Comparative work in XAI - What defines a good explanation?

Parssa Jashnieh
Bergische Universität Wuppertal
Institute for Technologies and Management of Digital Transformation
Submitted for the seminar ”Selected Topics in Data-Science”

***Abstract*—As machine-learning architectures, such as deep neural networks, gain widespread adoption due to their high accuracy and their deployment in current software, a growing need exists for explainibility (XAI), particularly in a legal context, to ensure regulatory compliance. Although there is no general consensus on what defines a good explanation, this paper aims to answer the aforementioned question in the context of a neural network trained on the binary MNIST Dataset. To achieve this multiple explainibility methods will be compared with different inputs of the zero target class. The further analysis will then try to identify key characteristics of a good explanation to derive features that are important to the ai model, to classify zeros.**

I. Introduction

Machine learning algorithms vary in their levels of transparency with neural networks being an example of opaque ones. Although the architecture of a neural network itself is transparent, its functionality is not. It remains unclear what each neuron or a single layer does without further examination. In contrast, machine learning algorithms[1] such as k-Nearest Neighbors (kNN) and decision trees are transparent, as they are not only transparent in their architecture but also in their functionality. [1]. However, due to their high accuracy compared to transparent models like decision trees, the usage of neural networks in software, such as in the field of medicine, is likely to rise [6, p. 6].

A. How do we evaluate performance?

There exist many metrics that can be used to evaluate the performance of a MLA, such as 1) accuracy, 2) AUC, 3) F1-Score and others. Each metric has its own characteristics and assumptions, making the choice of a metric dependent on several factors, including the nature of the given dataset and the aspect of performance being measured, such as generalization ability [9]. It is challenging to determine which evaluation metric is the most widely used without extensive analysis, which could be a study on its own. However, accuracy is likely one of the most commonly employed metrics.

B. The need for XAI

The issue with focusing solely on a metric like accuracy, even when it is close to one, is that the decisions made by the model, remain opaque to the users. This opaqueness arises because deep neural networks, with their extremely high number of parameters, become less understandable as the number of hidden layers increase. The opaqueness of neural networks can lead to user mistrust, as the decisions are not directly comprehensible. Additionally, detecting biases and ensuring regulatory compliance is challenging, when only the model's output is considered [2, pp. 31, 38]. To adress this, the field of explainable AI (XAI) aims to make AI model decisions understandable to users.

A good explanation generally involves a proxy that accurately depicts the most important criteria of the model concerning the output, making them comprehensible to humans while considering the target audience [2, pp. 5, 7]. In XAI, there is a suite of methodologies, such as RISE,LIME, SHAP, CAM and many others, with some being extensions or modifications of the aforementioned methods.

C. Method selection

Post-hoc explainibility methods are model agnostic as they are not dependent on specific architectures and can be applied to different models interchangably and independet. The explanations of the post-hoc methods can generally be divided into the following categories [1].

- Explanation by simplification aims to derive a more parsimonious model from the original as a proxy
- Explanation by feature relevance provides an explanation which measures the importance of each feature with multiple inputs
- Visual explanations highlight important regions to the model graphically
- Local explanations approximate the models behaviour in specific regions and compare those regions when encountering different inputs

Because the datapoints in the binary MNIST dataset consists of images containing handdrawn numbers of zeros and ones, it makes sense to use visual explanations. Visual explanations are typically represented as heatmaps, which in the context of XAI are referred to as importance or saliency maps. It has to be noted that not every visual explanation is reliant on heatmap usage. For example many visual post-hoc explainability methods such as RISE, can also be used on a non-image input, like a text. Using a sentiment-analysis model in combination with a method like RISE, will not output a heatmap, but relevany scores of each word or token [4], which of course can be highlighted with colour. These relevany scores can be

[1] which will be referred to as MLA's in the following

interpreted as a heatmap as well, although that is rather unusal. This also shows that the XAI Terminology, is not well thought out, as these terms are used interchangably, as XAI itself, is a relatively young field [2, p. 32].

Nonetheless, the following figure shows the exemplatory generation of saliency maps using different xai-methods with the first zero found in the training dataset.

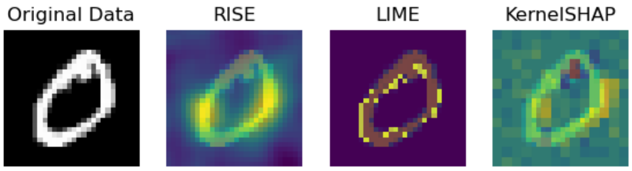

Fig. 1. Heatmaps generated with dianna ai, called importance or saliency maps, to visually accentuate regions that were most important to the model for the classification as a zero. The image on the left is the original input, while the right images depict the saliency map generated with different XAI methods. While both RISE and SHAP imply that the lateral symmetry is most important for the classification as a zero, LIME implies that its not only the lateral symmetry, but other areas and pixels as well.

