Knowledge-based XAI through CBR: There is more to explanations than models can tell

Rosina O. Weber[1], Manil Shrestha[2], and Adam J Johs[1]

Information Science[1], Computer Science[2]
Drexel University, Philadelphia, PA 19104
{rw37, ms5267, ajj37}@drexel.edu

Abstract. The underlying hypothesis of knowledge-based explainable artificial intelligence is: the data required for data-centric artificial intelligence agents (*e.g.,* neural networks) are less diverse in contents than the data required to explain the decisions of such agents to humans. The idea is that a classifier can attain high accuracy using data that express a phenomenon from one perspective whereas the audience of explanations can entail multiple stakeholders and span diverse perspectives. We hence propose to use domain knowledge to complement the data used by agents. We formulate knowledge-based explainable artificial intelligence as a supervised data classification problem aligned with the CBR methodology. In this formulation, the inputs are case problems composed of both the inputs and outputs of the data-centric agent, and case solutions, the outputs, are explanation categories obtained from domain knowledge and subject matter experts. This formulation does not typically lead to an accurate classification, preventing the selection of the correct explanation category. Knowledge-based explainable artificial intelligence extends the data in this formulation by adding features aligned with domain knowledge that can increase accuracy when selecting explanation categories.

Keywords: explainable artificial intelligence, knowledge, expertise, interpretable machine learning, case-based reasoning

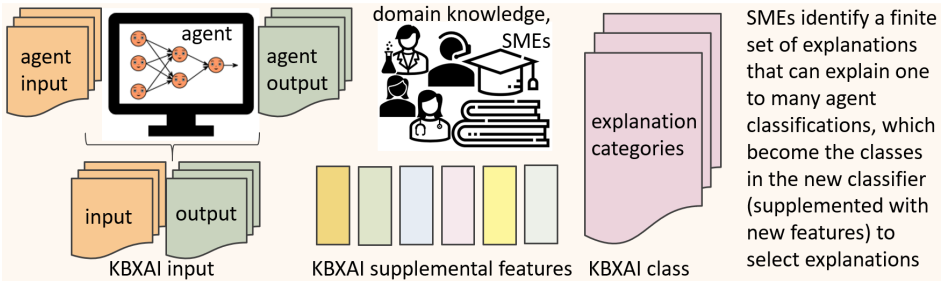

Fig. 1: Overview of KBXAI: Step 1 obtains explanation categories; Step 2 supplements the data with new features to select explanations to explain classifications

1 Introduction

Explainable artificial intelligence (XAI) (*e.g.*, [1]) is a sub-field of artificial intelligence (AI) research that arose with substantial influence from interpretable machine learning (IML) (*e.g.,* [2, 3]). The focus of IML has always been (*e.g.,* [4]) to promote interpretability to ML experts who need to comprehend how concepts are learned to advance the state of the art. The research in IML can be broadly categorized as *feature attribution* (*e.g.,* [5]), *instance attribution* (*e.g.,* [6]), and *example-based* (*e.g.,* [7]). XAI has been proposed [8] with a focus on explainability to users—despite this distinction, most methods for XAI are limited to considering only the interpretability of models (*e.g.,* [5, 6, 9]).

The literature in XAI implies that seeking explanations solely within AI models is insufficient. This implication is supported by various authors (*e.g.,* [10, 11] who have provided contents of explanations not available in models (See Section 2.1 for detailed discussion). Knowledge-based explainable artificial intelligence (KBXAI) is an approach to XAI that seeks to bridge this gap by acquiring knowledge for explanations from domain knowledge and subject matter experts (SMEs). In this paper, we introduce and describe how to implement KBXAI with case-based reasoning (CBR). KBXAI (See Fig. 1) is implemented in two steps: 1) defining explanation categories, and 2) case extension learning.

The next section presents background and related works. With the goal of examining challenges and opportunities brought to bear with the introduction of KBXAI, we illustrate and discuss KBXAI in three problem contexts, each with different data types—tabular, image, and text.

