Techniques for Interpretable Machine Learning

Mengnan Du, Ninghao Liu, Xia Hu
Department of Computer Science and Engineering, Texas A&M University
{dumengnan,nhliu43,xiahu}@tamu.edu

ABSTRACT

Interpretable machine learning tackles the important problem that humans cannot understand the behaviors of complex machine learning models and how these classifiers arrive at a particular decision. Although many approaches have been proposed, a comprehensive understanding of the achievements and challenges is still lacking. This paper provides a survey covering existing techniques and methods to increase the interpretability of machine learning models and also discusses the crucial issues to consider in future work such as interpretation design principles and evaluation metrics in order to push forward the area of interpretable machine learning.

1. INTRODUCTION

Machine learning models have powered breakthroughs in nearly every field and witnessed a wide range of real-world applications. The best performing accuracy is generally achieved by complex models, such as ensemble models and deep neural networks (DNNs). Despite their astonishing successes, we still lack understanding of their behaviors and how these classifiers arrive at a particular decision. *Interpretable machine learning* gives machine learning models the ability to explain or to present in understandable terms to humans [10]. There is a growing interest among the machine learning community in interpreting machine learning models and gaining insights into their working mechanisms.

Interpretable machine learning techniques are crucial particularly when machine learning models produce some unexpected behaviors. Sometimes, machine learning models may fail unexpectedly and not perform in the way that users would want them to perform. These unexpected model behaviors may frustrate and confuse users, making them wonder why. Moreover, there are lots of domains the decisions made by machine learning models are critical, including autonomous cars, medical diagnosis, etc. In lots of similar scenarios, making a mistake may cause severe consequences. For instance, an unexpected wrong decision made by a machine learning system in medical diagnosis could endanger human life safety and health. From the perceptive of machine learning system developers and researchers, the provided interpretation can help them better understand the problem, the data and why a model might fail, and eventually increase the system safety. For end users, explanation will increase their trust and encourage them to adopt machine learning systems.

Interpretable machine learning techniques can generally be categorized into two different groups: designing interpretable models and post-hoc interpretation, depending on the time when the interpretability is obtained [23]. The goal of designing interpretable models is to construct self-explanatory models which incorporate interpretability directly into the structures of a model. In contrast, post-hoc interpretation aims to provide interpretation for existing machine learning models without architectural modifications. We give an illustrative example in Figure 1 using a five-layer DNN, where interpretable model is designed by adding a new layer with interpretable constraints in order to induce interpretability to DNN, while post-hoc interpretation is provided for the DNN without any architecture modifications. The main difference between these two groups of techniques lies in the trade-off between model accuracy and interpretation fidelity. Interpretable models could provide accurate and undistorted interpretation but may sacrifice model prediction performance to some extent. Post-hoc interpretations are limited in their approximate nature while keeping the underlying model accuracy intact.

Based on the above categorization, we further differentiate two types of interpretability: model level interpretation, and prediction (or instance) level explanation. Model level interpretation means that users can understand how the model works globally by inspecting the structures and parameters of a complex model. Instance level explanation locally examines an individual prediction of a model, trying to figure out why the model makes the decision it makes. Still using the five-layer DNN in Figure 1 as an example, model level interpretation is achieved by understanding the representations captured by the neurons at an intermediate layer of the DNN, while prediction level explanation is obtained by identifying the contributions of each feature in a specific input to the prediction made by DNN. These two types of interpretability bring different benefits. Model level interpretation could illuminate the inner working mechanisms of machine learning models and thus can increase their transparency. Prediction level explanation will help uncover the causal relations between a specific input and its corresponding model prediction. Those two help users trust a model and trust a prediction, respectively.

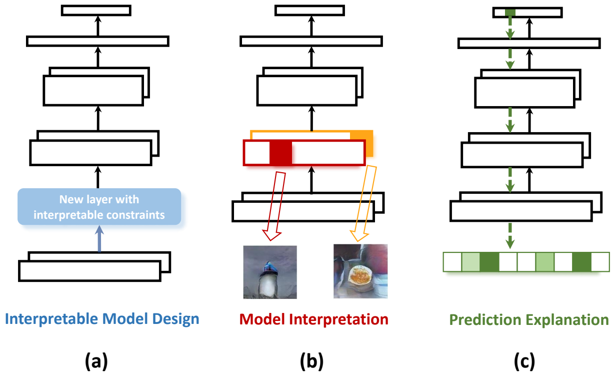

Figure 1: An illustration of three lines of interpretable machine learning techniques, taking DNN for example. (a) Interpretable model design. (b) Post-hoc interpretation of a model. (c) Post-hoc explanation of a prediction.

