Beyond TreeSHAP: Efficient Computation of Any-Order Shapley Interactions for Tree Ensembles

Maximilian Muschalik[1,*], **Fabian Fumagalli**[2,*], **Barbara Hammer**[2], **Eyke Hüllermeier**[1]

[1]LMU Munich, MCML Munich, D-80539 Munich, Germany
[2]Bielefeld University, CITEC, D-33619 Bielefeld, Germany
maximilian.muschalik@lmu.de, ffumagalli@techfak.uni-bielefeld.de, bhammer@techfak.uni-bielefeld.de, eyke@lmu.de

Abstract

While shallow decision trees may be interpretable, larger ensemble models like gradient-boosted trees, which often set the state of the art in machine learning problems involving tabular data, still remain black box models. As a remedy, the Shapley value (SV) is a well-known concept in explainable artificial intelligence (XAI) research for quantifying additive feature attributions of predictions. The model-specific TreeSHAP methodology solves the exponential complexity for retrieving exact SVs from tree-based models. Expanding beyond individual feature attribution, Shapley interactions reveal the impact of intricate feature interactions of any order. In this work, we present TreeSHAP-IQ, an efficient method to compute any-order additive Shapley interactions for predictions of tree-based models. TreeSHAP-IQ is supported by a mathematical framework that exploits polynomial arithmetic to compute the interaction scores in a single recursive traversal of the tree, akin to Linear TreeSHAP. We apply TreeSHAP-IQ on state-of-the-art tree ensembles and explore interactions on well-established benchmark datasets.

Introduction

Tree-based ensemble methods, in particular gradient-boosted trees (Friedman 2001), such as XGBoost (Chen and Guestrin 2016) or LightGBM (Ke et al. 2017), are among the most popular machine learning (ML) models and often achieve state-of-the-art (SOTA) performance on tabular data without extensive hyperparameter tuning (Shwartz-Ziv and Armon 2022). These ensemble methods utilize intricate prediction functions by employing tree structures of high depth, thereby obstructing interpretation of the model's internal reasoning. Yet, understanding a model's prediction is necessary for safe and reliable deployment, alongside addressing ethical and regulatory considerations (Adadi and Berrada 2018). Additive feature attributions, which split the individual features' contributions to the prediction, are a prevalent approach to improving the local interpretation of ML models (Lundberg and Lee 2017; Covert and Lee 2021; Chen et al. 2023). However, in complex real-world applications, such as bioinformatics (Lunetta et al. 2004; Boulesteix et al. 2012; Winham et al. 2012; Wright, Ziegler, and König 2016) or language-related tasks (Tsang, Rambhatla, and Liu 2020) features only attain meaningfulness when *interacting* with other features. In such scenarios, information about interactions complements additive feature attributions, which only show part of the picture (Wright, Ziegler, and König 2016).

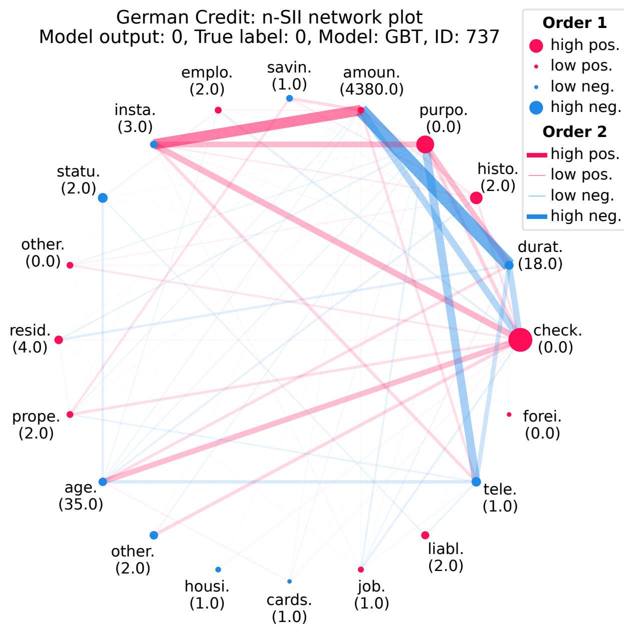

Figure 1: Network Plot after (Inglis, Parnell, and Hurley 2022) for a test instance of the *German Credit* dataset for visualizing local feature attribution and interaction.

In this work, we are interested in model-specific local XAI measures for tree-based models, such as XGBoost. In particular, the extension of predominant attribution measures based on the Shapley value (SV) (Shapley 1953) to any-order additive Shapley-based interactions to explain single predictions locally. Our work extends path dependent TreeSHAP (Lundberg et al. 2020), which exploits the structure of trees to reduce time complexity from exponential to polynomial, to any-order Shapley-based interactions.

Related Work. The SV (Shapley 1953) is a concept from cooperative game theory that has been proposed for model-agnostic explanations for local (Strumbelj and Kononenko

[*]These authors contributed equally.

