HHAI 2024: Hybrid Human AI Systems for the Social Good
F. Lorig et al. (Eds.)

doi:10.3233/FAIA240183

Exploring Large Language Models Capabilities to Explain Decision Trees

Paulo Bruno SERAFIM [a,1], Pierluigi CRESCENZI [a], Gizem GEZICI [b], Eleonora CAPPUCCIO [c,d], Salvatore RINZIVILLO [d] and Fosca GIANNOTTI [b]

[a] *Gran Sasso Science Institute*
[b] *Scuola Normale Superiore di Pisa*
[c] *Università di Pisa, Università di Bari Aldo Moro*
[d] *Istituto di Scienza e Tecnologie dell'Informazione, CNR*

Abstract. Decision trees are widely adopted in Machine Learning tasks due to their operation simplicity and interpretability aspects. However, following the decision process path taken by trees can be difficult in a complex scenario or in a case where a user has no familiarity with them. Prior research showed that converting outcomes to natural language is an accessible way to facilitate understanding for non-expert users in several tasks. More recently, there has been a growing effort to use Large Language Models (LLMs) as a tool for providing natural language texts. In this paper, we examine the proficiency of LLMs to explain decision tree predictions in simple terms through the generation of natural language explanations. By exploring different textual representations and prompt engineering strategies, we identify capabilities that strengthen LLMs as a competent explainer as well as highlight potential challenges and limitations, opening further research possibilities on natural language explanations for decision trees.

Keywords. Explainable AI, decision tree, natural language generation

1. Introduction

The high complexity of novel Machine Learning (ML) models is an important factor in the production of remarkable results since simpler models were not able to achieve the same level of performance. However, as the models grow, there have been growing concerns due to the opaque nature of widely used black-box algorithms. Among the problems that arise from the non-transparency are the trustworthiness, fairness, and accountability of the models [1]. In this context, the need for an approach to bridge the gap between the algorithmic perspective and human-centered explanation is evident.

Among key explanatory elements from disciplines such as social sciences and psychology, interactivity has been described as a core value for the implementation of human-centric explainable AI applications [2]. The explanation process has to be outlined as a continuous dialogue between a sender and a receiver, making the user engage in a conversation [3,4]. More recently, Large Language Models (LLM) have shown an exceptional ability to generate natural language responses [5]. As such, LLMs emerge

[1]Corresponding Author: Paulo Bruno Serafim, paulo.desousa@gssi.it.

as a promising tool to maintain an active dialogue with a user whose goal is to better understand a model decision process.

Decision trees are widely used in classification tasks and recognized for their simplicity, being considered a method interpretable by nature [6,7]. Nonetheless, even with decision trees with a large number of nodes or if the user is not familiar with trees [8]. Both situations demonstrate the usefulness of having a tool to summarize an explanation of the decision taken in non-technical terms. In this paper, we explore the capabilities of LLMs to explain decision trees for non-expert users.

Our contributions are as follows. First, we expand on prior work of generating Natural Language Explanations for decision trees using LLMs by proposing a general explanation pipeline. We explore prompt engineering strategies to evaluate whether the generated explanations are coherent, informative, and easy to understand. We also evaluate how sensible is the LLM to different text representations. Finally, we present further research directions that go toward more human-centered explainable artificial intelligence.

2. Related Works

Generating Natural Language Explanations (NLE) for Machine Learning models has been an active research topic [9]. Among works that generate NLE, we highlight the ones that use decision trees as the classifier. López-Trigo et al. [10] developed a pattern-matching method to generate local and global explanations following specific grammar rules. As an extension of the previous work, Alonso et al. [11] developed the ExpliClas web application to provide explanations to the users.

Graviidilis et al. [12] reported a method to generate natural language explanations by using a Natural Language Generation component in the pipeline. The tree structure is used to define a plan of action, which is then converted to a set of contextualized rules. More recently, Ziems et al. [13] used an LLM to generate NLEs for classification from decision trees trained in a Network Intrusion Detection (NID) problem. The dataset description, feature descriptions, and textual representation of the decision tree are combined to form an input text which is passed to an LLM to provide explanations.

