A survey of Explainable AI and Out-of-Distribution detection methods

Jonatan Hoffmann Hanssen
Master Student, Dept. of Informatics
The Faculty of Mathematics
and Natural Sciences
Oslo, Norway
jonatahh@ifi.uio.no

***Abstract*—Machine Learning in general and Deep Learning specifically has seen an impressive rise in performance in the last decade, leading to widespread adoption for a variety of different tasks and fields. However, the use of DL methods is not without problems, especially when used in safety-critical applications. Two of these problems are the inherent inexplainability of DL models, and their inability to detect when a data point is too different from their training data and thus unlikely to be predicted correctly by the model. These shortcomings give rise to the fields of Explainable Artificial Intelligence (XAI) and Out-of-Distribution (OOD) Detection. In this essay, we shall give a short introduction to these fields and look in detail at a selection of methods from both fields.**

***Index Terms*—explainable artificial intelligence, out of distribution detection, data outlier detection**

I. Introduction

Machine Learning generally, and Deep Learning specifically, have seen a tremendous increase in performance in recent years, performing comparable to humans in many tasks, for example image classification, speech recognition and many others [Dargan et al., 2020]. In medicine, deep learning has the potential to provide faster and more accurate detection of diseases by being trained on thousands of previous patients [Nazir et al., 2023].

However, deep learning methods are not without their flaws. Firstly, deep neural networks are inherently unexplainable due to the large number of parameters that any non-trivial network has. State of the art models will perform millions of operations to evaluate a single data point, and it is therefore impossible for humans to comprehend and explain the process which lead the model to make a particular decision. In medicine, this is a major limitation of deep learning methods, as both doctors and patients expect to be able to understand why a decision was made. Secondly, although neural networks may attain high accuracy on test data and appear to have learned great insights about the tasks they are employed in, they often lack robustness and can suffer large drops in performance on data points which are slightly different from the training data. As [Szegedy et al., 2014] has shown, it is possible to create data points which are imperceptibly different from normal data points, yet still fool otherwise high performing models.

These two problems lead to the two fields of machine learning which we shall discuss in this essay. These are Explainable Artificial Intelligence (XAI), and Out-of-Distribution (OOD) detection.

A. Terminology

Before proceeding, this small section will go through the choice of terminology used in this paper.

Explainability and *interpretability* are sometimes used to mean slightly different things, and some may place a distinction between *interpretable* or *explainable* machine learning. In this paper, we shall use the term *Explainable AI* broadly, and will not make this distinction.

We assume the reader is familiar with both machine learning in general and deep learning specifically. Whenever we use the term *model*, we typically mean a neural network, or other ML models such as Decision Trees or Support Vector Machines. When we use the term *network*, we exclusively mean a neural network, and not any other ML model.

II. Explainable AI

Below follows a thourough introduction to XAI, as well as detailed look at some important methods for explainability for neural networks applied to images.

A. The motivation for XAI

Given the impressive performance of DL methods, one might be convinced that these models do not need to be explainable or interpretable, and that we instead should just place our faith in the model without knowing exactly how it came to a decision. However, as [Doshi-Velez and Kim, 2017] points out, "a single metric, such as classification accuracy, is an incomplete description of most real-world tasks". Small differences between the data distribution when the test data was collected and when the model is deployed may have a large impact on the model's performance, or the model may have learned artifacts or specificities in the training dataset which were also present in the test dataset, leading to a false belief that the model has gained generalizable knowledge when it has not. By using explainable methods, we may reveal these shortcomings.