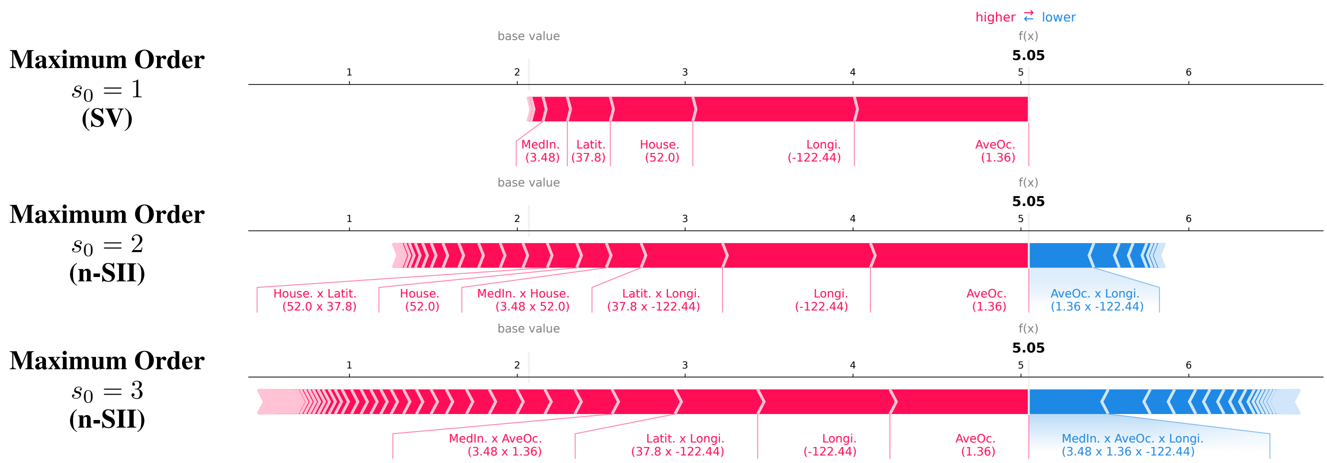

Figure 2: Force plots of positive (red) and negative (blue) SVs and n-SII scores for an instance of the *California* dataset The *longit.* feature has a high contribution, describing the proximity to the ocean, which affects the price. *TreeSHAP* ($s_0 = 1$) reveals this contribution. It also shows that *latit.* contributed positively. *TreeSHAP-IQ*, e.g. $s_0 \geq 2$, reveals that this contribution can be (mostly) attributed to the interaction *latit. x longit.*, which reveals that the *exact location*, and not *latit.*, is meaningful.

2014; Lundberg and Lee 2017) and global (Casalicchio, Molnar, and Bischl 2018; Covert, Lundberg, and Lee 2020) interpretation. In a model-agnostic setting, efficient approximations techniques, based on Monte Carlo (Castro, Gómez, and Tejada 2009; Castro et al. 2017; Kolpaczki et al. 2023; Fumagalli et al. 2023) or the representation of the SV as a constrained weighted least square problem (Lundberg and Lee 2017; Covert and Lee 2021; Jethani et al. 2022) have been proposed to overcome the exponential complexity. For tree-based models the SV can be computed in polynomial time using TreeSHAP (Lundberg et al. 2020) with more efficient variants (Yang 2021). Linear TreeSHAP (Yu et al. 2022) establishes a theoretical foundation that connects the computation to polynomial arithmetic, achieving SOTA computational and storage efficiency.

Limitations of the SV due to correlations and interactions have been widely studied by Slack et al. (2020), Sundararajan and Najmi (2020), and Kumar et al. (2020, 2021). Extensions to interactions have been proposed with the *Shapley Interaction Index* (SII) (Grabisch and Roubens 1999), its aggregation as *n-Shapley Values* (n-SII) (Bordt and von Luxburg 2023), the *Shapley Taylor Interaction Index* (STI) (Sundararajan, Dhamdhere, and Agarwal 2020) and the *Faithful Shapley Interaction Index* (FSI) (Tsai, Yeh, and Ravikumar 2023). All of these are subsumed in the broad class of the *Cardinal Interaction Index* (CII) (Grabisch and Roubens 1999). Model-agnostic approximations have been proposed for general CIIs (Fumagalli et al. 2023), STI (Sundararajan, Dhamdhere, and Agarwal 2020), SII and for FSI (Tsai, Yeh, and Ravikumar 2023). Local pairwise interactions for tree-based models were computed by Lundberg et al. (2020) and for interventional SHAP by Zern, Broelemann, and Kasneci (2023).

Other interaction scores were introduced by Tsang, Rambhatla, and Liu (2020), Zhang et al. (2021), Patel, Strobel, and Zick (2021), Harris, Pymar, and Rowat (2022), and Hiabu, Meyer, and Wright (2023). Interaction scores are further linked to functional decomposition (Hooker 2004, 2007; Lengerich et al. 2020; Herbinger, Bischl, and Casalicchio 2023). For tree-based models, limitations of feature attribution measures (Wright, Ziegler, and König 2016), and efficient implementations for interactions (Lengerich et al. 2020; Hiabu, Meyer, and Wright 2023) were discussed.