2 Related Works

2.1 Explanation Types

In this section, we review works where authors proposed various explanation contents for use in explanations of intelligent agents. This review is not exhaustive but illustrates how the breadth of explanation contents extends beyond models and data.

Lim [10] describes a taxonomy of *explanation types* that include *situation, inputs, outputs, why, why not, how, what if, what else, visualization, certainty*, and *control*. The item *input* refers to external sources an intelligent agent may have used. In the credit industry, for instance, companies purchase hundreds of thousands of credit profiles of unidentified applicants that are not directly considered in the explanations. Another item is *situation*, which Lim (ibid.) exemplifies with an industrial process where an anomaly is presented, triggering an agent's decision. Lim (ibid.) states that some users would like explanations to include what the *normal* process was prior to the anomalous event.

Nunes and Jannach [11] conducted a systematic review of the literature toward understanding the characteristics of explanation content provided to users across multiple intelligent systems. Explanations proposed in the literature were qualitatively coded to identify the types of contents communicated in explanations — 17 types of explanation contents were identified and grouped as: 1) user

preferences and inputs, 2) decision inference process, 3) background and complementary information, and 4) alternatives and their features. Nunes and Jannach (ibid.) consider multiple forms of background and complementary information, mostly external to the data or knowledge used by intelligent agents—*e.g.,* the background information a human would necessitate for a classification instance; this dovetails with Lim's *inputs* and *situation* explanation types.

Chari et al. [12] propose a taxonomy of approaches and algorithms to support user-centered AI system design. This taxonomy includes two components of scientific explanations divorced from AI models and the data used by AI agents: the scientific method and evidence from the literature. Additional contributions to explanation types are found in [13, 14].

2.2 Three main categories of IML and XAI methods

The three main categories of IML and XAI methods are feature attribution [5, 6, 15–18], instance attribution [3, 19–22], and example-based [7, 9, 23–25]—all predicated on obtaining explanations from an agent's model. Attribution methods explain model behavior by associating an input solved by an agent to elements of the model used by that agent, either by looking at the instance features (*i.e.,* feature attribution) or by looking at each instance as an integral component (*i.e.,* instance attribution). KBXAI neither employs attribution nor prescribes reliance on examples. KBXAI may use features from the model, but knowledge external to the model is required, signifying better alignment with an additional category of model-extrinsic methods. In the XAI categorization as *intrinsic* and *post-hoc*, KBXAI is *post-hoc* because it is implemented after rather than contemporaneously to the agent as with intrinsic methods (*e.g.,* [26]).

Feature attribution methods are relatively easy to compute and have risen in popularity (*e.g.,* [5, 16, 17, 15, 6, 18]). Among such methods are LIME [17], which creates perturbations and then fits them to a linear regression to explain a point that participates in the straight line with its coefficients. Saliency methods [5, 6, 16] are widely used to explain image models because such methods afford the construction of heat maps that emphasize regions (*i.e.,* features) of an image where weights are higher. Another prevailing method is SHAP [15], which adds rigor from Shapley values to feature attribution based on perturbations. Such methods have been criticized for producing the same explanation despite noise added to the data or changes made to the models [27, 28]. Feature attribution have been found to not work in neural architectures that use a memory [29].

Instance attribution methods provide the instances associated with a decision [3, 19–22]. These methods have been shown to have multiple uses such as debugging models, detecting data set errors, and creating visually indistinguishable adversarial training examples [19]. In addition to being computationally expensive [3], there are other criticisms to these methods–*e.g.*, attributed instances are often outliers and the sets of instances attributed to different samples have substantial overlap [22]. Methods that select training instances based on some similarity concept as the basis for explanations are known as example- or prototype-based (*e.g.,* [7, 9, 23–25]). Example-based methods are relatively easy

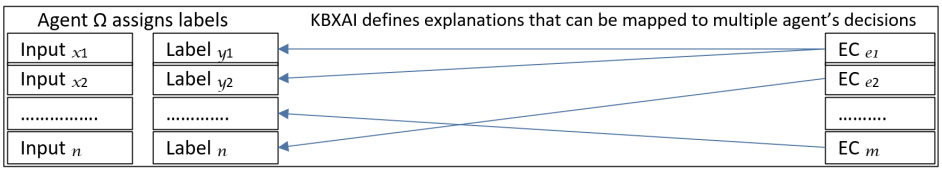

Fig. 2: Explanations are categorized and mapped to agent's input-output pairs

to compute and have been successful in user studies [7, 25, 9]; the core problem with such methods is the absence of attribution.