XAI is also important whenever the model is used in settings where its decisions have a high impact. If a model is used by a hospital in disease detection, both the patient and doctor will

probably want to be able to understand why the model has found that a disease is present. For them, high performance on a test set of different cases may not be enough. As [Nazir et al., 2023] states, "for the regulated healthcare domain, it is utmost important to comprehend, justify, and explain the AI model predictions for a wider adoption of automated diagnosis". In other high impact areas, such as autonomous driving, the impact of wrong decisions by the network can have fatal consequences, and customers and regulators will want to be absolutely sure that the models used are robust and base their decisions on relevant factors as opposed to quirks in the training data. Furthermore, the right to an explanation of an automated decision affecting a person is included in the EU's General Data Protection Regulation, which states that "In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right [...] to obtain an explanation of the decision reached after such assessment and to challenge the decision." [European Union, 2016].

B. Taxonomy of XAI

This sections goes through three axes which define an XAI method:

- Intrinsically explainable models versus post hoc methods
- Model dependence versus model agnostic methods
- Global versus local explanations

1) Intrinsically explainable models versus post hoc methods: Intrinsically explainable models are models which have sufficiently low complexity, such that it is feasible for a human to understand them without further modifications. Examples of such methods are linear regression, logistic regression and decision trees [Molnar, 2022].

Post hoc methods are methods which are applied to the model after training. These methods do not aim to constrain the model to be interpretable, but inspect the model after training.

2) Model dependence versus model agnostic methods: Model dependence/agnosticity denotes whether an XAI method uses specifics of a particular type of model to generate the explanation, or whether the method can generate an explanation without using specifics of the model at all. Explanations based intrinsically explainable models are clearly model dependent, while methods that only use the input and output of the model instead of looking at the internal operations are model agnostic. An example of a model dependent method (which is not simply an intrinsically explainable model) is Class Activation Mapping, which requires a CNN with a specific architecture to function, while an example of a model agnostic method is Shapley values, which uses the inputs and outputs to calculate the marginal effect of a single feature on the output value.

3) Global versus local explanations: Global explanations provide general relationships between the input features and outputs learned by the model over the entire dataset [van der Velden et al., 2022]. In this way, they can show how a specific feature affects the output in general, instead of just how it affects the output of a single point. These methods are ideal for finding trends in the data, but may not be suitable for a patient wanting an explanation for their specific case.

Local explanations do not describe general trends, but focus only on a single data point. These methods give insight into how the features influenced the prediction of a single data point, but these relationships may not hold for other data points, and as such these methods do not give the same insight into the general behaviour of the model.

C. Specific methods

The following section goes through several specific XAI methods, specifically ones related to images.

1) Class Activation Mapping (CAM): CAM [Zhou et al., 2015] is a model dependent, post hoc XAI method, which is used on Convolutional Neural Nets (CNNs). For a specific output node of a model (for example, the one denoting the presence of a specific class, such as "cat"), CAM outputs a heat map showing which areas of the input image contributed to this node. In this way, CAM gives a visual explanation to which parts of an image the model focused on when making a decision to classify an image to a specific class. This method is model dependent, because it requires a specific architecture in the final layers of the network to work.

CAM is a relatively simple method to understand. It exploits the fact that various convolutional layers of CNNs actually behave as object detectors, even when the training objective is classification [Zhou et al., 2015]. As [Lecun et al., 1998] explains, the earlier layers "extract elementary visual features such as oriented edges, end-points [or] corners", which can be used by subsequent layers to detect higher-order features. In this manner, the final convolutional layer will detect very high level visual features, combining the extracted information from all the previous layers. This layer is composed of several feature maps, where each map can be thought of as denoting the presence of some specific feature across the original image. The authors perform global average pooling (GAP) on these feature maps, giving a single value for each map, which is followed by a single dense layer and the Softmax activation function. In this way, each output node in the final layer is a weighted sum of all the global average pooled feature maps from the final convolutional layer. This means that we can represent the areas of the image which were used to perform the classification by performing the same weighted sum of the actual feature maps instead, which gives us a heat map which we can overlay on the original image (after upsampling the feature maps).

Figure 1 shows the process visually. From this we can see that the resulting Class Activation Map (bottom right) gives an intuitive explanation for why the image in the top left gives a high score for the presence of the class "Australian Terrier".