In this paper, we present a method to generate NLE from decision trees using LLMs for any task. We extend the work of Ziems et al. [13] by generalizing the pipeline to further problems, without specific requirements of NID. Moreover, we do not want the explanation to provide background information or technical answers, nor do we ask for them. Compared to the mentioned works, our goal is to explore LLM capabilities to obtain explanations that are easier to understand by a non-expert user.

3. Methodology

There are several steps involved in the process of producing an appropriate natural language explanation from an LLM. First, it is important to note that we are not using models with multi-modal capabilities, as such we only interact with the LLM through textual inputs. In order to obtain adequate responses, it is necessary to provide a well-structured prompt that contains all the required information. We construct the input following a series of subprocesses that leverage values from the task and the instance, supplemented by information that can improve explanation quality.

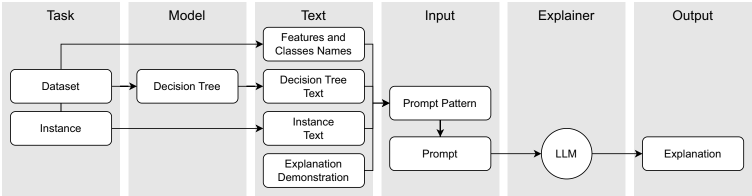

Figure 1. Natural Language Explanation Generation Pipeline

In the general framework, we train a decision tree from the task dataset and then generate a textual tree representation. Together with task features, classes, and an instance, we pass them as a prompt to an LLM. This approach is similar to the one present in Ziems et al. [13], but it is not restricted to Network Intrusion Detection as it was extended to generate explanations from decision trees trained on any task. Lastly, the LLM outputs an explanation, which concludes the generation pipeline, as illustrated in Figure 1.

The prompt contains all the relevant information for the LLM to generate the explanation in a textual format. We predefined a prompt pattern to construct the complete input text by adding specific information from the dataset, tree, and instance. Additionally, we include a demonstration and instructions to guide the LLM in generating the explanation. Each prompt component is described in detail in the following sections and a complete example can be found in Table 1.

Dataset Features and Classes. At the beginning of the prompt, we list the dataset feature names, which are directly extracted from the dataset. The class names comprise the second input to the prompt pattern. Each dataset may have a different number of classes and similar to the feature names, class names are not necessarily available. Together with the features, they form the dataset description.

Decision Tree Representation. As a graph, there are many ways to represent a decision tree. The most common is to use a diagram formed from nodes and directional edges. However, since we are using textual inputs to the LLM, it is necessary to transform the tree into a textual representation. Since we do not predefine a tree template, any externally generated textual representation of a decision tree can be used.

Demonstration. Prior work has shown that LLMs tend to give more appropriate responses when a demonstration is passed in the prompt [14]. As such, we follow these suggestions for better prompting and include an example to be used by the LLM as a guideline for its answer. We created one sample by ourselves which is a depiction of an expected explanation from a given instance.

Instance Description. The instance textual representation is formed by concatenating all feature names and their corresponding values, which is automatically done by the prompt construction script. Therefore, all feature names match exactly the ones that are present in the dataset description. Note that although the instance values have to be consistent with the features, it is not required to be a real instance from the dataset.

Instructions. Once again we refer to good prompt engineering strategies [14,5] in order to obtain more suitable explanations by giving clearer instructions to the LLM. Similar to Ziems et al. [13], we ask the LLM to avoid mentioning the mechanics of the tree and answer in natural language. Additionally, we expect all relevant features to be mentioned, while ignoring unused ones. Finally, we include a sentence to prevent numbers and favor more general terms like ‘high’ and ‘low’ instead.

Question. The last part of the prompt is the question itself. At this moment, all descriptions were extracted and additional information was assembled. Then, we finally ask the LLM to generate explanations that are similar to the demonstration while following the given instructions. An example of a complete prompt can be found in Table 1.