In this article, we summarize current progress of techniques for interpretable machine learning. In the next three sections, we will focus on the following three lines of research within the field of interpretable machine learning: (1) interpretable model design, including both model level and prediction level; (2) post-hoc interpretation of a model; (3) post-hoc explanation of a prediction.

2. INTERPRETABLE MODEL DESIGN

Interpretable machine learning can be achieved by designing self-explanatory models which incorporate interpretability directly into the model structures. These constructed interpretable models either are globally interpretable or could provide explanations when they make individual predictions.

2.1 Model-level Interpretable Model

Model-level interpretable models can be constructed in two ways: (1) directly trained from data as usual but with interpretability constraints; (2) being extracted from a complex and opaque model.

2.1.1 Adding Interpretability Constraints

In this subgroup, the interpretability of a model is promoted by incorporating interpretability constraints. Some representative examples include enforcing sparsity terms or imposing semantic monotonicity constraints in classification models [13]. Here sparsity means that a model is encouraged to use relatively fewer features for prediction, while monotonicity enables the features to have monotonic relations with the prediction. These constraints make a model simpler and increase the model's comprehensibility by users.

Besides, more semantically meaningful constraints could be added to a model to further improve interpretability. For instance, interpretable convolutional neural networks (CNN) add a regularization loss to higher convolutional layers of CNN to learn disentangled representations, resulting in filters that could detect semantically meaningful natural objects [38]. Another work combines novel neural units, called capsules, to construct a capsule network, which tries to increase the interpretability of CNN from a different perspective [31]. The activation vectors of an active capsule can represent various semantic-aware concepts like position and pose of a particular object present in an image. This nice property makes capsule network more comprehensible for humans and also promising for other applications such as image segmentation and object detection.

However, there are often trade-offs between prediction accuracy and interpretability when constraints are directly incorporated into models. The easier interpretable models may result in reduced prediction accuracy comparing the less interpretable ones.

2.1.2 Interpretable Model Extraction

An alternative is to apply interpretable model extraction, also referred as mimic learning [36], which may not have to sacrifice the model performance too much. The motivation behind mimic learning is to approximate a complex model using an easily interpretable model such as a decision tree, rule-based model, linear model. As long as the approximation is sufficiently close, the statistical properties of the complex model will be reflected in the interpretable model as well. Eventually, we obtain a model with comparable prediction performance, and the behavior of which is much easier to understand. For instance, the ensemble of decision trees can be transformed into a single decision tree [36] or an ordered rule list [9]. Moreover, a DNN is utilized to train a decision tree which mimics the input-output function captured by the neural network so that the knowledge encoded in DNN is transferred to the decision tree [5]. To avoid the overfitting of the decision tree, active learning is applied for training. These techniques convert the original model to a decision tree with better interpretability and maintain comparable predictive performance at the same time.

2.2 Prediction-level Explainable Model

Prediction-level explainable models are usually achieved by designing more justified model architectures that could explain why a specific decision is made. Different from the model-level interpretable models that offer a certain extent of transparency about what is going on inside a model, instance-level explainable models provide users understandable rationale for a specific prediction.

A representative scheme of constructing instance-level explainable models is employing attention mechanism [37, 4], which is widely utilized to explain predictions made by sequential models, e.g., Recurrent Neural Networks (RNNs). Attention mechanism is advantageous in that it gives users the ability to interpret which parts of the input are attended by the model through visualizing the attention weight matrix for individual predictions. Attention mechanism has been used to solve the problem of generating image caption [37]. In this case, a CNN is adopted to encode an input image to a vector, and an RNN with attention mechanisms is utilized to generate descriptions. When generating each word, the model changes its attention to reflect the relevant parts of the image. The final visualization of the attention weights could tell human what the model is looking at when generating a word. Similarly, attention mechanism has been incorporated in machine translation [4]. At decoding stage, the neural attention module added to neural machine translation (NMT) model assigns different weights to the hidden states of the decoder, which allows the decoder to selectively attend to different parts of the input sentence at each step of the output generation. Through visualizing the attention