So far, any-order Shapley interactions have only been studied in a model-agnostic setting, where the exponential complexity problem is approximately solved. Tree-based approaches have not considered the efficient computation of local any-order Shapley interactions.

Contribution. Our main contributions include;

1. *TreeSHAP-IQ*: An efficient algorithm for computing any-order SII scores for tree ensembles. TreeSHAP-IQ[1] is supported by a mathematical framework based on polynomial arithmetic, akin to Linear TreeSHAP.
2. *Unified Framework*: Application of TreeSHAP-IQ to the broad class of any-order CIIs.
3. *Application*: We efficiently implement TreeSHAP-IQ on SOTA tree-based models, such as XGBoost, and showcase how interaction scores enrich single feature attribution measures on several benchmark datasets.

Local Shapley-Based Explanations

Local Shapley-based explanations consider a model f on an n-dimensional feature space $\mathcal{X}$ with features $N := \{1, \dots, n\}$. The goal is to explain the prediction $f(x) \in \mathbb{R}$ for a selected explanation point $x \in \mathcal{X}$ and find an additive attribution $\phi = (\phi[1], \dots, \phi[n]) \in \mathbb{R}^n$, such that $f(x) = b_0 + \sum_{i \in N} \phi[i]$, where $b_0 \in \mathbb{R}$ is the baseline prediction, i.e. the prediction of x, if no feature information is

[1]TreeSHAP-IQ is implemented as part of the *shapiq* Python package at pypi.org/project/shapiq.

available. To compute a unique attribution score $\phi[i]$ for each feature $i \in N$, we extend the model with subsets of features $f : \mathcal{X} \times \mathcal{P}(N) \to \mathbb{R}$, where $\mathcal{P}(N)$ is the power set of N and $f(x,T)$ refers to the prediction of f at x, if only the features in $T \subseteq N$ are known. In the following, if we omit the subset, then $T = N$, i.e. $f(x) := f(x,N)$. We further omit the explanation point x if it is clear from context, and set $f(T) := f(x,T)$ and $b_0 := f(x,\emptyset)$. The contribution of each feature $i \in N$ is then the SV (Shapley 1953)

$$\phi(f,i) := \sum_{T \subset N \setminus \{i\}} \frac{1}{n \cdot \binom{n-1}{|T|}} \Big[f(T \cup \{i\}) - f(T) \Big].$$

The SVs define the unique attribution measure satisfying the following axioms: linearity (in f), symmetry (ordering does not impact ϕ), dummy (no impact on f implies $\phi(f,i) = 0$) and efficiency $f(x) = b_0 + \sum_{i \in N} \phi(f,i)$ (Shapley 1953).

In many real-world applications, single feature importance scores are not sufficient to understand a model, where features become only meaningful when *interacting* with others. The SV does not give any information about such *interactions* between two or more features. The SII has been the first extension of the SV to interactions of feature subsets.

Definition 1 (SII, Grabisch and Roubens 1999)**.** *The SII for an interaction $S \subseteq N$ is defined as*

$$I^{SII}(f,S) := \sum_{T \subseteq N \setminus S} \frac{1}{(n - |S| + 1) \cdot \binom{n-|S|}{|T|}} \delta_S(f,T),$$

where δ_S is the S-derivative of f for $T \subseteq N \setminus S$, i.e.

$$\delta_S(f,T) := \sum_{L \subseteq S} (-1)^{|S|-|L|} f(T \cup L).$$

The SII is the unique attribution measure that fulfills the (generalized) linearity, symmetry and dummy axiom, as well as a novel recursive axiom that links higher to lower order interactions (Grabisch and Roubens 1999). In contrast to the SV, the SII does not fulfill the (generalized) efficiency axiom, which states that the sum of interaction scores (including b_0) up to a *maximum order* s_0 equals the model prediction $f(x)$. This axiom is particularly useful in the ML context. Recently, Bordt and von Luxburg (2023) proposed a specific aggregation, known as n-SII of order s_0, which yields a unique index that satisfies the (generalized) efficiency axiom. A more general class constitutes the CII, where it was shown that every interaction index fulfilling the linearity, symmetry and dummy axiom can be represented as a CII (Grabisch and Roubens 1999, Proposition 5). Other CIIs were proposed that introduce a unique interaction index of order s_0 and require the efficiency axiom directly, such as the STI (Sundararajan, Dhamdhere, and Agarwal 2020) or the FSI (Tsai, Yeh, and Ravikumar 2023). While the computation of the SV and SIIs are of exponential complexity, it has been shown that the complexity for the SV can be reduced to polynomial time in the case of tree-based models.

The Shapley Value for Tree Ensembles

For tree-based models the computational complexity of the SV can be drastically reduced by utilizing the additive tree structure. Furthermore, there exists a natural way to handle missing features, which can be used to define the extended model $f(x,T)$. For simplicity, we consider in the following a single decision tree, where ensembles of trees can be similarly computed due to the linearity of the SV.