2.3 Domain knowledge in XAI

Domain knowledge has been used as part of explanation for recommender systems (*e.g.,* [30]), expert systems (*e.g.,* [31, 32]), and CBR systems (*e.g.,* [33–35]. For scientific insights and scientific discoveries, domain knowledge is considered i) a prerequisite for attaining scientific outcomes, ii) pertinent to enhancing scientific consistency, and iii) necessary for explainability [36]. In the biomedical domain, Pesquita [37] proposed augmenting post-hoc explanations with domain-specific knowledge graphs to produce *semantic explanations*.

Contextual decomposition explanation penalization [38] permits insertion of domain knowledge into deep learning with the aim of mitigating false associations, rectifying errors, and generalizing to other methods of interpretability; examples of incorporable domain knowledge range from human labeled ground truth explanations for every data point, to the importance of various feature interactions. [39]'s explanatory interactive learning method leverages human-in-the-loop revision to align model explanations with the knowledge of the expert in-the-loop.

3 Knowledge-based explainable AI

Based on the premise that data used by an agent is to be supplemented, we formulate KBXAI as a problem where input data is given to an agent to execute an intelligent task (for simplicity, henceforth this task is referred to as *classification*). Consider a classifier agent Ω using training instances from an input space X to an output space Y that uses labeled training instances $z_1, ..., z_n \in Z$ where $z_i = (x_i, y_i) \in X \times Y$.

As introduced in [35], KBXAI has two main steps: 1) defining explanation categories, and 2) case extension learning. Fig. 2 shows the first step when KBXAI uses domain knowledge to define a finite set of explanation categories (EC) $e \in EC$, which are defined by a mapping def: $Z \times EC \rightarrow 0$ or 1. We refer to these explanations as categories because they are meant to explain one to many classifications. Within KBXAI, the explanations are textual even when explaining images to facilitate incorporation of supplemental features.

This step creates a new classification problem, which we formulate as case-based. The case problems are the inputs and outputs (i.e., or labels) of the

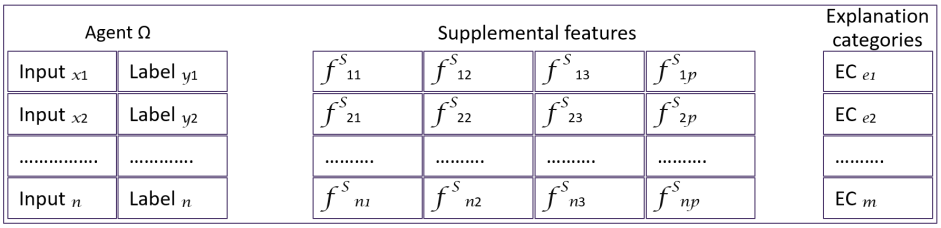

Fig. 3: Supplemental features contribute to select accurate explanation categories

agent. The case solutions are the explanation categories that we wish to select for each agent's classification. This formulation produces low accuracy because it is typically indeterminate. The reason for this is that the explanations include contents that are not in the data and model used by the agent Ω. The goal of KBXAI is *to successfully select the correct explanation category for a given input-output pair*. This prompts the need for the next step.

The second step, case extension learning, is when KBXAI supplements the data by proposing and evaluating supplemental features. The aim is to find features that improve the baseline accuracy to successfully select the correct explanation category for a given input-output pair (see Fig. 3). Proposing features from domain knowledge represents a knowledge engineering step.