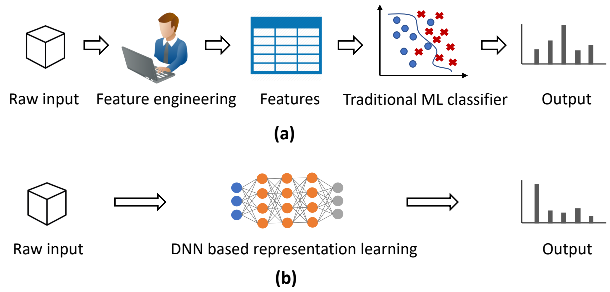

Figure 2: (a) Traditional machine learning pipeline using feature engineering. (b) End to end learning pipeline using DNN based representation learning.

scores, users could understand how words in one language depend on words in another language for correct translation. The attention techniques have also been applied to other domains such as recommender systems [32].

3. POST-HOC INTERPRETING A MODEL

In this section, we summarize techniques to provide post-hoc interpretations for pre-trained models without any modification to their structures. Machine learning models automatically learn useful patterns from a huge amount of training data and retain the learned knowledge into model structures and parameters. The major goal of model interpretation is to provide a global understanding about what knowledge has been acquired by these models, and illuminate the parameters or learned representations of a model in an intuitive manner to humans.

We classify existing machine learning models into two categories, i.e., traditional machine learning pipelines based on feature engineering, and end-to-end pipelines based on DNNs (see Figure 2), since we are capable of extracting some similar interpretation paradigms from each category of models. In the following, we will introduce how to provide interpretation for these two types of pipelines, respectively.

3.1 Traditional ML Interpretation

Models belonging to traditional machine learning pipelines mostly rely on feature engineering, which transforms raw data into features that better represent the underlying problem to the predictive models, as shown in Figure 2(a). The features are generally interpretable and the role of machine learning is to map the representation to output. Arguably, different interpretation schemes should be designed for different kinds of models, since these models may capture distinct interactions among features. We consider a simple yet effective interpretation measure which is widely applicable to most of the models belonging to the traditional machine learning pipeline, called *feature importance*, which indicates the contribution of the features for the underlying model when making decisions.

3.1.1 Model-agnostic Interpretation

Model-agnostic feature importance is broadly applicable to different kinds of machine learning models. It treats a model as a black-box and does not inspect internal model parameters.

A representative approach is *Permutation Feature Importance* [1]. The key idea is that the contribution score of a specific feature to the predictive performance of a model can be determined by calculating how the model prediction accuracy deviates after permuting the values of that feature. More specifically, given a pre-trained machine learning model with n features and a test set, the average prediction score of the model on the test set is p, which is also the baseline accuracy. We shuffle the values of a feature on the test set and compute the average prediction score of the model on the modified dataset. This process is iteratively performed for each feature and eventually n prediction scores are obtained for n features respectively. We then rank the importance of the n features according to the reductions of their score comparing to baseline accuracy p. There are several advantages for this approach. First, we do not need to normalize the values of the hand-crafted features. Second, it can be generalized to nearly any machine learning models with hand-crafted features as input. Third, this strategy has been proved to be robust and efficient of implementation.

3.1.2 Model-specific Interpretation

There also exist methods specifically designed to provide interpretations for different machine learning models. Unlike model-agnostic methods which treat machine learning models as black-boxes, model-specific methods usually derive interpretations by examining internal model structures and parameters. We introduce how to provide feature importance for two families of machine learning models: generalized linear models (GLM) and tree-based ensemble models.

Generalized linear models GLM is constituted of a series of models which are linear combination of input features and model parameters followed by feeding to some transformation function (often nonlinear) [22]. Examples of GLM includes linear regression, logistic regression, etc. The weights of a GLM directly reflect feature importance, so users can understand how the model works by directly checking their weights and visualizing them. However, it is worth noting that the weights may not be reliable when different features are not appropriately normalized and vary in their scale of measurement. Besides, the interpretability of a GLM will decrease when the feature dimensions become too large, which may be beyond the comprehension ability of humans.

Tree-based ensemble models Tree-based ensemble models, such as gradient boosting machines, random forests and XGBoost [7], are typically inscrutable to humans. There are several ways to measure the contribution of each feature. The first approach to measure feature importance is to calculate the accuracy gain when a feature is used in tree branches. The rationale behind is that without adding a new split to a branch for a feature, there may be some misclassified elements, while after adding the new branch, there are two branches and each one is more accurate. The second approach measures the feature coverage, i.e., calculating the relative quantity of observations related to a feature. The third approach is to count the number of times that a feature is used to split the data.