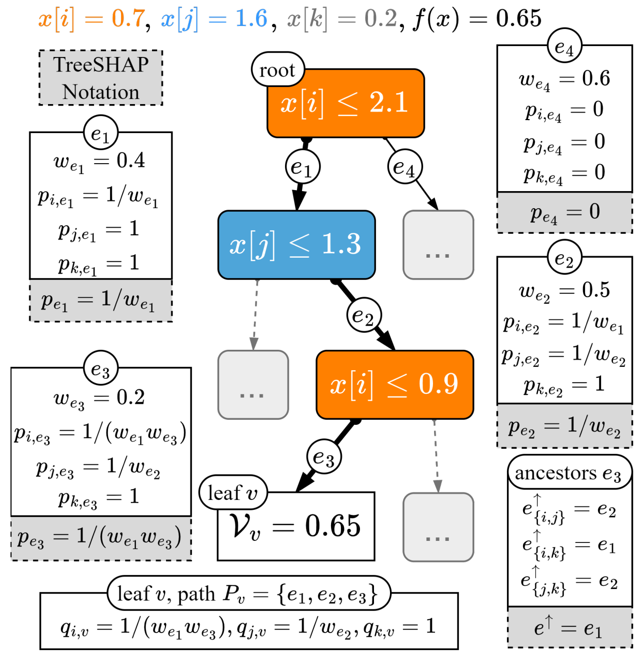

Figure 3: Notations in TreeSHAP-IQ and Linear TreeSHAP.

Notation. We consider a decision tree $\mathcal{T} = (V,E)$ as a rooted directed tree with a set of vertices V, referred to as decision nodes, and edges E. The root node is denoted as $r \in V$. Each decision node consists of a split feature $i \in N$ with a threshold value and predictions $\mathcal{V}_v$ at the leaf nodes. For each node $v \in V$, we let P_v be the set of edges from the root node to v and $\mathcal{L}(v)$ the set of leaf nodes reachable from v, where $\mathcal{L}(\mathcal{T}) = \mathcal{L}(r)$ is the set of all leaf nodes in the tree. For every edge $e \in E$ going from u to v, we denote u as the tail of e and v as the head of e, $h(e)$. We consider a weighted tree with weights $w_e \in (0,1)$ for every edge $e \in E$, which is defined as the proportion of observed data points at the tail of e, that split to the head of e. Additionally, we label each edge $e \in E$ with the feature associated with the tail of e, i.e. the feature that was used to split the observations on the decision node at the source of e. Further, $P_{i,v}$ and E_i are the edges in P_v and E with label $i \in N$. Our notation for decision trees is illustrated in Figure 3.

We also require polynomial arithmetic and refer to the set of polynomials with maximum degree d and coefficients in $\mathbb{R}$ as $\mathbb{R}[x]_d$. Polynomial multiplication is denoted with $\odot$ and division with $\lfloor \frac{a}{b} \rfloor$ or $\lfloor a/b \rfloor$. We denote with $\langle x,y \rangle$ the inner product of two vectors $x, y \in \mathbb{R}^d$ and refer to the inner product of the coefficients, if polynomials are considered.

Extended Model $f(x,T)$ for Decision Trees. A decision tree can be decomposed into distinct decision rules $R^v : \mathcal{X} \to \mathbb{R}$ for each leaf $v \in \mathcal{L}(\mathcal{T})$, which predict $\mathcal{V}_v$, if

x reaches v and zero otherwise. Note that each R^v induces a subspace of $\mathcal{X}$ at which the prediction of f is constant. The decision tree is thus given as $f(x) = \sum_{v\in\mathcal{L}(\mathcal{T})} R^v(x)$. We now define R^v_T, the prediction rule restricted to a set of *active* features $T \subseteq N$, where the remaining are considered to be unknown. If the split feature is unknown, we split based on the weights w_e, which is a common practice (Yu et al. 2022). If all features are unknown, we define $R^v_\emptyset := R^v_\emptyset(x) := \mathcal{V}_v \prod_{e\in P_v} w_e$. When adding feature $i \in N$ to the active set, the product of associated weights is replaced by the split criterion. This is formalized as a recursive property $R^v_{T\cup\{i\}} = q_{i,v}(x) R^v_T(x)$, where $q_{i,v}$ is the marginal effect of adding i to the active set. To define $q_{i,v}$, we let $x \in \pi_i(R^v)$, if $x[i]$ satisfies each split criterion regarding features i in the path of v, i.e. $\pi_i(R^v)$ is the region of feature i in the induced subspace of $\mathcal{X}$ by R^v. For v the marginal effect of adding feature $i \in N$ is then defined as

$$q_{i,v}(x) := \mathbf{1}(x \in \pi_i(R^v)) \prod_{e\in P_{i,v}} \frac{1}{w_e}, \quad (1)$$

where $\mathbf{1}(\cdot)$ is the indicator function. Furthermore, for $P_{i,v} = \emptyset$ we define $q_{i,v} = 1$. The restricted rule is thus defined as

$$R^v_T(x) := R^v_\emptyset \prod_{j\in T} q_{j,v}(x). \quad (2)$$

For a tree $\mathcal{T}_f$ and $T \subseteq N$ the restricted model at x is then

$$f(T) := f(x,T) := \sum_{v\in\mathcal{L}(\mathcal{T}_f)} R^v_T(x).$$

In the following, we omit the argument x in the notation. We proceed to compute the SV of $f(T)$, which is known as *path dependent* TreeSHAP (Lundberg et al. 2020).