II. Prerequisites

Post-hoc explainability methods such as RISE or LIME depend on AI models in order to provide proxies for the decisions made or important criteria found. This of course begs the question, for which model is opt for, in the scope of this paper, to gain further insights for the upcoming analysis.

A. The evaluated model

To generate the saliency maps, the dianna-ai package is used, which is essentially a wrapper for XAI-Methods [4]. For reasons of efficiency, dianna-ai assumes that the model on which the methods are called on, is able to work with batches. The models found in the onnx model zoo [11] were inspected with the NETRON Webpage [12]. With NETRON one is able to inspect the architecture of models saved as an onnx file. Unfortunately, the models found in the onnx model zoo do not support multiple batches, i.e multiple inputs at once, which is the reason why the model in [10] is used. It does support multiple input processing, but is trained on the binary mnist dataset instead. This is not an issue, as we are interested in the features of a good explanation in regards to the classification as a zero anyway. It is noteworthy, that this is the same model used in the dianna-ai tutorials to showcase the usage of the wrapper in general.

B. XAI methods used

There exist many XAI methods one can choose from to generate saliency maps for the MNIST dataset. Due to the limited time frame, a thorough examination of every method is not possible, rather the diannaai python package is used. Dianna itself, provides 1) RISE, 2) SHAP and 3) LIME as XAI-Methods.

III. What is a good Explanation anyway?

In the introduction, it was stated that a good explanation provides an accurate proxy of the important criteria considering the given model. Yet, this is rather the overall goal of XAI. The consideration of a good explanation depends not only on the important criteria but also on the client, i.e the subject for whom the explanation is intended for.

The aim of defining a good explanation is not a new undertaking. Disciplines such as Philosophy have long sought to establish a general definition of a good explanation [8, p. 9]. The challenge, particularly with a good explanation, is that it depends on several aspects, as noted by [3]. These aspects include dimensions of the explanation, such as whether it is complete or compact. Additional factors might include the recipients knowledge to create effective analogies and many more.

Nevertheless, the key point of this paper is not to propose a new generalized definition of a good explanation, as this would be too complex to achieve[2], given the limited time frame. Instead the focus lies on a specific target class in the MNIST dataset, aiming to derive features of a good explanation within this context.

A. How does one measure different explanations?

Figure 1 shows the generation of saliency maps using different XAI Methods, which provide different results. This proposes a new question. Which of these explanations is the best or rather how can one measure different explanations?

To answer this question different metrics are used. One of the most common metrics is the deletion metric. The main idea behind the deletion metric is to delete the k most important pixels in the saliency maps produced. By deleting pixel after pixel, the concerned model will output a new confidence. The more important the highlighted regions are, the likelier it is to observe a change in confidence. Ideally, the confidence should be lower after deleting the pixels in the predicted saliency maps. By running the model after deleting each pixel, this essentially creates a function, that can not only be plotted, but enables the calculation of the area under curve (AUC) as well. If the predicted regions are indeed important for the classification, the confidence should be lower with each deletion of a pixel. Hence, we can conclude that the best explanation with regards to the deletion metric, is the one that minimizes the auc (DAUC), for that specific input [5].

B. Why minimzing the AUC is not enough

Suppose that for that first zero in the dataset, RISE minimizes the AUC and is considered as the best explanation for that datapoint. One could not simply conclude that RISE generally performs the best, as the minimization of the AUC is input specific. It could very well be the case that for the following inputs other methods perform much better.

Because the produced saliency maps do differ significantly, as shown in Figure 1, we try to derive which method performs

[2] if even possible

the best statistically. To do so, we are interested in calculating a mean of the dauc values. This enables to us to direct our focus on the highlighted regions provided by the method which performs best on average.

IV. Methodology

Our aim is to generate the saliency maps with different XAI methods and to calculate the mean dauc-values of our samples, to direct our focus on the method, which performs the best on average. Before generating the saliency maps, we have to initialize the parameters with which the methods are called with.

A. RISE parameter initialization

RISE works by overlaying randomly generated bit masks on the original image. The overlayed images are then used as the input for the model, to generate the corresponding confidences. The key idea is that if the regions covered by the bitmasks are important, the confidence of the model should be lower significantly compared to the original confidence. The confidences are multiplied with the bitmasks, essentially overriding the ones (the overlayed regions), with the confidence of the model. The bitmasks are then summed and visualized as a heatmap [13]. But how many masks should be used to generate the saliency maps?

