.; Kononenko, I. Explaining classifications for individual instances. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 589–600. [CrossRef]
96. Fong, R.C.; Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3429–3437.
97. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
98. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv* **2014**, arXiv:1412.6806.
99. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 3319–3328.
100. Adebayo, J.; Gilmer, J.; Goodfellow, I.; Kim, B. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv* **2018**, arXiv:1810.03307.
101. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **2015**, *10*, e0130140. [CrossRef]
102. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017 2017; pp. 3145–3153.
103. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
104. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent individualized feature attribution for tree ensembles. *arXiv* **2018**, arXiv:1802.03888.

105. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2921–2929.
106. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 618–626.
107. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 839–847.
108. Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2668–2677.
109. Ghorbani, A.; Wexler, J.; Zou, J.Y.; Kim, B. Towards automatic concept-based explanations. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
110. Verma, S.; Dickerson, J.; Hines, K. Counterfactual explanations for machine learning: A review. *arXiv* **2020**, arXiv:2010.10596.
111. Tan, S.; Caruana, R.; Hooker, G.; Lou, Y. Distill-and-compare: Auditing black-box models using transparent model distillation. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 1–3 August 2018; pp. 303–310.
112. Wu, M.; Hughes, M.; Parbhoo, S.; Zazzi, M.; Roth, V.; Doshi-Velez, F. Beyond sparsity: Tree regularization of deep models for interpretability. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 1–3 August 2018; Volume 32.
113. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
114. Zafar, M.R.; Khan, N.M. DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv* **2019**, arXiv:1906.10263.
115. Ribeiro, M.T.; Singh, S.; Guestrin, C. Anchors: High-precision model-agnostic explanations. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 1–3 August 2018; Volume 32.
116. Gevrey, M.; Dimopoulos, I.; Lek, S. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecol. Model.* **2003**, *160*, 249–264. [CrossRef]
117. Montavon, G.; Binder, A.; Lapuschkin, S.; Samek, W.; Müller, K.R. Layer-wise relevance propagation: An overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 193–209.
118. Wachter, S.; Mittelstadt, B.; Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL Tech.* **2017**, *31*, 841. [CrossRef]
119. Bastani, O.; Kim, C.; Bastani, H. Interpretability via model extraction. *arXiv* **2017**, arXiv:1706.09773.
120. Che, Z.; Purushotham, S.; Khemani, R.; Liu, Y. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv* **2015**, arXiv:1512.03542.
121. Tan, S.; Caruana, R.; Hooker, G.; Koch, P.; Gordo, A. Learning global additive explanations for neural nets using model distillation. In Proceedings of the CLR 2019 Conference, Minneapolis, MN, USA, 3–5 June 2019.
122. Jiang, H.; Zhang, J.J.; Gao, W.; Wu, Z. Fault detection, identification, and location in smart grid based on data-driven computational methods. *IEEE Trans. Smart Grid* **2014**, *5*, 2947–2956. [CrossRef]
123. Shi, Z.; Yao, W.; Li, Z.; Zeng, L.; Zhao, Y.; Zhang, R.; Tang, Y.; Wen, J. Artificial intelligence techniques for stability analysis and control in smart grids: Methodologies, applications, challenges and future directions. *Appl. Energy* **2020**, *278*, 115733. [CrossRef]
124. Ardito, C.; Deldjoo, Y.; Sciascio, E.D.; Nazary, F.; Sapienza, G. ISCADA: Towards a Framework for Interpretable Fault Prediction in Smart Electrical Grids. In *IFIP Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 270–274.
125. Kim, S.G.; Ryu, S.; Kim, H.; Jin, K.; Cho, J. Enhancing the Explainability of AI Models in Nuclear Power Plants with Layer-wise Relevance Propagation. In Proceedings of the Transactions of the Korean Nuclear Society Virtual Autumn Meeting, Jeju, Korea, 21–22 October 2021.
126. Zhang, K.; Xu, P.; Gao, T.; ZHANG, J. A Trustworthy Framework of Artificial Intelligence for Power Grid Dispatching Systems. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 418–421.