3.1 Case-based implementation

We implement case extension learning with CBR. There are two main reasons for not adopting a data-centric approach like neural networks (NN), namely, lack of transparency and sample distribution discrepancy. When comparing the performance of an NN with and without one or more features, if the testing data used are the same for testing both variations then it places the testing and training at different distributions. This does not conform with the machine learning (ML) principle that testing and training must come from the same distribution (*e.g.,* [40–42]). CBR is transparent and allows evaluation of features without violating the ML principle. Through ablation using weighted k-Nearest Neighbor (kNN) and leave-one-out cross validation (LOOCV), when a feature is included, all instances that are left out in LOOCV include such feature, when excluded, the instances do not include it.

Implementing case extension learning with CBR through ablation is as follows. With the problem formulation as depicted in Fig. 2, we used ReliefF [43] to learn weights with local similarity as either a binary (*i.e.,* equal vs unequal) function or, when applicable, a function that computes the difference between values and normalizes based on the range of observed values. Average accuracy is computed with LOOCV. Baseline accuracy is computed with the agent's inputs and outputs, and explanation categories; this is before adding any supplemental features. We do not always we have access to the representations of the input to the agent. In these situations, we consider input as a nominal feature. Case extension entails proposing candidate supplemental features and evaluating how they impact overall average accuracy with respect to the baseline accuracy. The

supplemental features are evaluated one at a time and then in aggregation. Supplemental features may not be independent to the features that come from the agent's input, which is fine because we analyze their impact and keep the ones that better contribute to increasing accuracy. When features are redundant, they do not increase accuracy proportionally when combined.

Next we describe studies applying KBXAI in three data sets using different data types. The goal of these studies is to assess potential challenges for KBXAI. Each data set was obtained differently and the knowledge used to complement the problems also varies. None of the studies represent a complete real-world scenario where KBXAI could be fully deployed. The increments in accuracy shown are modest. Considering that we found supplemental features that caused accuracy to increase is what demonstrates the KBXAI hypothesis has potential.

3.2 KBXAI in Tabular data

This synthetic data set is a binary classification with labels accept and reject [44]. This data has 54 instances and three features with four allowable values each. The first feature, job stability (X1), corresponds to the job status of the applicant. This feature has integer values [2, 5], where 2 means lack of a job, and values 3, 4, and 5, respectively, that applicant has a job for less than one year, less than 3 years, or more than 3 years. The second feature is credit score, credit score (X2), and has values [0, 3], meaning less than 580, 650, 750 and more than 750. The third feature is the ratio of debt payments to monthly income (X3), with values [0, 3], meaning less than 25%, 50%, 75% and more than 75%.

The agent is a NN architecture with four hidden layers and 512 neuron and ReLU activation layers, ending with a sigmoid activation layer. The loss function is binary cross-entropy and the optimizer used is gradient descent. The classifier reached 100% accuracy, which is likely overfit given that we did not separate data because of the small number of samples.

Agent input-output		Supplemental features						Agent input-output				Supplemental features					
Nominal Feature	Predicted Class	X13	X15	X18	X27	X29	Accuracy	X1	X2	X3	Predicted Class	X13	X15	X18	X27	X29	Accuracy
x	x				x		**28.57%**	x	x	x	x	x	x	x	x	x	**66.07%**
x	x	x	x	x	x	x	26.79%	x	x	x	x					x	58.93%
x	x					x	25.00%	x	x	x	x	x					53.57%
x	x	x					19.64%	x	x	x	x			x			53.57%
x	x			x			19.64%	x	x	x	x				x		50.00%
x	x						19.64%	x	x	x	x		x				46.43%
x	x		x				17.86%	x	x	x	x						53.57%

Fig. 4: Case extension learning for tabular data with nominal input features vs. agent's input. Baseline accuracy is indicated in yellow and maximum in bold

To identify explanation categories, these authors used their own knowledge of credit assessment combined with online resources to identify 15 explanation categories that align to the 54 instances. The explanation categories are hypothetical rules combining feature values in both accept and reject classes. Some example explanation categories are, "*with lowest credit score, either job condi-*

tion and debt have to be excellent or both very good for acceptance"; "*no job and credit score is not excellent then reject"*; and "*despite no job, credit score is excellent then accept"*.

