Improving Explainable AI in Machine Learning Models Using SHAP

Emily Hed

I. Introduction

As **artificial intelligence (AI)** advances, smart machines are increasingly integrated into our daily lives. Recommendation systems, text analytics, autonomous vehicles, medical diagnostics, and other AI-driven technologies are shaping critical aspects of society, and many rely on these tools. However, as these systems become more complex, understanding a model's decision-making process is essential for detecting bias, debugging errors, ensuring transparency, and complying with regulations.

Explainable AI (XAI) elucidates AI decision-making using interpretability-focused machine learning techniques to understand model outputs. Current literature reveals two distinct XAI approaches: **machine learning (ML)** and **user experience (UX)**.

- The ML approach utilizes tools to explain model output. The field seeks to advance model performance, identify and mitigate bias, debug errors, and promote industry trust in AI systems [4].
- The UX approach investigates how users perceive and interact with AI explanations, drawing from psychology and **human-computer interaction (HCI)**. This approach evaluates whether a model's output is interpretable (users can understand it) and explainable (users can predict how changes in input affect output). This approach aims to foster user trust and reduce perceived risk [3].

XAI machine learning techniques can be categorized using model characteristics: global explanations, local explanations, model-specific, and model-agnostic (Figure 1).

SHAP (SHapley Additive exPlanations) is an XAI framework established on Shapley values from coalitional game theory. Shapley values provide a holistic view of how each feature (variable) influences a model's predictions [8]. SHAP's generic implementation is **model agnostic**, meaning SHAP can be applied to any model, regardless of underlying structure [1]. A key advantage of SHAP is its ability to provide both local and global model explanations [1].

- **Global explanations** describe the model as a whole, revealing which features exert the greatest influence on the model's predictions.
- **Local explanations** quantify how each feature impacts a single prediction.

SHAP is an open-source Python library that computes "SHAP values" [1] by considering all possible feature combinations and averaging their marginal contributions. This paper explores SHAP's role in enhancing machine learning interpretability, evaluating its strengths and weaknesses, and examining the SHAP library's key features.

[1]SHAP values are Shapley values applied to a conditional expectation function of a machine learning model [7].

II. SHAP

A. Overview

SHAP is an explainability method based on Shapley values in coalitional game theory. Coalitional game theory, also known as cooperative game theory, is a model that describes how groups of players, or coalitions, work together.

Let's look at an analogy:

> Imagine you're at a potluck dinner where each guest brings a dish. The overall meal enjoyment depends on the combination of dishes. Some dishes, like a well-seasoned main course, might have a greater impact on the meal's success, while others, like a simple side dish, contribute less. How do we determine how much each guest contributed to the overall meal enjoyment?

In this analogy, each guest represents a *feature* (variable), and their dish is the *contribution*. The overall meal satisfaction is the *prediction*. Shapley values determine how much each guest contributed to the meal enjoyment by evaluating different combinations of dishes and the resulting prediction. In other words, Shapley values quantify how each feature contributes to the overall prediction and how that prediction changes when joined to every possible combination of features [11].

Feature contribution [2] is a local measure of how a specific feature contributes to a single prediction whereas **feature importance** is a global measure that summarizes the overall impact of a feature across all predictions.

By considering all possible combinations of features, Shapley values provide a holistic view of how each feature influences the model's prediction.

B. Mathematical Properties and Strengths

SHAP has numerous strengths beyond feature importance. Shapley values are considered the "definition of a fair weight" due to their mathematically desirable axioms [2], [5]:

- *Efficiency.* The sum of all feature contributions equals the difference between the prediction and the model's average, ensuring Shapley values are fairly distributed among features. For example, if the model predicts 60

[2]*Feature contribution*, in the context of Shapley values, is synonymous with *Shapley values* and *SHAP values*. Therefore, the Shapley value or SHAP value *is* the feature contribution.