In the scope of this paper we took an heuristic approach to visually determine the number of sufficient masks for RISE. The different number of masks and their corresponding saliency maps are shown in the following figure.

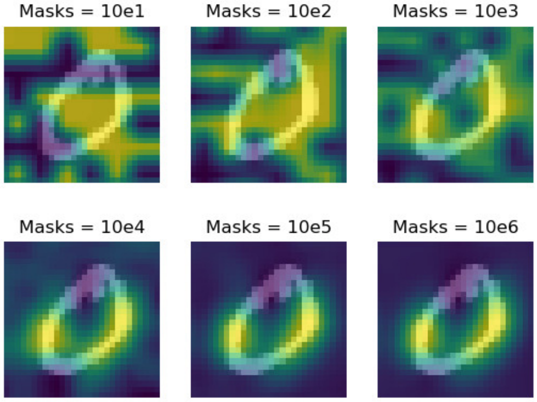

Fig. 2. Saliency maps generated with RISE, with different numbers of masks. While the saliency maps with ten to a thousand masks show a lot of noise, the saliency maps with ten thousand to a million masks, do not differ a lot visually, implying that ten thousand is a sufficiently large number for the number of masks.

As implied by Fig. 2, ten thousand seems to be a sufficiently large number for the mask generation, as the saliency maps with a hundred thousand and a million masks, do not differ much visually. It is noteworthy, that this derivation for the number of masks, implicitely assumes, that ten thousand masks is large enough for the following zeros as well. This is an assumption that we are willing to make, as comparing each and every zero in the dataset visually would be too time consuming.

B. RISE Parameter Bias

As we are not only interested in the RISE method, but the LIME and SHAP methods, we have to initialize the parameters for these as well. Both methods need to draw a number of samples in order to work. This parameter was also set to ten thousand, as this parameter worked best for RISE. One could argue that this introduces some sort of a RISE parameter bias, as we visually optimized that parameter for RISE and now use it for the other methods, but comparing the initialization of possible parameters across different methods/algorithms is a challenging task and could be a paper of its own. For example, one approach could be to construct a set or space of all parameters of interest, run the methods with each possible combination of parameters, to then find the parameters that worked best for each method. This would essentially boil down to a gridsearch. As the xai-methods are computationally heavy, running those with each and every possible combination of parameters, becomes challenging, as we would have to parallelize this gridsearch process, to even be viable. This is only one possible mechanism of comparison. Another (naive) approach could be to use the runtimes of the algorithms as proxies for the parameter initialization. Suppose for example that RISE needs ten seconds to construct a saliency map with ten thousand masks. Furthermore we assume, that LIME and SHAP take two seconds for the generation of the saliency map. We could then increase the parameters in regards to the LIME and SHAP method, to take ten seconds for the generation as well. This is a naive approach in the sense, that one has to compare different runtime complexeties. It could very well be the case, that LIME or SHAP is implemented more efficiently than RISE, meaning that by bumping up the numbers used in the parameters, could lead to an unfair advantage if we only consider the runtime. Another problem with that approach is that the runtime taken will most likely differ with each run because there will always be programs or services running in the background influencing the results. This could be mitigated by taking the averages of the runtimes with multiple runs, but shows how complex this approach can be, which is why we stuck with the ten thousand as the parameter for samples drawn. We also used different values, but as the results did not differ a lot visually, we concluded that ten thousand is a sufficiently large number for the other methods as well.

V. Results

In the previous section we mentioned how the number of sufficient parameters was derived. This section focuses on showcasing the generation of the saliency maps, as shown in Fig. 1 and calculating the corresponding dauc-values.

A. Generated saliency maps

As mentioned previously, the xai-methods provided by dianna are computationally heavy, as they are model agnostic and try to derive important features by repeatedly running the model with different inputs. As our computational ressources are limited, we were not able to generate the saliency maps with every method on every datapoint/image available. This is why we only generated the saliency maps for the first hundred training datapoints. We also could have used the test dataset, but we decided to stick with the training dataset, as we are interested to learn which features the model learned were important during training. Focusing on the testdata could have led to different results as the model is not familiar with those inputs. The generated saliency maps are available in the images/saliency_maps folder in the following github Repository [7]. In contrast to Fig. 1, they also highlight the calculated dauc-values and the corresponding datapoints with the most important regions deleted.