Although CAM is an intuitive and effective method of visualizing the inner workings of a CNN, it has some downsides. Firstly, it is highly model dependent, requiring that the model only have a single dense layer after the

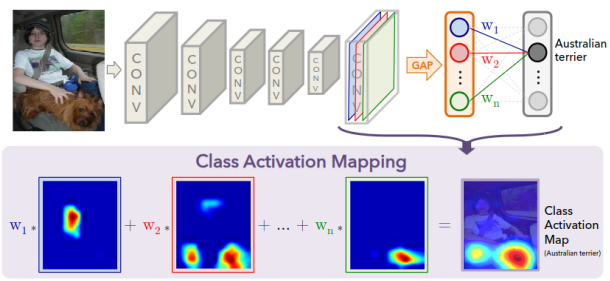

Fig. 1. Figure taken from [Zhou et al., 2015], showing the steps required to create a Class Activation Map

convolutions. Although there are some state of the art models which only use a single dense layer, this still places a limit on what models can be used, or requires the simplification of models that use more than a single dense layer. [Zhou et al., 2015, 4] notes a 1-2% drop in classification performance when performing this simplification. Secondly, the output of CAM is simply a weighted sum of all the feature maps after the final convolutional layer. As we move deeper in a CNN, we reduce the spatial resolution by downsampling, while increasing the number of channels (increasing the depth of the output while reducing the height and width). Because of this, the CAM will have a drastically lower resolution than the original image, often less than $10x10$ while the input image may be hundreds of pixels in both dimensions. Because of this, CAM can only show general areas, as opposed to pixel wise explanations.

2) Gradient Class Activation Mapping (Grad-CAM): Grad-CAM [Selvaraju et al., 2019] is an improvement on CAM, which generalizes the method to function with any CNN architecture, thus making the method much less model dependent and avoiding the performance drop incurred when simplifying the model with CAM. Instead of using the weights of a final layer to calculate a weighted sum of feature maps in the last convolutional layer, Grad-CAM uses gradients flowing from the relevant output node to the activation maps to calculate the weights for each feature map. Furthermore, the authors prove that this method is a strict generalization of CAM [Selvaraju et al., 2019, 5], so that no information is lost by using gradients instead of weights.

Like the simplicity of the CAM method, the calculation of the weights using the gradients is also quite simple, as seen in Equation 1.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\delta y^c}{\delta A_{ij}^k} \tag{1}$$

Here, c represents the index of the class we are interested in, k the index of the feature map, and i and j the width and height of the image. y^c is the element of the output vector y which corresponds to the class c, while A^k is the k'th feature map. Z is equal to $i * j$, and simply normalizes the sum. Thus, we are actually just performing global average pooling of the gradients of A^k with respect to y^c, which gives us a single value we can use as the weight for this feature map. Doing this for all feature maps for a specific class gives us all the weights we need to calculate a weighted sum, which we can upsample and visualize to get an explanation for the decision of the CNN.

3) Guided Backpropagation:

4) Layer-Wise Relevance Propagation (LRP): LRP is another XAI method which generates a visual explanation of the areas of an image which lead to a classification decision. Unlike CAM and Grad-CAM, LRP outputs a map which describes the relevance of every single pixel in the input image, and is thus produces a much more fine grained explanation than these other methods. LRP also differs in that [Bach et al., 2015] does not define it as a specific method, but rather as a concept defined by a certain set of constraints which can be satisfied by different implementations depending on the type of model.