Table 1. Example of prompt constructed according to the pattern.

```
Consider a dataset that has the following features: sepal length (cm), sepal width (cm),
petal length (cm), and petal width (cm). Each instance can be classified into one of the
following classes: setosa, versicolor, and virginica. A decision tree was trained on the
dataset and the following tree was obtained:
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal width (cm) <= 1.55
|   |   |--- petal length (cm) <= 4.95
|   |   |   |--- class: 1
|   |   |--- petal length (cm) >  4.95
|   |   |   |--- class: 2
|   |--- petal width (cm) >  1.55
|   |   |--- petal width (cm) <= 1.70
|   |   |   |--- class: 1
|   |   |--- petal width (cm) >  1.70
|   |   |   |--- class: 2
Given an instance of the iris dataset with features: sepal length (cm) = 3.6, sepal width (
cm) = 0.8, petal length (cm) = 4.3, and petal width (cm) = 2.9, and a confidence value of
57.81 %, a good explanation for why the instance was classified as virginica is: 'By
evaluating the feature values, it is possible to observe that both petal length and petal
width are high. This means that by following the decision tree path, the instance should be
classified as virginica, although the tree is not very confident in this result.'
An instance has features: sepal length (cm) = 7.3, sepal width (cm) = 2.9, petal length (cm)
 = 6.3, and petal width (cm) = 1.8.
To answer the following question, do not refer to the underlying mechanics of the decision
tree in any way, and only refer to the features using natural language. All the relevant
features must be mentioned in the answer, but features that were not used by the tree should
 be ignored. Moreover, do not use any technical jargon or numerical values in the response
and prefer to use terms like 'high' and 'low'.
Please explain in similar terms why the decision tree concluded that the given example is
virginica with a confidence of 97.06 %.
```

4. Experiments

In this section, we show that LLMs are able to generate appropriate explanations in natural language from decision trees. However, to obtain good explanations, the prompt must be carefully constructed, since LLMs are known to be sensitive to the input question. We performed experiments to evaluate different prompt structures until achieving the one that is presented in Section 3. We also evaluate LLM sensibility to different textual representations of the prompt components and their influence on the explanation quality.

4.1. Experiment Settings

Following the process described in Section 3, the first step is to train a decision tree given a task dataset. We used the scikit-learn toolkit [15] to train a decision tree and obtain information about feature names, class names, and an instance. The experiments have been performed to evaluate different classification datasets, although we only report results on "iris" for succinctness. It is important to note that in the scope of this paper, our focus is not on model performance but rather on the decision tree structure. Any externally trained decision tree can be used and the same process can be applied to any classification dataset as long as its textual representation is passed to the prompt construction.

Since our goal is to evaluate the capabilities of LLMs and not to compare them, we opted to focus on a single model and observe the quality of the results we can achieve. We used "gpt-3.5-turbo" in all experiments, which is a variant of the GPT-3 model [14] and currently used in the ChatGPT web application [16], through the OpenAI API. We did not perform any fine-tuning process or further modifications to the pre-trained model. No system instruction was used and all the prompt was passed as a user request. Therefore, we expect that similar results can be obtained by users without API access.

4.2. Prompt Engineering

To generate good explanations, we tested different prompt engineering strategies. Our goal was to evaluate which kind of prompt would give more adequate results.

Direct Question. The most basic way of interacting with an LLM is by simply asking a question. Our baseline approach follows this principle, in which we directly ask the model to generate a natural language explanation. In this experiment, only the dataset description, the decision tree textual representation, the instance, and the question have been passed to the prompt.

The model response: *"The decision tree concluded that the given example is virginica with a confidence of 97.06% because it followed a series of rules based on the features of the instance. First, it checked if the petal length was less than or equal to 2.45 cm, which was not the case for the given example. Then, it checked if the petal width was less than or equal to 1.55 cm, which was also not the case. Finally, it checked if the petal width was greater than 1.70 cm, which was true for the given example. Based on these rules, the decision tree determined that the given example belongs to the class virginica with a high confidence of 97.06%."*

The answer follows a sort of rule-based approach, listing for each decision node why a path was taken. This example shows that simply asking for an explanation is not sufficient to obtain a response that better helps non-expert users in comparison to the decision tree graph representation, although the reasoning was correct.

Demonstration. A well-known strategy to improve the output quality is to provide demonstrations since LLMs have been proven to be able to generate outputs that are similar to given examples [14,5]. The second experiment is based on this approach. In addition to the previous prompt components, we provide a demonstration, as shown in Table 1, and ask the LLM to generate a "similar" explanation instead.