3.2 DNN Representation Interpretation

Deep neural networks, in contrast to traditional machine learning models, not only discover the mapping from representation to output, but also learn *representations* from raw data [14], as illustrated in Figure 2(b). The learned deep representations are usually not human interpretable, hence the interpretation for DNNs mainly focuses on understanding the representations captured by the neurons at intermediate layers of these deep structures. In the following, we introduce interpretation methods for two major categories of DNN, i.e., CNN and RNN.

3.2.1 Interpretation of CNN Representation

In this section, we present methods for interpreting the representations learned by CNN classification models. CNN achieves high prediction accuracy by effectively learning non-linear interactions between features. However, the representations contained in the inner layers of CNN are nearly inscrutable. There has been a growing interest to understand the representations at different layers of CNN.

Among different strategies to understand CNN representations, the most effective and widely utilized one is through finding the preferred inputs for neurons at a specific layer of CNN. This is generally formulated in the *activation maximization* (AM) framework [33]. This process can be formulated as:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}}\, \mathbf{f}_l(\mathbf{x}) - \mathcal{R}(\mathbf{x}), \tag{1}$$

where $\mathbf{f}_l(\mathbf{x})$ is the activation value of a neuron at layer l for input $\mathbf{x}$, and $\mathcal{R}(\mathbf{x})$ is a regularizer. Starting from some random initialization, we optimize an image to maximally activate a neuron's activation status. Through iterative optimization process, the derivatives of the neuron activation value with respect to the image is utilized to tweak the image. After that, the visualization of the generated image will enable us to understand what the individual neuron is looking for in its receptive field. Note that we can in fact do this for arbitrary neurons in the network, ranging from neurons at the first layer all the way to the output neurons at the last layer, so as to understand what is encoded as representations at different layers.

While the framework is simple, getting it to work faces some challenges, among which the most significant one is the surprising artifact. The interpretation process may produce unrealistic images containing noise and high-frequency patterns. Due to the large searching space for images, if without proper regularization, it is possible to produce images that satisfy the the optimization objective to activate the neuron but are still unrecognizable. To tackle this problem, the optimization should be constrained using natural image priors so as to produce synthetic images which resemble natural images. Some researchers heuristically propose some handcrafted priors to incorporate into the optimization process, including total variation [21], α-norm, Gaussian blur, jitter, etc. On the other hand, the optimization could be regularized using stronger natural image priors produced by a generative model, such as GAN or VAE, which maps codes in the latent space to the image spaces [25]. Instead of directly optimizing the image, these methods optimize the latent space codes to find an image which can activate a given neuron that we want to interpret. Experimental results have shown that the priors produced by generative models lead to significant improvements in visualization.

The visualization results provide several interesting observations about the representations learned by CNN intermediate layers. First, the network learns representations at several levels of abstraction, transiting from general to task-specific from the first layer to the last layer. Take the CNN trained with the ImageNet dataset for example. Lower-layer neurons detect small and simple patterns, such as object corners and textures. Mid-layer neurons detect object parts, such as faces, legs. Higher-layer neurons respond to whole objects or even scenes. Interestingly, the visualization of the last (output) layer neurons illustrates that CNN exhibits a remarkable property to capture global structure, local details, and contexts of an object. Second, a neuron could respond to different images that are related to a semantic concept, revealing the multifaceted nature of neurons [27]. For instance, a face detection neuron can fire in response to both human faces and animal faces. Note that this phenomenon is not confined to high layer neurons, all layers of neurons are multifaceted. The neurons at higher layers are more multifaceted than the ones at lower layers. This behavior indicates that neurons at higher layers become more invariant to large changes within a class of inputs, such as colors and poses. Third, CNN learns distributed code for objects [39]. Objects can be described using part-based representations and these parts can be shared across different categories of objects.

3.2.2 Interpretation of RNN Representation

Following numerous efforts to interpret CNN, uncovering the abstract knowledge encoded by representations captured by RNN (including GRUs and LSTMs) has also attracted increasing interest in recent years. Language modeling, which targets to predict the next token given its previous tokens, is usually utilized to analyze the representations learned by RNN. This is due to the reason that accurately predicting the next token requires the model to capture different facets of language such as long-term dependencies. The studies indicate that RNN indeed learns useful representations automatically [18, 17, 28], which partially answer the question why deep neural networks are effective to process sequential data in NLP domain.