LRP assumes that we can model the relevance R_i for any node i in a neural network, and aims to find the relevance for all the input nodes (the pixels in an image). Relevance is the contribution of any node to the final prediction $f(x)$ of a network, and the idea is to take the relevance of the output layer (simply defined as the output $f(x)$), and iteratively propagate this backwards through the network. Relevance scores are subject to a conservation property, which means that the sum of relevances must be equal for all layers (Equation 2). Furthermore, nodes must also conserve relevance, such that the sum of relevances a node receives from the previous layer is equal to the amount it distributes to the next layer. Once the relevance scores have been propagated from the output to the input layer in accordance with these constraints, we have a measure for how each input pixel contributed to the final output.

$$f(x) = ... = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l+1} R_d^{(l+1)} = ... = \sum_d R_d^{(1)} \tag{2}$$

To distribute relevance between the nodes in a way which obeys the constraints defined in Equation 2, a rule for propagation of relevances must be defined. As [Bach et al., 2015] shows, simply satisfying the constraints is not guaranteed to lead to meaningful explanations, nor is the decomposition of relevance unique. However, they show that by using a suitable propagation rule, we gain a visual explanation which shows which areas contribute to the final decision and which areas make the final decision less likely [Bach et al., 2015, 28].

III. Out-of-Distribution (OOD) Detection

This section discusses OOD detection, the field which attempts to tackle the second problem discussed in the introduction; that ML models have significantly worse performance on OOD data points and will often "fail silently", making completely wrong predictions with apparent high confidence [Goodfellow et al., 2015]. OOD detection is a developing field,

and still in an initial stage [Cui and Wang, 2022]. In 2017, [Hendrycks and Gimpel, 2018] proposed a baseline OOD detection method. This section will discuss this method and the methods which follow it.

A. Motivation for OOD Detection

When training a model using supervised learning, we implicitly use the ”closed world assumption”, which means that we assume that test data will be drawn from the same distribution as the training data [Yang et al., 2024]. However, when a model is deployed, the data we see may not obey this assumption. Without OOD detection, the model will behave in the exact same way when encountering OOD samples or in distribution (ID) samples, and may even claim to be highly confident in its prediction although the sample is far away from the distribution of the training data [Liu et al., 2021, 1]. In any system where models make high impact decisions, this is a huge problem. We do not want a model to claim high confidence when predicting if a woman has lung cancer, if the model has only been trained on men. Thus, OOD detection methods are necessary, so that OOD samples can be caught before the model makes a prediction and dealt with correctly.

Intuitively, one might assume that distinguishing ID and OOD samples from each other can be solved by simple binary classification using a dataset of ID samples and one of OOD samples. Indeed, if one has sufficient amount of high quality OOD samples, this can be done. However, this can be difficult to obtain in practice [Yang et al., 2024, 15], thus requiring more sophisticated methods of OOD detection.

B. Semantic versus covariate shift

The first distinction to make in OOD detection tasks is whether an OOD sample is OOD because of *semantic* or *covariate* shift. Semantic shift refers to samples with different classes than the ones the model is trained on. A picture of a giraffe would represent a semantic shift for a model trained to differentiate between cats and dogs, as a giraffe does not belong to either the ”dog” or ”cat” class. Covariate shift refers to samples which come from a different distribution while still belonging to one of the classes of the original data set. A picture of a chihuahua could represent a covariate shift for the same cat-versus-dog model if the training data contained only other races of dogs. The detection of semantic shift, as opposed to covariate shift, is the main focus of most OOD detection tasks [Yang et al., 2024, 5]. In many applications, it is expected that the model should be able to generalize its prediction to covariate-shifted data, and therefore the focus is on detecting semantic shift [Yang et al., 2024, 5]. However, the field of medical image classification is one where detecting covariate shift is also important, as the model should only make predictions on data points which are very similar to its training data [Yang et al., 2024, 5].

C. Methods

This section will follow the same outline as section II; firstly, the overarching categories of methods will be discussed, followed by a more detailed look at a selection of specific methods within the field.

The field of OOD is separated into four categories of methods [Yang et al., 2024]:

- Classification-based methods
- Density-based methods
- Distance-based methods
- Reconstruction-based methods

Below follows a short explanation of each of the methods.