The model response: *"The decision tree classified the given example as virginica with a confidence of 97.06% because it evaluated the feature values and found that both petal length and petal width are high. According to the decision tree path, when petal length is greater than 2.45 and petal width is greater than 1.55, the instance is classified as virginica. In this case, the high values of petal length (6.3 cm) and petal width (1.8 cm) led the decision tree to confidently classify the instance as virginica."*

The output clearly shows an explanation more suited to non-expert users, with fewer technical terms. For example, the first sentence presents a summary of the classification mentioning "high" feature values. We believe that this new explanation, which does not contain decision rules, would be more comprehensible for a user who has no familiarity with decision trees. Nonetheless, there are still numerical values present, and including the tree path adds unnecessary complexity to the answer.

Provide Instructions. Finally, following the idea of making it clear to the LLM what we want to achieve, we provide instructions after the instance description and before the question. Our goal is to make clear in the prompt what should be included in the answer and what should be avoided. As a result, we obtained the following response from the model: *"The decision tree concluded that the given example is virginica with a confidence of 97.06% because the **petal length** is high and the **petal width is also high**. This combination of high petal length and petal width led the decision tree to classify the instance as virginica with **high confidence**."*

Without mentioning the path taken, decision rules, and feature values, the current response is easier to understand. It is also clear and succinct, which makes it more suitable for non-expert users. Thus, we proceed with the experiments by modifying the text while maintaining the prompt structure.

4.3. Textual Characterization

In this section, our goal is to evaluate the LLM sensitivity to distinct textual representations of the prompt components. More specifically, we analyze how much the explanations change if we keep the same prompt structure, but modify the text input.

Incorrect Classification. The first question we would like to answer is: does the LLM recognize that a given instance was incorrectly classified? To answer this question, we change the instance prediction in the prompt to an incorrect one. We keep the same feature values shown in Table 1, but we ask for an explanation for the instance being classified as versicolor. The question is: *"Please explain in similar terms why the decision tree concluded that the given example is **versicolor** with a confidence of 97.06 %."*

The answer obtained was: *"The decision tree concluded that the given example is versicolor with a confidence of 97.06% because the petal length is high and the **petal width is low**. This combination of feature values led the tree to classify the instance as versicolor with high confidence."*

The LLM does not recognize that the instance was incorrectly classified, which indicates that the LLM is not checking the correctness of the prediction. More interestingly, the LLM changed its judgment about one feature so that it could match the given prediction, as shown in the part *"petal width is low"* which has been changed to *"petal width is high"*. Note that if the petal width was lower than 1.55, the instance would be classified as versicolor, which is the class that matches the given prediction. In a sense, the LLM changed its internal generation tendencies to agree with the given prediction.

Tree Representation. In the next experiment, we return to use the correct classification, but we change the tree representation. Instead of using the textual representation given by scikit-learn, we generate a different representation of the same tree using the ChatGPT web application [16].

We obtained the answer: *"The decision tree classified the given example as virginica with a high confidence because the petal length is high and the petal width is also high. This combination of high petal length and petal width led the decision tree to conclude that the instance belongs to the virginica class."* As we can see, the response is almost identical to the one using the previous tree text. This indicates that by using the current prompt pattern, the generated explanation is agnostic to the tree representation.

External Explanation. In order to evaluate how the demonstration affects the explanation, we alter the example provided. We replace the previous text with an example taken from López-Trigo et al. [10]. Keeping the rest of the prompt the same, we obtain the following answer: *"The decision tree concluded that the given example is virginica* *with a confidence of 97.06% because its petal width is high, which is a key factor in determining the class. Additionally, its petal length is high, which also contributes to the classification as virginica. These features align closely with the criteria used by the decision tree to classify instances as virginica."* As we can see, although there are minor differences, the generated explanation is still very similar to the previous one.

By observing that the instructions are an important factor, we decide to modify them and evaluate if the explanation changes. Therefore, we removed the last sentence of the instructions and kept the rest of the prompt the same. The answer obtained was: *"The decision tree concluded that the given example is virginica with a confidence of 97.06% because its petal width (1.8 cm) is greater than the split value (1.7 cm), and its petal length (6.3 cm) is also greater than the split value (4.9 cm). These features align with the criteria used by the decision tree to classify instances as virginica."*

The output clearly shows that the LLM tried to follow a similar pattern as the given example, using terms like "split values" as well as showing the values inside parentheses. This result indicates that although demonstrations can be used to guide the LLM, the instructions are the most influential part of the prompt.