First, some work examines the representations of the last hidden layer of RNN and study the function of different units at that layer. To comprehend the function of a unit, these work analyze the real input tokens that maximally activate a unit. The studies demonstrate that some units of RNN representations are able to capture complex language characteristics, e.g., syntax, semantics and long-term dependencies. For instance, a study analyzes the interpretability of RNN activation patterns using character-level language modeling [18]. This work finds that although most of the neural units of RNN are hard to find particular human understandable meanings, there indeed exist certain dimensions in RNN hidden representations that are able to focus on specific language structures such as quotation marks, brackets, and line lengths in a text. In another work, a word-level language model is utilized to analyze the linguistic features encoded by individual hidden units of RNN [17]. The visualization results illustrate that some dimensions of units are mostly activated by certain specific semantic category, while some others could capture a particular syntactic class or dependency function. More interestingly, some hidden units could carry the activation values over to subsequent time steps, which explains why RNN can learn long-term dependencies and complex linguistic features.

Second, the research finds that RNN is able to learn hierarchical representations by inspecting representations at different hidden layers [16, 28]. This observation indicates that RNN representations bear some resemblance to their CNN counterpart. For instance, a bidirectional language model is constructed using a multi-layer LSTM [28]. The analysis of representations at different layers of this model shows that the lower-layer representation captures context-independent syntactic information. In contrast, higher-layer LSTM representations encode context-dependent semantic information. The deep contextualized representations can disambiguate the meanings of words by utilizing their context, and thus could be employed to perform tasks which require context-aware understanding of words.

4. POST-HOC EXPLAINING A PREDICTION

Instead of understanding the model globally, we zoom in to the local behavior of the model regarding a specific instance and provide local explanations for individual predictions. Prediction-level explanation targets to identify the contributions of each feature in the input towards a specific prediction made by a model. As prediction-level explanation methods usually attribute a model's decision to its input features, they are also called *attribution* methods. In this section, we first introduce the model-agnostic attribution methods which could be applied to any machine learning model. Then, we talk about attribution methods specific to DNN based predictions.

4.1 Model-agnostic Explanation

Model-agnostic interpretation methods allow explaining predictions of arbitrary machine learning models independent of the implementation. They provide a way to explain predictions by treating the models as black-boxes, which is advantageous in that explanations may be generated even without access to the internal structure and parameters of a model. They bring some risks at the same time, since we can not guarantee that the interpretation faithfully reflects the prediction process of a model without access to the interior model structures and parameters.

4.1.1 Local Approximation Based Explanation

Local approximation based explanation is based on the assumption that the machine learning predictions around the neighborhood of a given input can be approximated by an interpretable white-box model. The interpretable model does not have to work well globally, but it must approximate the black-box model well in a small neighborhood near the original example. Then the contribution score for each feature can be obtained by examining the parameters of the white-box model.

Some studies assume that the decision score around the neighborhood of an instance could be formulated as the linearly weighted combination of its input features [29]. Attribution methods based on this principle first sample the feature space in the neighborhood of the instance to constitute an additional training set. A sparse linear classifier is then trained using the generated samples and labels. This approximation model works the same as a black-box classifier locally but is much easier to inspect. The prediction of the original model can be explained by examining the weights of this sparse linear model instead. Lundberg and Lee generalize this kind of methods to a unified *additive feature attribution* framework [20] and summarize common principles for this class of attribution methods, which can

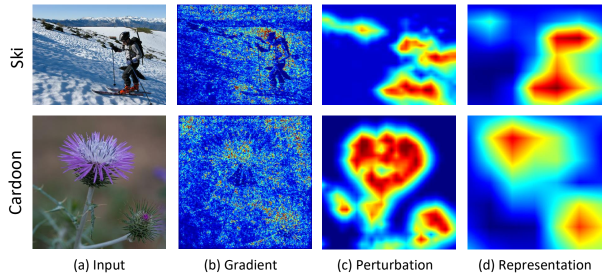

Figure 3: Attribution heatmaps produced by (b) Backpropagation based method, (c) Perturbation based method, (d) Investigation of representations.

be utilized to inform the development of future methods.