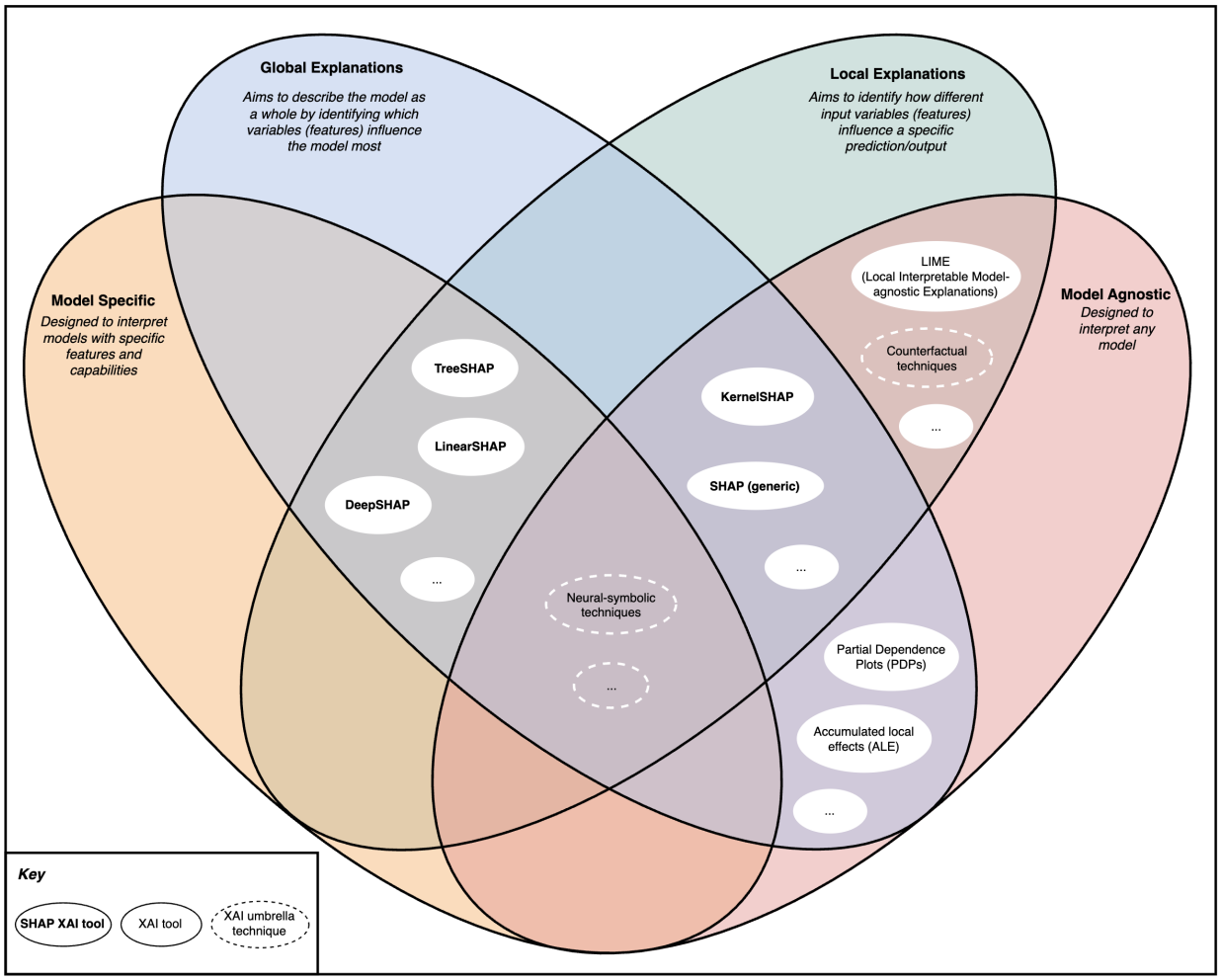

Fig. 1. A Venn diagram categorizing XAI techniques by global vs. local explanations and model-specific vs. model-agnostic approaches. An XAI technique can provide global and/or local explanations, but it cannot be both model-specific and model-agnostic. *TreeSHAP*, for example, provides both local and global explanations and is model-specific. Created by the author, compiled using resources from [4], [1].

and the average prediction is 50, the Shapley values sum to 10.