Demonstration for Different Task. In the last experiment, we analyze the generalization capabilities of the LLM. More specifically, we verify if the LLM is able to generate a natural language explanation given a demonstration from a different task. We change the task to explain the classification of an instance from the "wine" dataset. However, we still give the LLM a demonstration from the iris dataset. This is an example of obtained output: *"The decision tree classified the given example as class_0 with 100.00% confidence because the value of color intensity is low, and the value of ash is high. This led the tree to directly classify the example as class_0 without considering any other features."*

As we can see, the LLM is able to generate a coherent explanation for the classification of the instance from the wine dataset. This is a very interesting result, as it shows that the LLM can generalize the explanation generation process to different tasks while benefiting from the demonstration given in the prompt.

5. Discussion

After performing several experiments, the results show that Large Language Models are able to explain decision trees in a way that can be understood by non-expert users. It is also clear that the structure and given instructions are more influential than the textual representations or the demonstration. Below we discuss some limitations and positive aspects from the results we obtained, as well as envision some future research directions.

5.1. On the Capabilities and Limitations of LLM Explainers

By following the pipeline described in Setion 3, we were able to obtain good explanations from decision trees. Both the prompt structure and the given instructions were influential in the quality of the responses. The best explanations clearly stated the reasons for the classification without using feature values or resorting to the tree path. The collected evidence shows that LLMs can be good explanators but they present some limitations.

For example, the responses might contain technical terms or hard-to-follow sentences, unless the LLM is directly instructed not to do so. However, if we anticipate some desired explanation, the LLM might generate appropriate outputs. Another problem is that the generated explanations are not grounded in the truth, which makes the model provide incorrect predictions. Since the LLM was not able to check correctness, it will change the answer to agree with the user, even if the path is not followed correctly.

Nevertheless, one experiment showed that the output was agnostic to the textual representations of the tree, indicating the robustness of the method. Moreover, the LLM was also able to generalize an explanation given an example from a different task. In short, based on our experiments, LLMs show capabilities that allow them to be used as explainers, such as generating text adequate for non-expert users. However, they also present several limitations, including providing incorrect explanations, which constitutes a risk to use them in sensible circumstances.

5.2. Future Research Directions

The results obtained in this work open several research directions. A natural follow-up is to better evaluate the capabilities of LLMs by performing more robust experiments. For example, the structuring of inputs and demonstrations can be compared with recent prompt engineering strategies. Another clear continuation is to expand the framework to other LLMs, such as GPT-4 [17], Llama [18,19], and Gemini [20]. This would allow us to evaluate the quality of the explanations generated by different models.

Another interesting direction is to apply the framework to other classification models, such as KNNs, SVMs, and Bayesian networks. However, in order to evaluate the results among them, it is necessary to have a formal definition of the explanation quality with the corresponding metrics to automatically perform this analysis. Although not straightforward, it would allow the comparison of completely different models and tasks.

Prior work has shown the generation of explanations for black-box models using surrogate decision trees, such as LORE [21]. Since the generation pipeline presented in this paper shows promising results in small trees, it would be interesting to evaluate how it performs in larger trees, which are more common in typical cases. This would enable the generation of natural language explanations for a wide range of tasks and the ability to provide information that is returned from local explanations, like counterfactuals [22].

Finally, the goal of working in generating natural language explanations is to make the models more transparent and understandable to non-expert users. Therefore, the next step would be to enable a bi-directional interaction channel between the user and the LLM. This would allow the user to ask questions and clarify existing doubts, and the LLM to ask for more information when the prompt is not clear enough. To achieve this goal, it is also necessary to couple the generation methods with user evaluation studies. Research on Human-Centered XAI [4,23] indicates that this is a promising direction.