Sometimes, even the local behavior of a model may be extremely non-linear, linear explanations could lead to poor performance. Models which could characterize non-linear relationship are thus utilized as the local approximation. For instance, a local approximation based explanation framework can be constructed using if-then rules [30]. Experiments on a series of tasks show that this framework is effective at capturing non-linear behaviors. More importantly, the produced rules are not confined merely to the instance being explained and often generalize to other instances.

4.1.2 Perturbation Based Explanation

Perturbation based explanation methods follow the philosophy that the contribution of a feature can be determined by measuring how output score changes when the feature is altered. More important feature, once being changed, will cause more significant prediction score drop. The perturbation is generally performed across each feature sequentially, so as to determine the contributions of all features. There are different ways to implement the perturbation. The simplest one is directly removing a feature, but this is impractical in practice since very few machine learning models allow to set features as unknown. So instead of removing a feature, we can use occlusion where the feature is set to a reference value, such as setting zero value for word embedding and using a specific gray value for image pixels. Nevertheless, occlusion raises a new concern that new evidence may be introduced and that can be used by the machine learning model as a side effect [8]. For instance, if we occlude part of an image using green color and then we may provide undesirable evidence for the *grass* object. Thus we should be particularly cautious when selecting reference values, in order to avoid introducing extra pieces of evidence [40].

4.2 Model-specific Explanation

In contrast to model-agnostic explanation models, some explanation approaches are designed exclusively for a specific type of machine learning models, such as the tree ensemble models or DNN. In the following, we will introduce explanation methods for DNN, which treat the networks as white-boxes rather than black-boxes and explicitly utilize the interior network structure to derive explanations. We divide DNN-specific explanation techniques into three major categories: back-propagation based methods in a top-down manner; perturbation based methods in a bottom-up manner; investigation of deep representations in intermediate layers. We present an example in Figure 3, showing attribution heatmaps generated by a representative method from each these three categories respectively.

4.2.1 Back-propagation Based Explanation

Back-propagation based methods calculate the gradient (or its variant) of a particular neuron with respect to the input using back-propagation to derive the contribution of features. The underlying hypothesis is that larger gradient magnitude represents a more substantial relevance of a feature to a prediction. In the simplest case, we can visualize the gradient [33], which usually results in noisy and discontinuous saliency maps. Later a few methods are proposed to handle the visual noise problem. One typical approach introduces noise to the input so as to reduce noise in the explanation results [34]. Some other approaches back-propagate different forms of signals to the input, such as discarding negative gradient values at the back-propagation process [35], or back-propagating the relevance of the final prediction score through each layer of the network until the input layer is reached [3]. These back-propagation based methods are integrated into a unified framework where all methods can be reformulated as computing back-propagation using a modified gradient function [2]. This unification enables comprehensive comparison between different attribution methods and facilitates effective implementation under modern graph computational deep learning libraries, such as TensorFlow and PyTorch.

Back-propagation based methods are efficient in terms of implementation, as they usually need a few forward and backward calculations. On the other hand, these methods are limited in their heuristic nature and may generate explanations of unsatisfactory quality, which are noisy and highlight some irrelevant features, as shown in Figure 3 (b).

4.2.2 Perturbation Based Explanation

Perturbation based methods derive attribution scores by perturbing the input to a DNN with minimum noise and observing the change of DNN prediction. Model-agnostic perturbation methods mentioned in the previous section could be computationally very expensive when handling an instance with high dimensions, since they need to sequentially perturb the input. In contrast, DNN-specific perturbation could be implemented efficiently through gradient descent optimization. For instance, one representative work formulates the perturbation in an optimization framework to learn a perturbation mask, which explicitly preserves the contribution values of each feature [12]. Note that this framework generally needs to impose various of regularizations to the mask to produce meaningful explanation rather than surprising artifacts [12]. Although the optimization based framework has drastically boosted the efficiency, generating an explanation still needs hundreds of forward and backward operations. To enable more computationally efficient implementation, a DNN model can be trained to predict the attribution mask [8]. Once the mask neural network model is obtained, it only requires a single forward pass to yield attribution scores for any inputs.

4.2.3 Investigation of Deep Representations

Either perturbation based explanations or back-propagation based explanations ignore the intermediate layers of the DNN that might contain rich information for interpretation. To bridge the gap, some studies explicitly utilize the *deep representations* of the input to perform attribution.