Fig. 4 summarizes two examples of case extension learning with this data set. On the left of Fig. 4, we implement the agent's input as a nominal feature, on the right we use the three features used by the agent. The baseline accuracy is 17.8% and 53.5%, respectively. This is not surprising, given the agent's input are the basis of the agent's learning.

We created 29 features by combining values of subset of features and decisions. We only describe those that improved accuracy. Feature X3 is obtained with a function that assigns 1 when despite the debt-income ratio being greater than 75%, the applicant still is approved for credit. For Feature X15, the function assigns 1 if debt-income ratio is less than 25% and the result is approved. Feature X18 is the same as X13 for rejected applications. Feature X27 is valued 1 when credit score is less than 650 and the decision is accept. Feature X29 requires credit score to be below 750 and debt-income ratio not to be <25% when the class is reject to receive value 1 and is zero otherwise.

(a) (b)

Fig. 5: **(a)** Example explanation category: On the left is input image with true label dog; on the right, two images with high median similarity, originally misclassified by the agent that are selected as the explanation category for the input **(b)**The feature Triangular Markings is valued at 1 when authors agreed they saw the three dots and zero otherwise

With the same supplemental features, the accuracy improves about 60% when using nominal values for input, and about 23% when the input features are used (Fig. 4). We also note the combinations of supplemental features reveal different performances in these two executions, potentially suggesting that some of the supplemental features are redundant with respect to the agent's input. Note how they improved accuracy when considered alone and in combination with other features. The best performing feature changes from X27 to X29 in the two executions.

3.3 KBXAI in Image data

The data for the study with images is a subset of CIFAR-10 [45]. Out of 10 classes, we selected four, namely, dogs, trucks, cats, and horses. We formulated these data as a binary classification of dog or not dog. The entire data set has 5,000 images per class for training and 1,000 for testing. We trained a VGG-16 architecture that reached 85% accuracy for the binary classification.

Agent input-output		Supplemental features					
True Image Category	Predicted Class	Saliency	Truck Similarity	Frog Similarity	Triangular Markings	Two Paws	**Accuracy**
x	x		x	x			**36.67%**
x	x		x	x		x	35.00%
x	x		x	x	x		34.17%
x	x	x	x		x	x	34.17%
x	x			x			32.50%
x	x	x					27.50%
x	x				x		25.83%
x	x		x				18.33%
x	x						24.16%

Fig. 6: For image data, accuracy increased from 24% to 36%

To identify explanation categories, we adopted an example-based strategy for selecting example images for explanations (See Section 2.1). The strategy is to select images that are like the image whose classification we want to explain. The candidate images to be used for explanations are the false negatives produced by the binary VGG-16 architecture classifier. The false negatives are all images of the class dog that were misclassified (outliers). See example in Fig. 5a.

To create explanation categories, we selected a subset of Cifar-10 test instances from the selected classes. For each instance, we computed the cosine between the embedding vectors of each test instance and all false negative dog images from the initial test excluding the image itself. To create the explanation category, we used two candidate images with the highest similarity score. Embedding vectors of images were created with an autoencoder.

After explanation categories are created, we now create the data set that maps the agent's input instances and their classification to their explanation category. Note this step was done as a proxy to having a subject matter expert selecting explanation categories. We used this approach because we did not want to have any of the authors interfere with this selection because we would select supplemental features. This mapping refers to identifying the correct explanation category for each instance. We selected explanation categories for each instance by computing the median value of the cosine similarity between the embedding vectors of the instances and the two images in the explanation category. Once this step was completed, we removed duplicates from the resulting explanation categories. We then removed the explanation categories obtained with lower cosine values. We then examined the mapping of testing instances to explanation categories to select the explanation categories that explained more instances as a measure of their popularity. Finally, we took the 12 most popular explanation categories and randomly selected 10 testing instances mapped to each. The final data set has 120 instances and 12 explanation categories.