B. Mean dauc-values

As we are not interested in specific dauc-values, but rather the average of all dauc-values, to determine which method performed the best statistically, the (normalized) mean-dauc-values are shown in the following list. The number of k_deleted_pixels is 150.

- RISE: 0.75, standard deviation: 0.15
- LIME: 0.98, standard deviation: 0.05
- SHAP: 0.99, standard deviation: 0.0044

VI. Discussion

According to the mean dauc-values mentioned in the previous section, RISE minimizes the dauc on average, which means that we can divert our focus on the saliency maps generated by RISE.

A. Poor Performance of LIME and SHAP

Before diving into the saliency maps generated by RISE, we can take a look at the mean dauc-values provided by LIME and SHAP and explain the poor performance on the samples tested. As the maximum dauc-value only depends on the number of pixels deleted and the number of pixels deleted is 150, we can also conclude that LIME and SHAP do perform poorly, considering that they are very close to that maximum. There can be a number of reasons that lead to this behaviour. LIME for example, works by fitting an interpretable model in local regions weighted by a proximity factor, that punishes datapoints that lay far away from the original datapoint [14]. In contrast to RISE, the probabilities of the model are not as important but the classifications themselves, in order to fit the model. Because we have grayscale images consisting of 28 * 28 pixels/features, varying these pixels only in local regions most likely has not much of an influence on the overall shape of the zeros. This does not lead to a significant change in confidence, which would explain the high values observed in the average dauc-values. Similar effects could explain the poor performance on SHAP as well.

B. Which features are most important for the classification as a zero?

Essentially all saliency maps provided by RISE highlight the importance of the lateral symmetry found in the zeros. Thus, we can conclude that at least for the first hundred zeros found in the training dataset, the lateral symmetry is most important for the classification as a zero with regards to the binary mnist model used, as this method performed the best on average.

C. Different classes of zeros

When taking a look at the saliency maps provided by RISE, the lateral symmetry is most important to the zeros, however the lateral symmetry does not have to be equally important. There seem to exist zeros, for which the horizontal sides of the zero seem to be of equal importance and those for which that seems to not be the case.

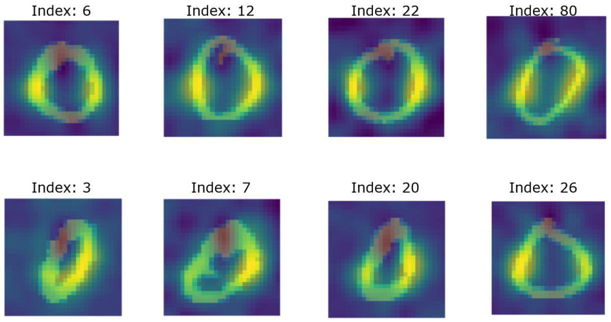

Fig. 3. Saliency maps generated with RISE. All samples imply that the lateral symmetry is the most important feature for the classification as a zero. The first row contains entries for which the lateral symmetry seems to be of equal importance, while the second row highlights a different class of zeros, indicating that the lateral symmetry does not have to be of equal importance in general.

The zeros for which the lateral symmetry is not of equal importance, could be explained with the functionality of RISE. It could be the case that both sides are not covered equally with the generated masks. To mitigate that, we saved the indices of the entries that seem to have asymmetrical lateral importance and rerun RISE with a higher number of masks (fifty thousand). The results stayed the same, further confirming that ten thousand masks are a sufficiently large number for our usecase. Nevertheless, this is not a sufficient proof. Even a higher number of masks is no guarantee that both sides are covered equally. To be certain of the existence of the other class, one has to compare the generated masks themselves to exclude the variation of the binary mask generation as a factor that could lead to asymmetrical lateral symmetry, but this would exceed the given timeframe.

D. Implicite Interpolating?

When taking a look at the saliency maps generated and the corresponding dauc-values, sometimes despite deleting whole regions from the zeros, this does not always lead to low dauc-values. This could highlight the irrelevancy of said regions, but moreover indicates that our model still had a high confidence on average on the datapoint being a zero. Is the model (implicitely) able to interpolate missing regions, or how can the high confidence despite less information content be explained? If our model would be able to interpolate missing regions, then by running the xai-methods again on the zeros where said regions are missing, we would most likely see the accentuation of the deleted regions, if indeed important. In the section ”Good Explanations are not static” in the provided notebook, a function is defined which infers the saliency map generation with RISE if the dauc-value is less than a hundred. The generated images can be found in the ”saliency_maps_for_hypothesis_testing” folder and the indices of the contained images, do match the images in regular ”saliency_maps” folder, allowing a comparison. When comparing the images, our hypothesis of implicite interpolation becomes unlikely, as the deleted regions are as important as the background, i.e not of relevancy. Comparing the generated saliency maps with RISE, we can observe that new regions are declared as important because the original regions which where declared as important originally are missing. This leads us to the conclusion that a good explanation varies based on features available. If originally important regions are missing, then the model weighs other regions differently. This essentially means that a good explanation is not static, but orients itself on features available. Despite our results showing that lateral symmetry is most important, if missing, different regions become more important to the model.