1) Classification-based methods: Classification-based methods usually use the softmax score or logits of a model to attempt to distinguish OOD and ID samples. [Hendrycks and Gimpel, 2018] made the observation that while the softmax score may be a poor indication of the actual confidence of the model on a single data point, it is still higher on average for ID samples as opposed to OOD samples. By using this simple distinction, they created a baseline model which separated OOD and ID samples. Using input perturbations and temperature scaling, [Liang et al., 2020] further improved on this method, by amplifying the difference in softmax score of ID and OOD data. There are several other state of the art methods which utilize a classification-based approach, and these make up a large part of the representative methodologies for OOD detection today [Yang et al., 2024, 8]. As such, we shall devote the majority of section III-D to classification-based methods.

2) Density-based methods: Density-based methods explicitly try to model the in-distribution [Yang et al., 2024], which is then used to detect outliers in low likelihood regions. Although the idea is intuitive, learning the distribution of the data set can often be prohibitively expensive, and thus these methods often lag behind classification-based methods [Yang et al., 2024].

3) Distance-based methods: Distance-based methods attempt to detect OOD samples by calculating their distance to ID samples. Many different distance measures are used, such as Mahalanobis distance to estimated Gaussian distributions, cosine distance to the first singular vector of the data set or Euclidean distance in an embedding space.

4) Reconstruction-based methods: Reconstruction-based methods are based on encoder-decoder frameworks, where the core idea is that the model will be much worse at reconstructing OOD data than ID. By measuring the reconstruction loss, we can detect OOD samples.

D. Specific methods

Below follows a more detailed look a selection of specific OOD detection methods.

1) Baseline model: The baseline model created by [Hendrycks and Gimpel, 2018] is extremely simple, yet effective. It simply compares the softmax score the predicted class to a threshold, and labels it as OOD if it falls below this threshold. This works reasonably well, because the softmax scores for ID data generally is higher than for OOD data. However, such as simple method has its shortcomings, and

there are many ways to improve the method, as will be shown in the sections that follow.

2) Out-of-Distribution Detector for Neural Networks (ODIN): [Liang et al., 2020] improves on the work of [Hendrycks and Gimpel, 2018] by introducing two simple modifications to the method which amplify the difference between the softmax score of ID and OOD samples. Firstly, they alter the input image slightly by adding small perturbations based on the gradients of the cross-entropy loss, as shown in equation 3

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \epsilon \,\mathrm{sign}(-\nabla_{\boldsymbol{x}} \log_{\hat{y}}(\boldsymbol{x}; T)) \tag{3}$$

Secondly, they add temperature scaling to the softmax calculation, as shown in equation 4:

$$S_i(\tilde{\boldsymbol{x}}; T) = \frac{\exp\left(f_i(\tilde{\boldsymbol{x}})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\tilde{\boldsymbol{x}})/T\right)} \tag{4}$$

Thus, the OOD detector has the following form, given a threshold δ:

$$g(\boldsymbol{x}; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if } \max_i S(\tilde{\boldsymbol{x}}; T) \leq \delta, \\ 0 & \text{if } \max_i S(\tilde{\boldsymbol{x}}; T) > \delta. \end{cases} \tag{5}$$

With these modifications, they report large improvements over the baseline [Liang et al., 2020, 4]. To explain this increase, we should look at the mathematical justification for these modifications.

The idea behind perturbing the input image based on the gradient of the cross entropy loss is that ID data points have a higher gradient than OOD data in general. By moving our data point slightly in the direction of the negative gradient, we should expect to see a higher softmax score than if we did not move, regardless of whether the data point is ID or OOD. However, because the gradients are larger for ID data, we expect that the difference between the new softmax scores will be larger than they were before the perturbations, because the ID data point has moved further towards higher softmax values, as shown in 2, taken from [Liang et al., 2020, 8]. Here we see two data points, one ID (red) and one OD (blue), which are both perturbed. As we can see, the resulting softmax scores after the perturbations differ more than before the perturbation.