- *Symmetry.* If two features contribute equally to all possible coalitions, their Shapley values are equal.
- *Dummy.* A feature that does not impact the predicted value has a Shapley value of 0.
- *Linearity.* If two coalition games are combined, the Shapley value for each feature is the sum of its values in both games.

These properties ensure that SHAP is a fair and reliable method for explaining model predictions.

As a model-agnostic XAI method, SHAP can explain any machine learning model, including enigmatic black-box models like deep neural networks.

Another key advantage of SHAP is its ability to provide both local and global explanations through plot visualization tools.

- *Local explanations* provide instance-specific explanations [1]. For example, consider a model that predicts the probability of a loan applicant defaulting based on factors like annual income and credit score. A local explanation lists the SHAP value for each feature, indicating how it influenced the final prediction. Figure 3 displays the global average prediction, $E[f(x)] = 0.28$, and shows that a *Monthly Debt* of 12316 *increased* the probability of this specific individual defaulting on their loan by 0.04.
- *Global explanations* average SHAP values across instances to reveal features that generally exert the greatest influence on predictions [1]. Figure 2 indicates *Current Loan Amount* typically has the greatest influence on individual predictions. The information gained from global analysis can be used to fine-tune the model and address potential biases [2].

C. Use Cases

SHAP can interpret model output for numeric and non-numeric data, such as text. The following examples illustrate SHAP's use cases based on a model's training data:

- *Tabular.* Structured data organized in rows and columns, typically stored in spreadsheets. Each row is an observation, and each column is a specific observation feature (e.g., age, gender, income). Examples include census data and medical records.

- *Text.* Unstructured natural language data such as documents, articles, social media posts, or emails.
- *Image.* Image data, represented as pixel grids with height, width, and color channels (e.g., RGB channels in a color image). Examples include satellite and medical images.
- *Genomic.* Genetic data from organisms, typically a sequence of nucleotides (A, T, C, G in DNA) or other biological features. Examples include DNA sequences and gene expression data.

SHAP is compatible with any ML model, regardless of training data.

D. Explainers

SHAP is a model-agnostic and model-specific XAI tool, as the SHAP library possesses several "explainers," some of which are model-specific and others model-agnostic (Figure 1). For example, the model-specific `TreeExplainer` (known as *TreeSHAP*) is specifically designed for tree-based ML models (e.g., RandomForest, XGBoost, LightGBM, CatBoost). The model-agnostic `KernelExplainer` (known as *KernelSHAP*) is compatible with any machine learning model. Other explainers include LinearSHAP, DeepSHAP, and GradientSHAP, among others [7].

E. Visualization Methods

After selecting an appropriate explainer for a model's architecture, SHAP has several plots for visualizing the calculated SHAP values. Each tool provides unique data insights. Below are a few examples [7]:

- *Bar plot.* Displays each feature's *global average* SHAP value, representing its average contribution to the target variable (Figure 2). Features are ranked by influence, from most to least. Variations include local bar plot and cohort bar plot.
- *Waterfall plot.* Displays the vector SHAP value for each feature in a *local* prediction, illustrating each feature's contribution (Figure 3). The waterfall structure reveals the additive nature of positive and negative contributors from the model's base value (global average prediction) to the local prediction, building from the bottom up. The most important feature is listed first.
- *Heatmap.* Displays a plot with all instances on the x-axis, features on the y-axis, and SHAP values encoded on a color scale. Darker colors represent greater SHAP effects. Figure 4 illustrates the SHAP values for a model trained to predict whether individuals in the 1990s earned more than $\$50,000$ per year. The black bar chart to the right depicts each feature's global mean SHAP value, while the $f(x)$ line represents the predicted value for the specific instance.
- *Beeswarm plot.* Displays the distribution of SHAP values for each feature across all instances in the data set (Figure 5). Where SHAP values are dense, points are stacked vertically. The x-axis represents the SHAP value, indicating the importance of each feature in determining the prediction. The point's color represents the feature's value (red is high, blue is low). A red dot increases the prediction value, whereas a blue dot decreases.