Acknowledgements

This work is supported by the European Community under the Horizon 2020 programmes: ERC-2018-ADG G.A. 834756 *XAI*, by HumanE-AI-Net GA 952026, and the NextGenerationEU programme under the funding schemes PNRR-PE-AI (M4C2, investment 1.3, line on AI) FAIR (Future Artificial Intelligence Research) Prot. PE00000013. Research partially funded by project GAMING “Graph Algorithms and MinINg for Green agents”, National Recovery and Resilience Plan, NextGenerationEU. We would like to thank Noah Ziems and his co-authors for sharing their code.

References

[1] Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion. 2020;58:82-115.

[2] Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence. 2019;267:1-38.

[3] Liao QV, Varshney KR. Human-Centered Explainable AI (XAI): From Algorithms to User Experiences; 2022. Available from: https://arxiv.org/abs/2110.10790.

[4] Ehsan U, Wintersberger P, Liao QV, Watkins EA, Manger C, Daumé III H, et al. Human-Centered Explainable AI (HCXAI): Beyond Opening the Black-Box of AI. In: Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems. CHI EA '22. ACM; 2022. .

[5] Bubeck S, Chandrasekaran V, Eldan R, Gehrke J, Horvitz E, Kamar E, et al.. Sparks of Artificial General Intelligence: Early experiments with GPT-4; 2023.

[6] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A Survey of Methods for Explaining Black Box Models. ACM Comput Surv. 2018 aug;51(5).

[7] Burkart N, Huber MF. A Survey on the Explainability of Supervised Machine Learning. J Artif Int Res. 2021 may;70:245–317. Available from: https://doi.org/10.1613/jair.1.12228.

[8] Piltaver R, Luštrek M, Gams M, Martinčić-Ipšić S. What makes classification trees comprehensible? Expert Systems with Applications. 2016;62:333-46.

[9] Cambria E, Malandri L, Mercorio F, Mezzanzanica M, Nobani N. A survey on XAI and natural language explanations. Information Processing and Management. 2023;60(1):103111.

[10] López-Trigo B, M Alonso J, Bugarín A. Generación automática de explicaciones en lenguaje natural para árboles de decisión de clasificación. In: Triguero FH, Lara AT, Arroyo SD, editors. XVIII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2018); 2018. p. 481-6.

[11] Alonso JM, Bugarín A. ExpliClas: Automatic Generation of Explanations in Natural Language for Weka Classifiers. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE); 2019. p. 1-6.

[12] Gavriilidis K, Munafo A, Pang W, Hastie H. A Surrogate Model Framework for Explainable Autonomous Behaviour. In: Workshop on Explainable Robotics (ICRA); 2023. .

[13] Ziems N, Liu G, Flanagan J, Jiang M. Explaining Tree Model Decisions in Natural Language for Network Intrusion Detection. In: XAI in Action: Past, Present, and Future Applications; 2023. .

[14] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language Models are Few-Shot Learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in Neural Information Processing Systems. vol. 33. Curran Associates, Inc.; 2020. p. 1877-901.

[15] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.

[16] OpenAI. ChatGPT; 2023. Available from: https://chat.openai.com/chat.

[17] OpenAI. GPT-4 Technical Report; 2023. Available from: https://arxiv.org/abs/2303.08774.

[18] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al.. LLaMA: Open and Efficient Foundation Language Models; 2023.

[19] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al.. Llama 2: Open Foundation and Fine-Tuned Chat Models; 2023.

[20] Gemini Team, Anil R, Borgeaud S, Wu Y, Alayrac JB, Yu J, et al.. Gemini: A Family of Highly Capable Multimodal Models; 2023. Available from: https://arxiv.org/abs/2312.11805.

[21] Guidotti R, Monreale A, Ruggieri S, Pedreschi D, Turini F, Giannotti F. Local Rule-Based Explanations of Black Box Decision Systems; 2018. Available from: https://arxiv.org/abs/1805.10820.

[22] Guidotti R, Monreale A, Giannotti F, Pedreschi D, Ruggieri S, Turini F. Factual and Counterfactual Explanations for Black Box Decision Making. IEEE Intelligent Systems. 2019;34(6):14-23.

[23] Miller T. Explainable AI is Dead, Long Live Explainable AI! Hypothesis-driven Decision Support using Evaluative AI. In: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency. FAccT '23. Association for Computing Machinery; 2023. p. 333-42.