The interpretation of the temperature scaling is slightly more complex. By performing a Taylor expansion and omitting third and higher orders, we can rewrite the softmax function as $S \propto (U_1 - U_2/2T)/T$, with U_1 and U_2 defined as follows [Liang et al., 2020, 4]:

$$U_1(\boldsymbol{x}) = \frac{1}{N-1} \sum_{i \neq \hat{y}} [f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})] \tag{6}$$

$$U_2(\boldsymbol{x}) = \frac{1}{N-1} \sum_{i \neq \hat{y}} [f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]^2 \tag{7}$$

3) Virtual Logit Matching (ViM):

4) Virtual Outlier Synthesis (VOS):

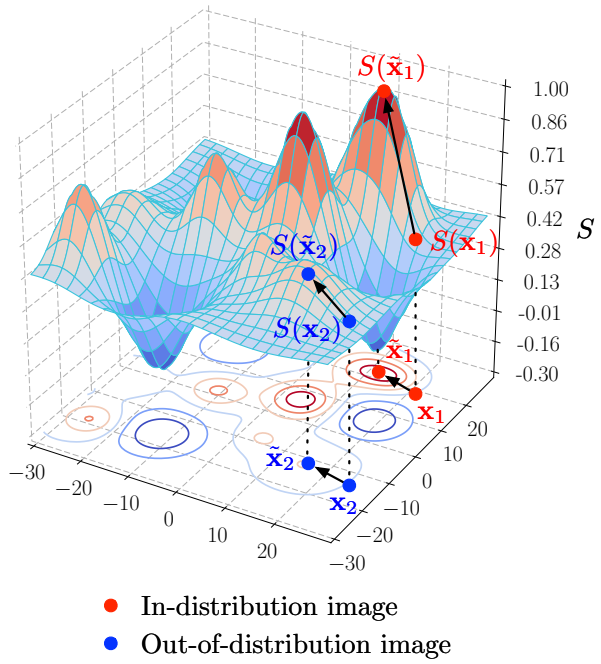

Fig. 2. Figure taken from [Liang et al., 2020], showing the difference in gradients between ID and OOD data points

References

[Bach et al., 2015] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

[Cui and Wang, 2022] Cui, P. and Wang, J. (2022). Out-of-distribution (ood) detection based on deep learning: A review. *Electronics*, 11(21).

[Dargan et al., 2020] Dargan, S., Kumar, M., Ayyagari, M. R., and Kumar, G. (2020). A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092.

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

[European Union, 2016] European Union (2016). Article 71: European data protection board. Accessed: February 13, 2024.

[Goodfellow et al., 2015] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples.

[Hendrycks and Gimpel, 2018] Hendrycks, D. and Gimpel, K. (2018). A baseline for detecting misclassified and out-of-distribution examples in neural networks.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324.

[Liang et al., 2020] Liang, S., Li, Y., and Srikant, R. (2020). Enhancing the reliability of out-of-distribution image detection in neural networks.

[Liu et al., 2021] Liu, W., Wang, X., Owens, J. D., and Li, Y. (2021). Energy-based out-of-distribution detection.

[Molnar, 2022] Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.

[Nazir et al., 2023] Nazir, S., Dickson, D. M., and Akram, M. U. (2023). Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in biology and medicine*, 156:106668.

[Selvaraju et al., 2019] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

[Szegedy et al., 2014] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks.

[van der Velden et al., 2022] van der Velden, B. H., Kuijf, H. J., Gilhuijs, K. G., and Viergever, M. A. (2022). Explainable artificial intelligence (xai) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470.

[Yang et al., 2024] Yang, J., Zhou, K., Li, Y., and Liu, Z. (2024). Generalized out-of-distribution detection: A survey.

[Zhou et al., 2015] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2015). Learning deep features for discriminative localization.