Annals of Mathematics and Computer Science ISSN: 2789-7206
Vol 29 (2025) 104-111 https://doi.org/10.56947/amcs.v29.635

FEATURE ATTRIBUTION METHODS IN MACHINE LEARNING: A STATE-OF-THE-ART REVIEW

SAIDJON KAMOLOV[1*]

Abstract. This paper presents a state-of-the-art survey of feature attribution techniques employed in explainable AI. We organize the existing literature into a proposed taxonomy of model-agnostic and model-specific approaches. We analyze the formal definitions, mathematical formulations, usage contexts, strengths, and limitations of these methods. A comparative analysis highlights key trade-offs concerning model agnosticism, explanation form, computational cost, and fidelity to the model. We find that while model-agnostic techniques offer broad applicability by treating models as oracles, often at a higher computational cost, model-specific methods leverage internal model architecture or gradients for potentially more efficient and faithful explanations, albeit with reduced generality.

Keywords. feature attribution, feature selection, explainable AI, deep learning
2020 Mathematics Subject Classification. Primary 68T01; Secondary 68T07

1. Introduction

Over the last decade, explainable AI (XAI) has increasingly focused on methods that shed light on the inner workings of complex "black-box" machine learning models [1]. One of the most influential directions within XAI is feature attribution, which assigns an importance score to each input feature for a specific prediction [2]. The principal aim of these approaches is to clarify how much each feature has contributed to the final model output for a given instance. Most feature attribution methods are post-hoc (applied after model training) and local (targeting individual predictions), and they typically produce feature-based explanatory summaries (often visualized for user inspection) [3].

Feature attribution techniques can be divided into two broad categories: model-agnostic approaches, which treat the model purely as an oracle that can be queried for predictions, and model-specific approaches, which exploit internal gradients or model architectures [2]. Examples of model-agnostic methods include LIME and SHAP, while deep learning-specific methods encompass saliency maps, Integrated Gradients, DeepLIFT, Layer-Wise Relevance Propagation (LRP), Grad-CAM,

Date: Received: Jun 28, 2025; Accepted: Aug 12, 2025.
* Corresponding author

and attention-based attribution. The discussion below proposes a taxonomy that groups these techniques according to their methodological underpinnings and then analyzes their formal definitions, mathematical formulations, and usage contexts in order to highlight their strengths and limitations.

2. Proposed Taxonomy of Feature Attribution Methods

This review organizes feature attribution methods under two main headings: model-agnostic approaches and model-specific approaches. The former typically rely on input–output behavior alone (e.g., perturbing inputs, fitting surrogate models, or enumerating subsets), whereas the latter leverage network gradients, reference-based comparisons, or specialized architectural structures (often in deep neural networks). Model-agnostic methods are subdivided into local surrogate-based explanations (exemplified by LIME) and Shapley-value-based explanations (exemplified by SHAP). Model-specific methods are further divided into gradient-based saliency maps, reference-based backpropagation methods (e.g., Integrated Gradients and DeepLIFT), Layer-Wise Relevance Propagation, Grad-CAM, and attention-based explanations.

3. Model-Agnostic Methods

3.1. **LIME: Local Interpretable Model-Agnostic Explanations.**

One of the best-known model-agnostic methods is LIME [4]. It explains individual predictions by approximating the local decision boundary of a complex classifier or regressor with an interpretable surrogate model. Formally, for a black-box model f and an instance x, LIME fits a simpler model $g \in G$ (often a sparse linear model) that optimizes a loss function $L(f, g, \pi_x)$ plus a complexity penalty $\Omega(g)$. The function π_x defines a locality measure around x. This optimization can be written:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g).$$

In practice, LIME proceeds by randomly perturbing the features of x to generate a synthetic neighborhood of samples, querying the black-box model f to obtain labels or predictions for these perturbed samples, and weighting them by their similarity to x. A sparse linear model is then trained on this neighborhood in order to mimic f locally. The surrogate model's coefficients provide an approximation of each feature's importance for the prediction in question.