Linear TreeSHAP. TreeSHAP exploits the tree structure to compute the SV in polynomial time (Lundberg et al. 2020). Linear TreeSHAP improved this computation and provided a theoretical framework by linking the computation to polynomial arithmetic (Yu et al. 2022). Plugging (2) into the definition of the SV and using the fact that $q_{j,v} - 1 = 0$, if a feature does not appear in the path, yields

$$\phi(R^v, i) = (q_{i,v} - 1) \sum_{T\subseteq\mathcal{F}(R^v)\setminus\{i\}} \frac{R^v_\emptyset}{n \cdot \binom{n-1}{|T|}} \prod_{j\in T} q_{j,v}, \quad (3)$$

where $\mathcal{F}(R^v)$ is the set of all features that appear in R^v. It was shown that this sum can be efficiently stored using the coefficients of a specific polynomial.

Definition 2 (Summary Polynomial (SP), Yu et al. 2022)**.** *The SP of leaf node v is $G^{SP}_v(y) := R^v_\emptyset \prod_{j\in\mathcal{F}(R^v)} (q_{j,v} + y)$.*

For feature $i \in N$, $R^v_\emptyset \sum^{|S|=k}_{S\subset\mathcal{F}(R)\setminus\{i\}} \prod_{j\in S} q_{j,v}$ is the coefficient of y^{d-k-1} in $\frac{G}{q_{i,v}+y}$, where $d := |\mathcal{F}(R^v)|$ is the number of features in each path. Note that this corresponds to the non-weighted terms in the sum of (3) for $k = 0, \dots, d-1$. The SV of a single decision rule can thus be represented as

$$\phi(R^v, i) = (q_{i,v} - 1)\psi\left(\left\lfloor \frac{G_v}{q_{i,v} + y} \right\rfloor\right), \quad (4)$$

where $\psi : \mathbb{R}[x]_d \to \mathbb{R}$ is a function that properly weights the coefficients, such that it corresponds to the sum in (3). It is formally defined (Yu et al. 2022) as

$$\psi_d(A) := \frac{\langle A, B_d \rangle}{d+1} \text{ with } B_d(y) := \sum_{k=0}^{d} \binom{d}{k}^{-1} y^k. \quad (5)$$

We write $\psi(A) = \psi_d(A)$, where d is the degree of A. It was then shown that ψ is additive and scale invariant.

Proposition 1 (Yu et al. 2022)**.** *For ψ and $p, q \in \mathbb{R}[x]_d$,*

$$\psi_d(p + q) = \psi_d(p) + \psi_d(q) \textit{ and } \psi\left(p \odot (1+y)^k\right) = \psi(p).$$

Using (4) based on leaf nodes, a representation of the SV in terms of edges is presented, which is explicitly computed by recursively traversing the tree. For this representation, the SP is extended to every edge in the path of $P_{i,v}$ as

$$G_u := \bigoplus_{v\in\mathcal{L}(u)} G_v \text{ with } G^1 \oplus G^2 := G^1 + G^2 \odot (1+y)^{d_1-d_2},$$

where the order is such that $d_1 > d_2$, i.e. $\oplus$ is an operation on the set of polynomials $\oplus : \mathbb{R}[x]_{d_1} \times \mathbb{R}[x]_{d_2} \to \mathbb{R}[x]_{\max(d_1,d_2)}$ that sums the polynomial while scaling them to the same degree. Note that due to the properties of ψ, we have $\psi(G_u) = \sum_{v\in\mathcal{L}(u)} \psi(G_v)$. For edge $e \in E$ and its feature i, we further introduce the inter-path value of $q_{i,v}$ as

$$p_e := \mathbf{1}\left(x \in \pi_{h(e)}\right) \prod_{e'\in P_{i,h(e)}} \frac{1}{w_{e'}}.$$

Note that $p_{e^*} = q_{i,v}$ if e^* is the last edge in $P_{i,v}$. An edge-based representation of the SV is then provided.

Theorem 1 (Yu et al. 2022)**.** *Let $i \in N$ and denote for e the closest ancestor in the set E_i by $e^\uparrow$, where $e^\uparrow = \perp$ and $p_{i,\perp} = 1$ in case it does not exist. Then,*

$$\phi(f, i) = \sum_{e\in E_i} (p_e - 1)\, \psi\left(\left\lfloor \frac{G_{h(e)}}{y + p_e} \right\rfloor\right) - (p_{e^\uparrow} - 1)\, \psi\left(\left\lfloor \frac{G_{h(e)} \odot (y+1)^{d_{e^\uparrow} - d_e}}{y + p_{e^\uparrow}} \right\rfloor\right).$$

Using this edge-based representation, Linear TreeSHAP computes the SV by traversing once through the tree. To improve efficiency, the SP is stored in a multipoint interpolation form. For more details, we refer to Appendix B.