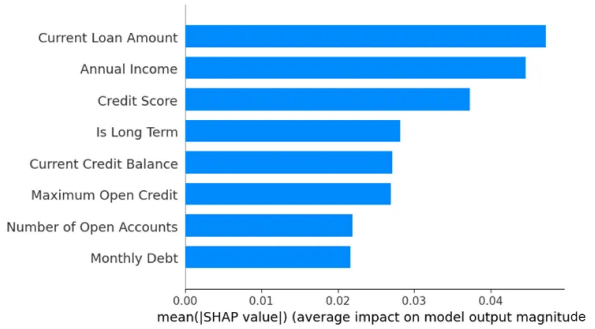

Fig. 2. A SHAP bar plot [9]. Displays each feature's *global average* SHAP value, representing its average contribution to the target variable. Features are ranked by influence, from most to least.

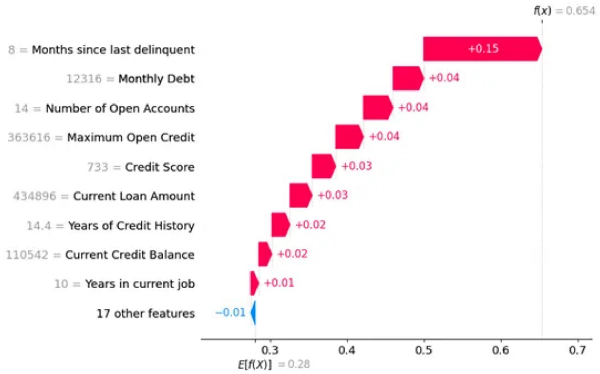

Fig. 3. A SHAP waterfall plot [9]. Displays the vector SHAP value for each feature in a *local* prediction, illustrating each feature's contribution. The waterfall structure reveals the additive nature of positive and negative contributors from the model's base value (global average prediction) to the local prediction, building from the bottom up. The most important feature is listed first.

F. Limitations

Shapley values in their original form suffer from exponentially increasing computational complexity as the number of features grows [1]. Implementations of SHAP, such as KernelSHAP and DeepSHAP, estimate Shapley values using fewer computational resources.

Although Shapley values offer comprehensive insight into a model's decision making, SHAP is prone to the following limitations:

- *Challenges with high-dimensional data.* A high-dimensional dataset is one with many features. SHAP can become computationally infeasible with high-dimensional data, limiting its ability to provide accurate and timely explanations [2].
- *Interpretability vs. Complexity (Accuracy) trade-off.* SHAP offers local interpretability at the cost of global accuracy. It can approximate each feature's contribution, but may not accurately reflect the model's global, intricate relationships [2].
- *Sensitivity to correlated features.* SHAP assumes that features are independent, but real-world datasets often

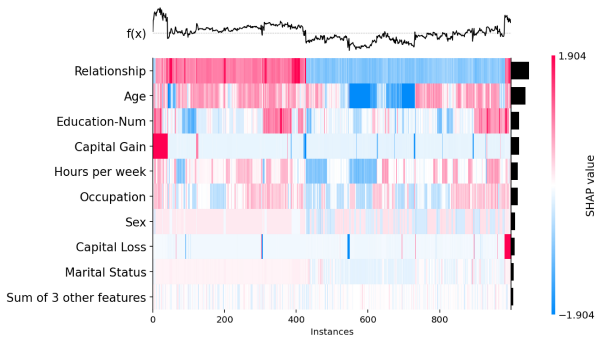

Fig. 4. A SHAP heatmap plot [7]. Displays a plot with all instances on the x-axis, features on the y-axis, and SHAP values encoded on a color scale. Darker colors represent greater SHAP effects. This chart depicts the SHAP values for a model trained to predict whether individuals in the 1990s earned more than $\$50,000$ per year. The black bar chart to the right depicts each feature's global mean SHAP value, while the $f(x)$ line represents the predicted value for the specific instance.