Based on the observation that representations at higher layers of DNN capture the high-level content of the input as well as its spatial arrangement, a guided feature inversion framework is proposed to provide instance-level interpretations of DNN [11]. This framework offers two novel ways to take advantage of the deep neural network representations. First, the representations at higher layers of DNN are inverted to a synthesized image which simultaneously encodes the location information of the target object in a mask and matches the feature representation of the original image. Second, the attribution map is obtained by linearly combining the channels at a high-level layer of the targeted DNN model, since some channels of higher layers of DNN respond strongly to semantically meaningful object parts. This explanation paradigm achieves promising results on a variety of DNN architectures, indicating that the intermediate information indeed contributes significantly to the attribution. Besides, the representations in the intermediate layers serve as a strong regularizer, which increases the possibility of the explanation in characterizing the behaviors of DNN under normal operating conditions. Thus it reduces the risks of generating surprising artifacts, leading to more meaningful explanations.

5. APPLICATIONS

There are numerous applications for interpretable machine learning. In this section, we introduce three representative ones, including model validation, model debugging, and knowledge discovery.

5.1 Model Validation

The first application is model validation, where interpretation could help to examine that machine learning models employ the true cues instead of wrong reasons or bias which widely exist among training data. For instance, a post-hoc attribution approach has been utilized to analyze three question answering models and attributes the models' prediction to words in the question [24]. The attribution visualizations show that these models often ignore the important part of the questions and rely on irrelevant words to make decisions, even though the models' prediction accuracy on the test set is high. They further indicate that the weakness of the model is caused by the inadequacies in test data. Possible solutions to fix this problem include modifying training and testing data or introducing inductive bias to the data. In this case, the attribution method helps validate whether or not the machine learning system has truly understood the task at hand.

5.2 Model Debugging

The second application is model debugging, especially when human have superior performance compared to machine learning models. A representative example is adversarial learning [26]. Recent work demonstrated that machine learning models, such as DNNs, can be guided into making erroneous predictions with high confidence, when processing accidentally or deliberately crafted inputs [26, 19]. When small and imperceptible perturbations are added to images, the CNN's prediction may change dramatically. However, these inputs are quite easy to be recognized by humans. In this case, interpretation facilitates humans to identify the possible model deficiencies and analyze why these models may fail. More importantly, we may further take advantage of human knowledge to figure out possible solutions to promote the performances and reasonability of models.

5.3 Knowledge Discovery

The third application is knowledge discovery, where the derived interpretation allows us to obtain new insights from machine learning systems through comprehending their decision making process. With interpretation, the area experts and the end users could also provide realistic feedbacks. Eventually, new science and new knowledge which are originally hidden in the data could be extracted. For instance, a rule-based interpretable model has been utilized to predict the mortality risk for patients with pneumonia [6]. One of the rules from the model suggests that having asthma could lower a patient's risk of dying from pneumonia. It turns out to be true since patients with asthma were given more aggressive treatments which led to better outcomes.

6. RESEARCH CHALLENGES

Despite some recent progress in interpretable machine learning, there are still some urgent challenges, especially on model design as well as model evaluation.

6.1 Model Design

The first challenge is related to the model design, especially for post-hoc interpretation models. We argue that an interpretation method should be restricted to truly reflect the model behavior under *normal operation conditions*. This criterion has two meanings. Firstly, the interpretations should be faithful to the mechanism of the underlying machine learning model to be explained. Post-hoc interpretation methods propose to approximate the behavior of machine learning models. Sometimes, the approximation is not sufficiently accurate, and the interpretation may fail to precisely reflect the actual operation status of the original model. For instance, an interpretation method may give an explanation that makes sense to humans, while actually, the machine learning classifier works in an entirely different way. Second, even when interpretations are of high fidelity to the underlying models, they may fail to represent the model behavior under normal conditions. Model interpretation and surprising artifacts are often two sides of the same coin. The interpretation process could generate examples which are out of distribution from the statistics in the training dataset, including nonsensical inputs and adversarial examples [15], which are beyond the capability of current machine learning models. Take DNN-specific attribution methods for example [12, 8]. The perturbation may produce some modified inputs which are far away from natural image statistics and drive the DNN to generate unexpected outputs. Without careful design, the model interpretations or the post-hoc explanation models may trigger the artifacts of machine learning models, rather than produce meaningful interpretations.