TreeSHAP-IQ: Computation of Local Shapley Interactions for Tree Ensembles

Computing the exact SV for tree ensembles can reliably quantify the impact of single features on the model's predictions. However, in many applications, certain features become only meaningful when *interacting* with other features. In this case, the SV is not sufficient to understand how the model predicts, and more complex explanations in terms of Shapley interactions are necessary. In the following, we propose TreeSHAP Interaction Quantification (TreeSHAP-IQ), an efficient algorithm for computing any-order SII scores, which follows naturally by extending the SP to interactions. All proofs are deferred to Appendix A.

Theoretical Foundation of TreeSHAP-IQ

We now present the theoretical foundation of TreeSHAP-IQ. The notations in this section extend on Linear TreeSHAP (Yu et al. 2022) and are illustrated in Figure 3. We compute the S-derivative for R^v and $T \subseteq N \setminus S$ as

$$\delta_S(R^v, T) = R^v_T \sum_{L \subset S} (-1)^{|S|-|L|} \prod_{j \in L} q_{j,v}, \quad (6)$$

which follows from (2) and the recursive property. We thus represent the SII for a single decision rule as follows.

Proposition 2. *For a leaf v in $\mathcal{T}_f$, it holds $I^{SII}(R^v, S) =$*

$$\left(\sum_{L \subseteq S} (-1)^{|S|-|L|} \prod_{j \in L} q_{j,v}\right) \psi\left(\left\lfloor \frac{G_v}{\prod_{j \in S}(q_{j,v}+y)} \right\rfloor\right).$$

Proposition 2 yields a compact representation in terms of leaf nodes and decision rules, which reduces to the representation of (4) for single feature subsets. Similar to Linear TreeSHAP, the representation of SII in terms of leaf nodes is not suitable for efficient computation. We thus again establish an edge-based representation, similar to Theorem 1. By Proposition 2, the computation of an interaction for a subset $S \subset N$ requires knowledge of all $q_{i,v}$ with $i \in S$, which have to be tracked during the traversal of the tree. We thus first extend the inter-path values p_e to every feature as

$$p_{i,e} := \mathbf{1}\left(x \in \pi_{i,h(e)}\right) \prod_{e' \in P_{i,h(e)}} \frac{1}{w_{e'}},$$

where $x \in \pi_{i,u}$ if $x[i]$ satisfies each decision criterion in $P_{i,u}$. Note that $p_{j,e} = p_e$ and $\pi_{j,h(e)} = \pi_{h(e)}$, if j is the label of e. Our goal in the following is to provide an algorithm similar to Linear TreeSHAP that traverses the decision tree once and recursively computes the interaction scores. The SP thereby remains unchanged, but we introduce further polynomials of order $|S|$ to efficiently maintain the sum as well as the denominator in Proposition 2.

Definition 3 (Interaction Polynomial (IP))**.** *The IP of $S \subset N$ and edge e is $H^{IP}_{S,e}(y) := \prod_{j \in S}(p_{j,e} - y)$.*

Note that the coefficient of y^k in $H_{S,e}$ is exactly $\sum_{L \subset S}^{|L|=|S|-k} (-1)^{|L|} \prod_{j \in L} p_{j,e}$ for $k = 0, \ldots, |S|$. Therefore, the sum of the coefficients of the IP equals the sum in (6). We thus define the coefficient sum.

Definition 4 (Coefficient sum κ)**.** *We define the function $\kappa_d : \mathbb{R}[x]_d \to \mathbb{R}$ as $\kappa_d(A) := \langle A, y^d + \cdots + y + 1 \rangle$. We write $\kappa(p) = \kappa_d(p)$, where d is the degree of p.*

Applying κ to the IP yields the following properties.

Proposition 3. *For the sum of coefficients of the IP, it holds*

$$\kappa(H^{IP}_{S,e}) = \sum_{L \subseteq S} (-1)^{|S|-|L|} \prod_{j \in L} p_{j,e}. \quad (7)$$

If there exists $j \in S$ with $p_{j,e} = 1$, then $\kappa(H^{IP}_{S,e}) = 0$.

Proposition 3 shows that $\kappa(H^{\text{IP}}_{S,e})$ corresponds to the edge-based representation of the sum in Proposition 2. If e is the last edge in $P_{j,v}$, then $p_{j,e} = q_{j,v}$ for all $j \in N$ and thus $\kappa(H^{\text{IP}}_{S,e})$ retrieves the sum in Proposition 2. Furthermore, if $p_{j,e} = 1$, then it is intuitive that all inter-path contributions with $j \in S$ are zero, since j does not impact the model's prediction in this part of the tree. This property allows us to update interaction scores only if all features of the subset have occurred in the path. We further describe the quotient in Proposition 2 using another polynomial of order $|S|$.