127. Liu, Y.; Zhang, N.; Wu, D.; Botterud, A.; Yao, R.; Kang, C. Searching for critical power system cascading failures with graph convolutional network. *IEEE Trans. Control Netw. Syst.* **2021**, *8*, 1304–1313. [CrossRef]
128. Wali, S.; Khan, I. Explainable Signature-based Machine Learning Approach for Identification of Faults in Grid-Connected Photovoltaic Systems. *arXiv* **2021**, arXiv:2112.14842.
129. Zhang, D.; Li, C.; Shahidehpour, M.; Wu, Q.; Zhou, B.; Zhang, C.; Huang, W. A bi-level machine learning method for fault diagnosis of oil-immersed transformers with feature explainability. *Int. J. Electr. Power Energy Syst.* **2022**, *134*, 107356. [CrossRef]
130. Zhu, L.; Lu, C.; Kamwa, I.; Zeng, H. Spatial–temporal feature learning in smart grids: A case study on short-term voltage stability assessment. *IEEE Trans. Ind. Informatics* **2018**, *16*, 1470–1482. [CrossRef]
131. Wu, S.; Zheng, L.; Hu, W.; Yu, R.; Liu, B. Improved deep belief network and model interpretation method for power system transient stability assessment. *J. Mod. Power Syst. Clean Energy* **2019**, *8*, 27–37. [CrossRef]

132. Gorzałczany, M.B.; Piekoszewski, J.; Rudziński, F. A modern data-mining approach based on genetically optimized fuzzy systems for interpretable and accurate smart-grid stability prediction. *Energies* **2020**, *13*, 2559. [CrossRef]
133. Kruse, J.; Schäfer, B.; Witthaut, D. Exploring deterministic frequency deviations with explainable AI. In Proceedings of the 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aachen, Germany, 25–28 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 133–139.
134. Kruse, J.; Schäfer, B.; Witthaut, D. Revealing drivers and risks for power grid frequency stability with explainable AI. *Patterns* **2021**, *2*, 100365. [CrossRef]
135. Wang, Z.; Zhou, Y.; Guo, Q.; Sun, H. Interpretable neighborhood deep models for online total transfer capability evaluation of power systems. *IEEE Trans. Power Syst.* **2021**, *37*, 260–271. [CrossRef]
136. Kaur, D.; Islam, S.N.; Mahmud, M.; Dong, Z. Energy forecasting in smart grid systems: A review of the state-of-the-art techniques. *arXiv* **2020**, arXiv:2011.12598.
137. Fan, C.; Xiao, F.; Yan, C.; Liu, C.; Li, Z.; Wang, J. A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning. *Appl. Energy* **2019**, *235*, 1551–1560. [CrossRef]
138. Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **2019**, *182*, 72–81. [CrossRef]
139. Kim, J.Y.; Cho, S.B. Electric energy consumption prediction by deep learning with state explainable autoencoder. *Energies* **2019**, *12*, 739. [CrossRef]
140. Grimaldo, A.I.; Novak, J. Combining machine learning with visual analytics for explainable forecasting of energy demand in prosumer scenarios. *Procedia Comput. Sci.* **2020**, *175*, 525–532. [CrossRef]
141. Kuzlu, M.; Cali, U.; Sharma, V.; Güler, Ö. Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. *IEEE Access* **2020**, *8*, 187814–187823. [CrossRef]
142. Lu, Y.; Murzakhanov, I.; Chatzivasileiadis, S. Neural network interpretability for forecasting of aggregated renewable generation. In Proceedings of the 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aachen, Germany, 25–28 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 282–288.