6.2 Model Evaluation

The second challenge involves the interpretation evaluation. We introduce below the evaluation challenges for interpretable model design and post-hoc interpretation.

The challenge for interpretable model mainly lies in how to quantify the interpretability of a model. There are broad sets of interpretable models which may be designed according to totally different principles and have various forms of implementations. Take the recommender system field as an example, both interpretable latent topic models and attention mechanism [32] could provide some extent of interpretability. In this case, however, how can we compare the interpretability between model-level interpretable model and instance-level explainable model? There is still no consensus on what interpretability means and how to possibly measure the interpretability. Finale and Been provide three types of evaluation approaches for interpretability: application grounded metric, human grounded metric, and functionally grounded metric [10]. These three metrics are complementary to each other and bring their own pros and cons regarding the degree of validity and the cost to perform an evaluation. Adopting which metric or which combination of metrics usually depends on the task at hand so as to make more informed evaluations.

For post-hoc interpretation, comparing to evaluate its interpretability, it is equally important to assess the faithfulness of the explanation to the original model, which is often omitted by existing literature. Take human-grounded interpretability evaluation for example. Some studies assess the attribution performance for CNN-based prediction by comparing the attribution heatmaps with human-annotated ground truth bounding boxes. It brings some risks. The consistency of explanation with human labels doesn't mean it is a reasonable explanation since models may not always use the same evidence as humans to make a decision. Besides, it is hard to tell whether an unexpected explanation is caused by misbehavior of the model or limitation of the explanation method. Therefore, better metrics to measure the faithfulness of explanations are needed, in order to complement existing evaluation methods. The degree of faithfulness can determine how confident we can trust a explanation. Nevertheless, the design of appropriate faithfulness metric remains an open problem and deserves further investigation.

7. CONCLUSIONS

Interpretable machine learning is an open and active field of research, with numerous interpretation approaches continuously emerging every year. We give a clear categorization and comprehensive overview of existing techniques for transparent and interpretable machine learning, aiming to help the community to better understand the capabilities and weaknesses of different interpretation approaches. Although the techniques of interpretable machine learning are advancing quickly, some key challenges remain unsolved, and future solutions are needed to further promote the progress of this field.

8. REFERENCES

[1] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[2] M. Ancona, E. Ceolini, C. Oztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*, 2015.

[5] O. Bastani, C. Kim, and H. Bastani. Interpretability via model extraction. *Fairness, Accountability, and Transparency in Machine Learning Workshop*, 2017.

[6] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2015.

[7] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM.

[8] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6970–6979, 2017.

[9] H. Deng. Interpreting tree ensembles with intrees. *International Journal of Data Science and Analytics*, 2018.

[10] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. 2017.

[11] M. Du, N. Liu, Q. Song, and X. Hu. Towards explanation of dnn-based prediction with guided feature inversion. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[12] R. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *International Conference on Computer Vision (ICCV)*, 2017.

[13] A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 2014.

[14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015.

[16] J. Howard and S. Ruder. Fine-tuned language models for text classification. *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[17] A. Kádár, G. Chrupała, and A. Alishahi. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780, 2017.

[18] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *International Conference on Learning Representations Workshop*, 2016.

[19] N. Liu, H. Yang, and X. Hu. Adversarial detection with model interpretation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[20] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4768–4777, 2017.

[21] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[23] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[24] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere. Did the model understand the question? *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[25] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[26] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *International Conference on Machine Learning Workshop*, 2016.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *Proceedings of NAACL-HLT*, 2018.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2016.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. AAAI Conference on Artificial Intelligence (AAAI), 2018.

[31] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[32] S. Seo, J. Huang, H. Yang, and Y. Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *ACM Conference on Recommender Systems (RecSys)*. ACM, 2017.

[33] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations Workshop*, 2014.

[34] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *International Conference on Machine Learning Workshop*, 2017.

[35] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *International Conference on Learning Representations workshop*, 2015.

[36] G. Vandewiele, O. Janssens, F. Ongenae, F. De Turck, and S. Van Hoecke. Genesim: genetic extraction of a single, interpretable model. *Advances in Neural Information Processing Systems Workshop*, 2016.

[37] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

[38] Q. Zhang, Y. N. Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*, 2015.

[40] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *International Conference on Learning Representations (ICLR)*, 2017.