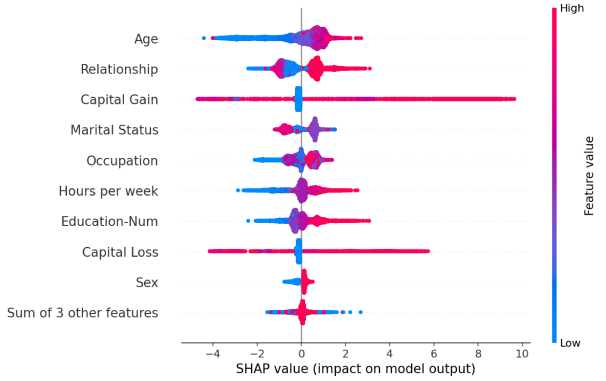

Fig. 5. A SHAP beeswarm plot [7]. Displays the distribution of SHAP values for each feature across all instances in the data set. Where SHAP values are dense, points are stacked vertically. The x-axis represents the SHAP value, indicating the importance of each feature in determining the prediction. The point's color represents the feature's value (red is high, blue is low). A red dot increases the prediction value, whereas a blue dot decreases.

contain correlated variables. SHAP cannot be used for causal inference [10].

- *Subject to human error.* An accurate interpretation of SHAP values requires domain expertise. Without it, there is a risk of misinterpretation, confirmation bias, and misleading conclusions [10].

A 2023 paper, *The Inadequacy of Shapley Values for Explainability*, argues that Shapley values can provide misleading results [6]. Further analysis and cross-validation of these claims will be explored in Phase C.

III. Future Research

In future iterations of my research, I will aim to answer the following questions:

1) How are Shapley values calculated? What equations are used?
2) What are TreeSHAP's underlying computational mechanics?
3) How does SHAP compare with other XAI methods, such as LIME (Local Interpretable Model-Agnostic Explanations)?
4) Given discrepancies in the adequacy of Shapley values for explainability, is reconciling these inconsistencies possible?

IV. Conclusion

As AI becomes more integrated into society, transparency in model decision-making is essential for debugging errors, detecting bias, ensuring transparency, and complying with regulations.

SHAP, a model-agnostic XAI tool, leverages Shapley values from cooperative game theory to provide a mathematically rigorous approach for determining feature contributions in machine learning models. SHAP provides both local and global explanations.

The SHAP Python library includes multiple visualization tools, such as bar plots, waterfall plots, heatmaps, and beeswarm plots, each offering unique insights into feature importance and model behavior.

Despite its strengths, SHAP is not without limitations. Challenges such as computational complexity, sensitivity to feature correlation, and the necessity of domain expertise for accurate interpretation highlight areas for further refinement.

Future research will investigate the mathematical equations used to derive Shapley values, examine TreeSHAP's computational mechanics, and compare SHAP and other XAI methods, such as LIME (Local Interpretable Model-Agnostic Explanations). Additionally, given discrepancies regarding the adequacy of Shapley values for explainability, future research will investigate whether these inconsistencies can be reconciled.

References

[1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.
[2] Brain John Aboze. A comprehensive guide into shap values, May 2023.
[3] David A. Broniatowski. Psychological foundations of explainability and interpretability in artificial intelligence. Apr 2021.
[4] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, and Rajiv Ranjan. Explainable ai (xai): Core ideas, techniques, and solutions. 55(9), January 2023.
[5] Soufiane Fadel and Statistics Canada. Explainable machine learning, game theory, and shapley values: A technical review, Feb 2022.
[6] Xuanxiang Huang and Joao Marques-Silva. The inadequacy of shapley values for explainability, 2023.
[7] Scott Lundberg. Welcome to the shap documentation, 2018.
[8] Scott Lundberg. shap, Aug 2023.
[9] Brendan Ng. Interpreting loan default prediction models using shap, Jan 2024.
[10] Conor O'Sullivan. 4 significant limitations of shap, Apr 2023.
[11] Marek Pawlicki, Aleksandra Pawlicka, Federica Uccello, Sebastian Szelest, Salvatore D'Antonio, Rafał Kozik, and Michał Choraś. Evaluating the necessity of the multiple metrics for assessing explainable ai: A critical examination. *Neurocomputing*, 602:128282, 2024.