LIME offers significant advantages, most notably its flexibility: it can be applied to tabular data, text data (switching words on or off), and images (perturbing superpixels). It is also considered highly interpretable for end users, as the surrogate explanation typically highlights only a small number of high-impact features [3]. However, this approach can suffer from instability due to the random sampling process, and its accuracy depends on the assumption that the decision boundary is reasonably linear in the vicinity of the instance [3]. Furthermore, the choice of sampling parameters (e.g., kernel width, number of samples) can greatly affect the resulting explanation, and in highly non-linear scenarios, the linear approximation may fail to capture the nuances of the true boundary. LIME also incurs a potentially large computational overhead because it requires many

evaluations of the model for each local explanation, which becomes expensive for complex architectures or large datasets.

3.2. **SHAP: Shapley Additive Explanations.** Another well-known model-agnostic framework is SHAP [5], which unifies several existing feature-attribution methods under the theoretical umbrella of Shapley values from cooperative game theory. Shapley values provide a principled way to distribute a "payout" (in this case, the model's output) among all features by averaging each feature's marginal contribution over all possible subsets. SHAP frames the explanation as an additive model

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

where z' is a binary vector indicating the presence or absence of features, and ϕ_i is the Shapley value associated with feature i. The Shapley value for feature i is

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\ (M - |S| - 1)!}{M!} \Big(f_x(S \cup \{i\}) - f_x(S) \Big),$$

where N is the full set of features and $f_x(\cdot)$ indicates the model's output when only the specified subset of features is used. In principle, Shapley values guarantee properties such as local accuracy, missingness, and consistency [3].

Several specialized SHAP implementations have been proposed. KernelSHAP is model-agnostic and uses sampling and local regression (somewhat analogous to LIME's surrogate approach). TreeSHAP provides exact computations for tree-based models, while DeepSHAP, GradientSHAP, and LinearSHAP are adaptations for deep networks, gradient-based models, and linear models, respectively [3]. Although SHAP is theoretically appealing due to its firm grounding in game theory, its exact computation can be prohibitively expensive for high-dimensional data, and even the approximate variants (e.g., KernelSHAP) often demand numerous model evaluations. Moreover, KernelSHAP may generate unrealistic feature combinations, potentially leading to misleading interpretations, and Shapley values are known to be sensitive in the presence of highly correlated features. Despite these challenges, SHAP remains widely used and is supported by well-developed libraries, making it a prevalent choice for model-agnostic local explanations.

4. Model-Specific Methods

4.1. **Gradient-Based Saliency Maps.** Saliency maps, first introduced by Simonyan et al. (2014) for explaining CNN-based image classifiers, rely on computing the gradient of a specific output class score with respect to the input features [6]. Let $S_c(x)$ be the score for class c and x_i be the i-th feature. The saliency map at feature x_i is given by

$$R_i = \frac{\partial S_c(x)}{\partial x_i}.$$

A large gradient magnitude indicates that small changes in x_i substantially affect the class score, implying that x_i is important. This approach can be visualized as

a heatmap over image pixels or, in text tasks, can be projected onto embeddings to highlight salient tokens [6].

Gradient-based saliency is conceptually simple and computationally fast, typically requiring only one backward pass. However, it reflects only local sensitivity and can be highly noisy. Models that saturate in certain regions may report negligible gradients even for important features [6]. Numerous refinements have been proposed, including SmoothGrad [7], which reduces noise by averaging gradients over several noisy input samples, and Guided Backpropagation [8], which modifies the backward pass through ReLU layers to isolate positively contributing features. Despite these enhancements, vanilla saliency maps do not guarantee completeness and may fail to capture global interactions among features.

4.2. **Reference-Based Backpropagation Methods: Integrated Gradients and DeepLIFT.**

4.2.1. *Integrated Gradients.* Integrated Gradients (IG) [9] addresses some of the limitations of plain gradients by integrating the gradient along a path from a baseline input $x^{(0)}$ to the actual input x. For a model F and feature i, IG is defined as

$$\mathrm{IG}_i(x) = (x_i - x_i^{(0)}) \times \int_{\alpha=0}^{1} \frac{\partial F\big(x^{(0)} + \alpha(x - x^{(0)})\big)}{\partial x_i}\, d\alpha.$$

This integral is approximated numerically (e.g., via a Riemann sum with 50–300 steps). The method satisfies certain axioms such as completeness, meaning that the sum of all feature attributions equals $F(x) - F(x^{(0)})$ [9]. It is more robust than vanilla saliency because it considers gradients along the entire path. IG does, however, depend on the choice of baseline, and an inappropriate baseline can result in attributions that do not reflect realistic feature contributions.