Definition 5 (Quotient Polynomial (QP))**.** *The QP of $S \subset N$ and edge e is $H^{QP}_{S,e}(y) := \prod_{j \in S}(p_{j,e} + y)$.*

If e^*_S is the last edge in P_v of leaf node v that contains any feature of S, then $p_{j,e^*_S} = q_{j,v}$ for every $j \in S$ and hence we can rewrite Proposition 2 using Proposition 3 as

$$I^{\text{SII}}(R^v, S) = \kappa(H^{\text{IP}}_{S,e^*_S})\psi\left(\left\lfloor G_v / H^{\text{QP}}_{S,e^*_S} \right\rfloor\right). \quad (8)$$

Clearly, Proposition 2 reduces to (4) for the case of the SV. In contrast to the SV, the edge-based computation includes all inter-path values of $p_{j,e}$ with $j \in S$. To extend Theorem 2, we therefore need to extend the notion of ancestor edges to ancestors with respect to a subset $S \subset N$.

Proposition 4. *For a decision rule R^v of a leaf node v and a subset $S \subset N$, let $P_{S,v} := \bigcup_{i \in S} P_{i,v}$ and $e^{\uparrow}_S$ as the closest ancestor of e in $P_{S,v}$. The SII of R^v is then given by*

$$I^{SII}(R^v, S) = \sum_{e \in P_{S,v}} \kappa(H^{IP}_{S,e})\psi\left(\left\lfloor \frac{G_v \odot (y+1)^{d_e - d_v}}{H^{QP}_{S,e}} \right\rfloor\right) - \kappa(H^{IP}_{S,e^{\uparrow}_S})\psi\left(\left\lfloor \frac{G_v \odot (y+1)^{d_{e^{\uparrow}_S} - d_v}}{H^{QP}_{S,e^{\uparrow}_S}} \right\rfloor\right).$$

Using Proposition 4, we can state our main theorem.

Theorem 2. *For $S \subset N$, let $E_S := \bigcup_{i \in S} E_i$ be the set of edges that split on any feature in S, and denote the closest ancestor of e in $P_{S,v}$ as $e^{\uparrow}_S$. The SII is then computed as*

$$I^{SII}(f, S) = \sum_{e \in E_S} \kappa(H^{IP}_{S,e})\psi\left(\left\lfloor \frac{G_{h(e)}}{H^{QP}_{S,e}} \right\rfloor\right) - \kappa(H^{IP}_{S,e^{\uparrow}_S})\psi\left(\left\lfloor \frac{G_{h(e)} \odot (y+1)^{d_{e^{\uparrow}_S} - d_e}}{H^{QP}_{S,e^{\uparrow}_S}} \right\rfloor\right).$$

Note that for $S = \{i\}$, Theorem 2 reduces to Theorem 1.

Implementation of TreeSHAP-IQ. Theorem 2 allows for an efficient computation of the SII, with the SP being handled alike to Linear TreeSHAP. The IQ and QP are updated for each interaction subset that contains the feature of e. We again use the multipoint interpolation form to store and update the polynomials G_v, $H^{\text{IP}}_{S,e}$, and $H^{\text{QP}}_{S,e}$. TreeSHAP-IQ traverses the decision tree once for every explanation point. At each edge (decision node), TreeSHAP-IQ updates all interactions that contain the currently encountered feature, $\binom{n-1}{|S|-1}$ in total. However, the update can be restricted to those interactions, where all features have been observed in the path. We refer to Appendix B for more details.

Complexity of TreeSHAP-IQ. Consider m explanation points, $\ell_{\mathcal{T}} := |\mathcal{L}(\mathcal{T})|$ as the number of leaves and $d_{\max}$ as the maximum depth of the tree.

Linear TreeSHAP has a computational complexity of $\mathcal{O}(m \cdot \ell_{\mathcal{T}} \cdot d_{\max})$ and storage complexity of $\mathcal{O}(d_{\max}^2 + n)$ (Yu et al. 2022). We now consider the complexity of TreeSHAP-IQ, if all interactions of order $s := |S|$ are computed. In contrast to Linear TreeSHAP and the SP, where only the current feature value has to be updated, TreeSHAP-IQ needs to update the IP, the QP and the interaction scores for all interaction subsets that contain the currently observed feature. This increases the computational complexity by a factor of $\binom{n-1}{s-1}$. Furthermore, all interaction scores have to be stored, requiring storage of $\binom{n}{s}$. To store the IQ and QP, we require further a storage capacity of $\mathcal{O}(d_{\max}^2 \cdot \binom{n}{s})$. The computational complexity is thus summarized as follows.

TreeSHAP-IQ complexity for the SII of order s

Computational Complexity	Storage Complexity
$\mathcal{O}\left(m \cdot \ell_{\mathcal{T}} \cdot d_{\max} \cdot \binom{n-1}{s-1}\right)$	$\mathcal{O}\left(d_{\max}^2 \cdot \binom{n}{s}\right)$

For the computation of the SV, the computational complexity of TreeSHAP-IQ is similar to Linear TreeSHAP. The storage capacity is increased by a factor n, as we store the IP and QP for every feature. Moreover, for pairwise interactions, TreeSHAP-IQ mirrors the complexity of the computation proposed by Lundberg et al. (2020) using Linear TreeSHAP. However, our method distinguishes itself by relying on a single initialization of the tree parameters.