143. Wenninger, S.; Kaymakci, C.; Wiethe, C. Explainable long-term building energy consumption prediction using QLattice. *Appl. Energy* **2022**, *308*, 118300. [CrossRef]
144. Li, C.; Dong, Z.; Ding, L.; Petersen, H.; Qiu, Z.; Chen, G.; Prasad, D. Interpretable Memristive LSTM Network Design for Probabilistic Residential Load Forecasting. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 2297–2310. [CrossRef]
145. Mohandes, B.; El Moursi, M.S.; Hatziargyriou, N.; El Khatib, S. A review of power system flexibility with high penetration of renewables. *IEEE Trans. Power Syst.* **2019**, *34*, 3140–3155. [CrossRef]
146. Luo, X.; Dooner, M.; He, W.; Wang, J.; Li, Y.; Li, D.; Kiselychnyk, O. Feasibility study of a simulation software tool development for dynamic modelling and transient control of adiabatic compressed air energy storage with its electrical power system applications. *Appl. Energy* **2018**, *228*, 1198–1219. [CrossRef]
147. Antonopoulos, I.; Robu, V.; Couraud, B.; Kirli, D.; Norbu, S.; Kiprakis, A.; Flynn, D.; Elizondo-Gonzalez, S.; Wattam, S. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renew. Sustain. Energy Rev.* **2020**, *130*, 109899. [CrossRef]
148. Kouzelis, K.; Tan, Z.H.; Bak-Jensen, B.; Pillai, J.R.; Ritchie, E. Estimation of residential heat pump consumption for flexibility market applications. *IEEE Trans. Smart Grid* **2015**, *6*, 1852–1864. [CrossRef]
149. Mathew, A.; Roy, A.; Mathew, J. Intelligent residential energy management system using deep reinforcement learning. *IEEE Syst. J.* **2020**, *14*, 5362–5372. [CrossRef]
150. Kumar, H.; Mammen, P.M.; Ramamritham, K. Explainable ai: Deep reinforcement learning agents for residential demand side cost savings in smart grids. *arXiv* **2019**, arXiv:1910.08719.
151. Li, A.; Xiao, F.; Zhang, C.; Fan, C. Attention-based interpretable neural network for building cooling load prediction. *Appl. Energy* **2021**, *299*, 117238. [CrossRef]
152. Kelly, J.; Knottenbelt, W. Neural nilm: Deep neural networks applied to energy disaggregation. In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, Seoul, Korea, 4–5 November 2015; pp. 55–64.
153. Houidi, S.; Fourer, D.; Auger, F. On the use of concentrated time–frequency representations as input to a deep convolutional neural network: Application to non intrusive load monitoring. *Entropy* **2020**, *22*, 911. [CrossRef]
154. Murray, D.; Stankovic, L.; Stankovic, V. Explainable NILM networks. In Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring, Online, 18 November 2020; pp. 64–69.
155. Wang, W.; Yu, N.; Shi, J.; Navarro, N. Diversity factor prediction for distribution feeders with interpretable machine learning algorithms. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2–6 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
156. Aryan, P.R.; Ekaputra, F.J.; Sabou, M.; Hauer, D.; Mosshammer, R.; Einfalt, A.; Miksa, T.; Rauber, A. Explainable cyber-physical energy systems based on knowledge graph. In Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems, Online, 18 May 2021; pp. 1–6.
157. Toubeau, J.F.; Bottieau, J.; Wang, Y.; Vallee, F. Interpretable Probabilistic Forecasting of Imbalances in Renewable-Dominated Electricity Systems. *IEEE Trans. Sustain. Energy* **2021**, *13*, 1267–1277. [CrossRef]

158. Zhang, K.; Zhang, J.; Xu, P.D.; Gao, T.; Gao, D.W. Explainable AI in Deep Reinforcement Learning Models for Power System Emergency Control. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 419–427. [CrossRef]
159. Machlev, R.; Perl, M.; Belikov, J.; Levy, K.; Levron, Y. Measuring Explainability and Trustworthiness of Power Quality Disturbances Classifiers Using XAI-Explainable Artificial Intelligence. *IEEE Trans. Ind. Informatics* **2021**, *18*, 5127–5137. [CrossRef]
160. Khan, A.A.; Beg, O.A.; Jin, Y.; Ahmed, S. An Explainable Intelligent Framework for Anomaly Mitigation in Cyber-Physical Inverter-based Systems. *arXiv* **2022**, arXiv:17912006.v1.