For images, we utilized both model and domain knowledge to propose features (Fig. 6). From the model, we computed the image's saliency [5]. Saliency brought the 24% baseline to 25%. Truck similarity is a feature that assigns 1 to truck images and the cosine similarity between the embedding of image instance to embedding of a truck we selected as typical (See left of Fig. 7). This feature is from the data as trucks are also part of the data set.

All other features are from commonsense knowledge, which here replaces domain knowledge. Frog similarity is the cosine similarity between embeddings of image instance and frog image we selected as typical of a frog (See left of Fig. 7). Frog similarity alone increased accuracy from 24 to 32.5%. The two last features were selected based on what the authors perceived in the pictures as possibly explaining the recognition of a dog. Note these images are so blurred that human accuracy is estimated to be around 90% [46]. We found that dogs commonly have their eyes and nose as a triangle, marked in red in Fig. 5b. Analogously, two paws refer to the images where the two front paws of the dog are clearly distinguishable.

Fig. 7: Images of a typical truck and a typical frog selected for features truck and frog similarity

This example shows that the combination of features from the model and from commonsense knowledge together work better to increase accuracy. Individually, frog similarity showed the best performance. Note that frog images were not included in the data, making this a feature that is extrinsic to the model and the data.

Agent input-output		Supplemental features		
Input	Predicted Class	W2V	Size	**Accuracy**
✗	✗	✗	✗	**72.64%**
✗	✗	✗		70.27%
✗	✗		✗	71.54%
✗	✗			63.43%

Fig. 8: Starting from a high baseline of 63.43%, two features increased accuracy to 72.64%

3.4 KBXAI in Textual data

The KBXAI implementation with textual data was was previously published in [35]. The data set was built from a selection of 10 scientific articles. The agent used is a citation recommender [47] that produces articles to be cited in the input article. We submitted the 10 articles as inputs to the recommender 10 times to create 100 cases for KBXAI. The explanation categories were learned from the domain of citation analysis. There are only two explanation categories, namely, *background* and *paraphrasing*. Fig. 8 show results of case extension learning. The baseline of 63.43% was increased to an accuracy of 72.64% with only two features. The baseline accuracy is higher than with previous data probably due to a binary selection of explanation categories. The improvement was only 14.5%. Like in image data, only nominal features were used, which limits accuracy.

4 Concluding remarks

Incorporation of knowledge engineering inevitably carries its problems such as difficulty to scale. However, KBXAI only requires knowledge acquisition to capture the additional perspectives that account to multiple stakeholders; it is not meant to acquire knowledge for an entire agent. The explanations tend to repeat and this is why we group them in categories. An open question is approaches where domain knowledge can be learned to be incorporated into KBXAI. The features we added to the tabular example are functions based on rules. This is aligned with explanation-based learning [48], pointing to a future direction.

The studies in this paper suggest it is necessary to leverage the representation used by the agent and not nominal features to represent the agent's input. This imposes on KBXAI, and consequently on CBR, that it should handle any type of representation. To overcome this, one direction is to adopt ANN-CBR twins [24] and utilize the original representations. The next steps also include implementation in real-world scenarios and evaluation with humans.

Acknowledgments The authors thank the anonymous reviewers for suggestions to improve this paper. Support for the preparation of this paper was provided by NCATS, through the Biomedical Data Translator program (NIH award 3OT2TR003448-01S1). Authors Weber and Shrestha are also partially funded by DARPA-PA-20-02-06-POCUS-AI-FP-023.

References

1. Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, PP:1–1, 09 2018.
2. Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.
3. Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting black box predictions using fisher kernels. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3382–3390. PMLR, 2019.
4. Robert Andrews, Joachim Diederich, and Alan B Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389, 1995.
5. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
6. Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.
7. Conor Nugent and Pádraig Cunningham. A case-based explanation system for black-box systems. *Artificial Intelligence Review*, 24(2):163–178, 2005.
8. David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, 2019.
9. Tomas Folke, Scott Cheng-Hsin Yang, Sean Anderson, and Patrick Shafto. Explainable ai for medical imaging: explaining pneumothorax diagnoses with bayesian teaching. *arXiv preprint arXiv:2106.04684*, 2021.
10. Brian Y Lim. *Improving understanding and trust with intelligibility in context-aware applications*. PhD thesis, Carnegie Mellon University, 2012.
11. Ingrid Nunes and Dietmar Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3):393–444, 2017.
12. Shruthi Chari, Oshani Seneviratne, Daniel M Gruen, Morgan A Foreman, Amar K Das, and Deborah L McGuinness. Explanation ontology: A model of explanations for user-centered ai. In *International Semantic Web Conference*, pages 228–243. Springer, 2020.
13. Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.

14. Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*, 2020.
15. Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.
16. Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
17. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *22nd ACM SIGKDD*, pages 1135–1144, 2016.
18. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.
19. Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.
20. Chih-Kuan Yeh, Joon Sik Kim, Ian EH Yen, and Pradeep Ravikumar. Representer point selection for explaining deep neural networks. *arXiv preprint arXiv:1811.09720*, 2018.
21. Dominique Mercier, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Interpreting deep models through the lens of data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.
22. Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.
23. Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in neural information processing systems*, pages 1952–1960, 2014.
24. Eoin M Kenny and Mark T Keane. Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ann-cbr twins for xai. In *Twenty-Eighth International Joint Conferences on Artifical Intelligence (IJCAI), Macao, 10-16 August 2019*, pages 2708–2715, 2019.
25. Eoin M Kenny, Courtney Ford, Molly Quinn, and Mark T Keane. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 294:103459, 2021.
26. Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
27. Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, and Kim-Been Hardt, Moritz. Sanity checks for saliency maps. In *32nd NeurIPS*, pages 9525–9536, 2018.
28. Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.
29. Anurag Koul, Sam Greydanus, and Alan Fern. Learning finite state representations of recurrent policy networks. *arXiv preprint arXiv:1811.12530*, 2018.
30. Markus Zanker and Daniel Ninaus. Knowledgeable explanations for recommender systems. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 657–660. IEEE, 2010.

31. Moore-JD Swartout, WR. Explanation in second generation expert systems. In *Second generation expert systems*, pages 543–585. Springer, 1993.
32. Michael R Wick and William B Thompson. Reconstructive explanation: Explanation as complex problem solving. In *IJCAI*, pages 135–140, 1989.
33. Ralph Bergmann, Gerd Pews, and Wolfgang Wilke. Explanation-based similarity: A unifying approach for integrating domain knowledge into case-based reasoning for diagnosis and planning tasks. In *European Workshop on Case-Based Reasoning*, pages 182–196. Springer, 1993.
34. Agnar Aamodt. Explanation-driven case-based reasoning. In *European Workshop on Case-Based Reasoning*, pages 274–288. Springer, 1993.
35. Rosina Weber, Adam Johs, Jianfei Li, and Kent Huang. Investigating textual case-based xai. In *LNCS, Springer*, volume 11156, page 431–447, 07 2018.
36. Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216, 2020.
37. Catia Pesquita. Towards semantic integration for explainable artificial intelligence in the biomedical domain. In *HEALTHINF*, pages 747–753, 2021.
38. Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International Conference on Machine Learning*, pages 8116–8126. PMLR, 2020.
39. Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *arXiv preprint arXiv:2001.05371*, 2020.
40. Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758*, 2018.
41. Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6970–6979, 2017.
42. Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457. IEEE, 2017.
43. I. Kononenko and Robnik-Šikonja M. Šimec, E. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.
44. Shideh Shams Amiri, Rosina O Weber, Prateek Goel, Owen Brooks, Archer Gandley, Brian Kitchell, and Aaron Zehm. Data representing ground-truth explanations to evaluate xai methods. *arXiv preprint arXiv:2011.09892*, 2020.
45. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.
46. Tien Ho-Phuoc. Cifar10 to compare visual recognition performance between deep neural networks and humans. *arXiv preprint arXiv:1811.07270*, 2018.
47. Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. Content-based citation recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 238–251, 2018.
48. Gerald DeJong and Raymond Mooney. Explanation-based learning: An alternative view. *Machine learning*, 1(2):145–176, 1986.