4.2.2. *DeepLIFT.* DeepLIFT [10] also compares the activation of each neuron to that neuron's activation at a reference input. Instead of integrating gradients, DeepLIFT systematically decomposes the difference in the output $F(x) - F(x^{(0)})$ into contributions from each input such that

$$\sum_i C_i = F(x) - F(x^{(0)}).$$

It uses rules like the "Rescale" or "RevealCancel" rule to handle non-linearities and propagate contributions backward through the network. Like IG, DeepLIFT is fast and reference-based, but it is not implementation-invariant, which can lead to different attributions for mathematically equivalent network structures. Despite this drawback, it has proven highly valuable in practice due to its efficiency and interpretive fidelity, particularly in application areas like genomics [10].

4.3. **Layer-Wise Relevance Propagation (LRP).**

Layer-Wise Relevance Propagation (LRP) [11] was designed to trace the network's output back to the input

features by distributing the "relevance" at each layer to its preceding layer. For a neuron k receiving inputs from neurons j, the relevance of neuron j is:

$$R_j = \sum_k \frac{z_{jk}}{\sum_{j'} z_{j'k} + \epsilon} R_k,$$

where $z_{jk} = x_j w_{jk}$, R_k is the relevance at neuron k, and ϵ is a small stabilizer. LRP preserves the total relevance across layers and has shown promise in highlighting which pixels of an image are most responsible for a certain classification [12]. Nevertheless, the procedure depends on particular propagation rules (e.g., α-β rules), and different configurations or equivalent network reformulations can lead to different heatmaps.

4.4. **Grad-CAM: Gradient-Weighted Class Activation Mapping.** For image classification in particular, Grad-CAM [13] produces spatial maps that localize the regions responsible for a given class score. It does so by computing the gradient of the class score with respect to a chosen convolutional layer's feature maps. The resulting gradients are globally averaged, yielding weights α_k for each feature map A^k_{ij}. Grad-CAM then produces a weighted combination of these maps, followed by a ReLU operation:

$$L^c_{\text{Grad-CAM}}(i,j) = \text{ReLU}\Big(\sum_k \alpha_k A^k_{ij}\Big).$$

This yields a coarse heatmap indicating where the model "looks" in deciding on class c. It is widely used in computer vision due to its intuitive visual nature, though its resolution is constrained by the size of the convolutional layer, and it typically only indicates positively contributing factors. Moreover, it remains restricted to CNN-based architectures rather than serving as a general-purpose black-box explanation tool.

4.5. **Attention-Based Explanations.** Modern neural models, especially in NLP, frequently employ attention mechanisms that assign weights to tokens or image regions. It is natural to interpret these attention weights as measures of feature importance. Attention-based explanations can be highly appealing, since no additional computations are necessary and the resulting weight distributions can be straightforwardly visualized. However, the well-known perspective "Attention Is Not Explanation" [14] emphasizes that attention weights alone may not consistently reflect true feature importance, since the same model outputs can sometimes be produced by different attention configurations. Subsequent work has argued that "Attention Is Not Not Explanation," suggesting that attention can be a useful proxy under certain conditions, but is best validated or combined with more rigorous attribution methods [15]. Consequently, attention-based explanations are often treated as an exploratory tool rather than a definitive account of the model's behavior.

5. Comparative Analysis and Discussion

Table 1 compares these methods according to their model-agnosticism, explanation form, computational requirements, and fidelity. While the table is a high-level overview, it highlights key trade-offs in explainability research.

Table 1. Comparison of feature attribution methods across key dimensions.