E. Further Improvements

Due to the aforementioned limited timeframe, we had to take shortcuts in some aspects to deliver our results. We would like to mention the shortcomings as these aspects could be improved, allowing for a much more detailed analysis.

- Increase the number of entries for which the saliency maps are generated
- Introduce a bias-free parameter initialization method for each xai-method used
- Not only take into account the dauc-metric, but other metrics such as a localization metric as well

VII. Conclusion

Our goal was to identify features that are important to the binary mnist model to classify zeros. In order to do that, we generated saliency maps using the xai-methods provided by dianna. The generated saliency maps were then evaluated using the deletion metric. This was done to determine which method performed the best statistically to divert the attention on the best performing one. In the case of the first hundred entries, RISE performed the best, which is why we analysed the importance maps provided by RISE. During the analysis of these maps we were able to conclude that not only is lateraly symmetry most important for the classification as a zero, i.e. features of a good explanation, but most likely two different classes of zero exist, where the lateral symmetry itself does not have to be of equal importance.

VIII. Acknowledgment

The author would like to express his gratitude towards Kienitz, Philip for the valuable feedback provided during the course of writing this manuscript.

References

[1] Plamen P. Angelov et al. “Explainable Artificial Intelligence: An Analytical Review”. In: *WIREs Data Mining and Knowledge Discovery* 11.5 (2021), e1424. ISSN: 1942-4795. DOI: 10.1002/widm.1424. (Visited on 05/26/2024).

[2] Alejandro Barredo Arrieta et al. *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. Issue: arXiv:1910.10045 _eprint: 1910.10045. Dec. 2019. (Visited on 06/06/2024).

[3] Roberto Confalonieri et al. “What Makes a Good Explanation? Cognitive Dimensions of Explaining Intelligent Machines”. In: (July 2019).

[4] *GitHub - Dianna-Ai/Dianna: Deep Insight And Neural Network Analysis*. URL: https://github.com/dianna-ai/dianna (visited on 06/06/2024).

[5] Tristan Gomez, Thomas Fréour, and Harold Mouchère. *Metrics for Saliency Map Evaluation of Deep Learning Explanation Methods*. Issue: arXiv:2201.13291 _eprint: 2201.13291. June 2022. (Visited on 05/28/2024).

[6] Mingguang He et al. “Deployment of Artificial Intelligence in Real-World Practice: Opportunity and Challenge”. In: *Asia-Pacific Journal of Ophthalmology* 9.4 (July 2020), pp. 299–307. ISSN: 21620989. DOI: 10.1097/APO.0000000000000301. (Visited on 06/06/2024).

[7] Parssa Jashnieh. *ParssaJ/Selected-Topics-Seminar*. June 2024. URL: https://github.com/ParssaJ/selected-topics-seminar (visited on 06/16/2024).

[8] Peter Lipton. “WHAT GOOD IS AN EXPLANATION?” In: *University of Cambridge* (), pp. 1–22.

[9] Hossin M and Sulaiman M.N. “A Review on Evaluation Metrics for Data Classification Evaluations”. In: *International Journal of Data Mining & Knowledge Management Process* 5.2 (Mar. 2015), pp. 01–11. ISSN: 2231007X, 22309608. DOI: 10.5121/ijdkp.2015.5201. (Visited on 06/14/2024).

[10] Christiaan Meijer and Yang Liu. *ONNX Model Trained on the Binary MNIST Dataset*. Jan. 2022. URL: https://zenodo.org/records/5907177 (visited on 06/29/2024).

[11] *Models/Validated/Vision/Classification/Mnist at Main \cdot Onnx/Models*. URL: https://github.com/onnx/models/tree/main/validated/vision/classification/mnist (visited on 06/29/2024).

[12] *Netron*. URL: https://netron.app/ (visited on 06/29/2024).

[13] Vitali Petsiuk, Abir Das, and Kate Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models*. Issue: arXiv:1806.07421 _eprint: 1806.07421. Sept. 2018. (Visited on 07/04/2024).

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. Issue: arXiv:1602.04938 _eprint: 1602.04938. Aug. 2016. (Visited on 07/04/2024).