Method	Model-Agnostic?	Explanation Form	Comp Cost	Fidelity to Model
LIME	Yes	Local surrogate model (linear)	Medium	Moderate: local fidelity depends on parameters [3]
SHAP	Yes	Shapley values	High	High: strong theoretical guarantees [3]
Saliency (Gradient)	No	Gradients as importance scores	Low	Low: noisy, local sensitivity [6]
Integrated Gradients	No	Path-integrated gradients	Medium	High: satisfies completeness [9]
DeepLIFT	No	Difference-from-baseline attribution	Low	High: decomposes output difference [10]
LRP	No	Layer-wise relevance heatmap	Low	High: relevance conservation [12]
Grad-CAM	No (CNN-specific)	Visual heatmap of spatial regions	Low	Moderate: coarse localization, positive-only [13]
Attention Weights	No (Transformer-specific)	Weight distribution as importance	Very Low	Low: not reliably faithful [14]

The main dimensions that differentiate these approaches include the interpretability of their explanations, whether or not they are model-agnostic, the computational burden, and their fidelity (i.e., how well they truly align with the model's internal logic). For tabular data, additive methods such as LIME or SHAP tend to produce more directly interpretable outputs. In computer vision, gradient-based methods, Layer-Wise Relevance Propagation, or Grad-CAM often yield heatmaps that localize the regions the model deems most important. In NLP, attention-based attributions may be natural to visualize but might not always capture genuine feature importance; gradient-based or reference-based techniques can be more faithful [9, 14].

Model-agnostic methods are generally more flexible, but may involve considerable sampling and approximations that can lead to high computational costs or instability [4, 5]. By contrast, model-specific methods often benefit from direct access to gradients or architectural features, producing more faithful and efficient explanations at the cost of limited generalizability.

6. Conclusion

Feature attribution methods have become indispensable for understanding machine learning models, especially as these models are deployed in sensitive domains requiring transparent decision-making. Model-agnostic approaches such as LIME [4] and SHAP [5] laid the groundwork for universal local explanations. By only requiring access to a model's inputs and outputs, these techniques can be applied to virtually any learning algorithm, although they must often resort to computationally intensive sampling and surrogate modeling. In parallel, deep learning–specific methods like saliency maps, Integrated Gradients [9], DeepLIFT [10], Layer-Wise Relevance Propagation [11], and Grad-CAM [13] have leveraged neural network gradients and architectures to produce efficient, high-fidelity attributions. Attention-based explanations [14] add yet another lens for interpretability, but they must be handled with caution since attention weights may not always correspond to genuine feature importance.

In practice, researchers and practitioners should choose attribution methods according to the domain, data modality, computational constraints, and fidelity requirements. In high-stakes settings, it is often advisable to compare results from multiple attribution methods to ensure reliability. There is an inherent trade-off between simplicity, faithfulness, and scalability that remains at the center of XAI research. Future work is expected to focus on unifying local attributions with higher-level conceptual explanations, designing robust metrics to evaluate the quality of explanations, and developing methods that are not only technically sound but also accessible and trustworthy for end-users. With continuous advances in the field, feature attribution will remain integral to building responsible, transparent, and accountable AI systems.

References

[1] Hooker, S., Erhan, D., Kindermans, P. J., & Kim, B. (2019). A benchmark for interpretability methods in deep neural networks. Advances in neural information processing systems, 32.

[2] C. Garbin. Machine learning interpretability with feature attribution. Retrieved from https://cgarbin.github.io, May 2, 2025.

[3] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, 82-115.

[4] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[5] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

[6] Molnar, C. (2020). Interpretable machine learning. Lulu. com.

[7] Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825.

[8] Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

[9] Sundararajan, M., Taly, A., & Yan, Q. (2017, July). Axiomatic attribution for deep networks. In International conference on machine learning (pp. 3319-3328). PMLR.

[10] Shrikumar, A., Greenside, P., & Kundaje, A. (2017, July). Learning important features through propagating activation differences. In International conference on machine learning (pp. 3145-3153). PMlR.
[11] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140.
[12] Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning, 193-209.
[13] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).
[14] Jain, S.,& Wallace, B. C. (2019). Attention is not explanation. arXiv preprint arXiv:1902.10186.
[15] Fantozzi, P., & Naldi, M. (2024). The explainability of transformers: Current status and directions. Computers, 13(4), 92.

[1]Faculty of Engineering, Tajik Technical University, Dushanbe, Tajikistan.
Email address: said.kamolov@yahoo.com; said.kamolov@ttu.tj