Extending TreeSHAP-IQ to General CIIs

TreeSHAP-IQ can be extended to the broad class of CIIs. A CII is defined as $I^{\text{CII}}(f,S) := \sum_{T \subseteq N \setminus S} w_s^{\text{CII}}(|T|)\delta_S(f,T)$ with non-negative weights w_s^{CII} that depend on the interaction order $s := |S|$ (Grabisch and Roubens 1999; Fumagalli et al. 2023). This includes other approaches of extending the SV to interactions, such as STI and FSI, as well as Banzhaf interactions (Patel, Strobel, and Zick 2021). Observe from the proofs, that different weights in CIIs solely impact the SP, and in particular ψ. To extend the SP for CIIs, we let $d := |\mathcal{F}(R^v)|$ and scale G_v to the degree of n, which does not impact ψ due to the scale invariance. We then observe

$$\psi\left(\left\lfloor \frac{G_v \odot (1+y)^{n-d}}{\prod_{j \in S}(1+q_{j,v})} \right\rfloor\right) = R_\emptyset^v \sum_{T \subseteq N \setminus S} w_s^{\text{SII}}(t) \prod_{j \in S} q_{j,v},$$

where $w_s^{\text{SII}}(t)$ is the CII weight for SII, cf. Definition 1. Recall from (5) that these weights are retrieved from the polynomial B_{n-s}. Thus, we generalize $\psi_d^{\text{CII}} : \mathbb{R}[x]_d \to \mathbb{R}$ to

$$\psi_d^{\text{CII}}(A) := \langle A, W_d \rangle \text{ with } W_d^{\text{CII}}(y) := \sum_{k=0}^{d} w_{n-d}^{\text{CII}}(k) y^k.$$

If G_v is scaled to degree n, then ψ_d^{CII} is always evaluated with a polynomial of degree $d = n - |S|$. Further, note that for SII, we have $\psi_d(A) \equiv \psi_d^{\text{SII}}(A)$, where the quotient $d+1$ is included in the weights, i.e. $W_d^{\text{SII}}(y) = B_d/(d+1)$.

Datasets	# Instances	# Features	Target	Speed-Up
Credit	1 000	20	$\{0,1\}$	$\sim 10^4$
Bank	45 211	16	$\{0,1\}$	$\sim 10^3$
Adult	45 222	14	$\{0,1\}$	$\sim 10^3$
Bike	17 379	12	$\mathbb{R}$	$\sim 10^1$
COMPAS	6 172	11	$\{0,1\}$	$\sim 10^2$
Titanic	891	9	$\{0,1\}$	$\sim 10^1$
California	20 640	8	$\mathbb{R}$	~ 1

Table 1: Overview of datasets and speed-up compared to a naive computation

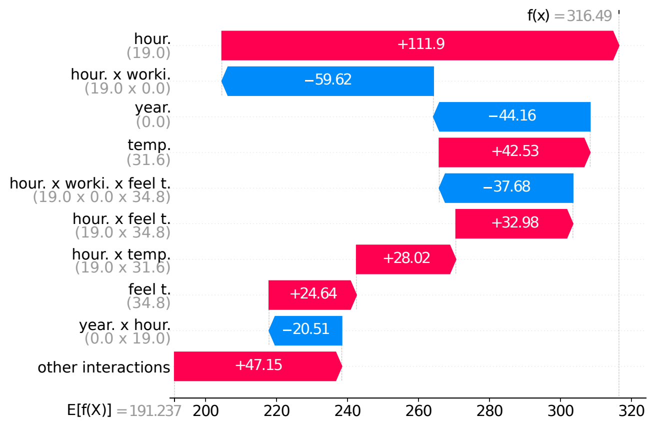

Figure 4: Waterfall chart for n-SII scores with $s_0 = 3$ and a prediction of the *Bike* regression dataset.

Implementation of CIIs in TreeSHAP-IQ. Using ψ^{CII}, any CII can be computed by TreeSHAP-IQ. In contrast to $\psi \equiv \psi^{\text{SII}}$, the scale invariance does not hold for CIIs. Therefore, the SP cannot be reduced to the degree $d := |\mathcal{F}(R^v)|$. However, if we maintain the SP at the maximum degree n, then all previous results apply. If the SP is stored in multipoint interpolation form, then this merely requires a multiplication with the corresponding term of $(y+1)^{n-d}$, which can be efficiently precalculated. Thus, the computational complexity is not affected by this extension. Provided $d_{\max} \geq n$, the storage complexity is not affected either.

Experiments

We apply TreeSHAP-IQ[2] on XGBoost (XBG) (Chen and Guestrin 2016), gradient-boosted trees (GBTs), random forest (RF), and decision tree (DT) algorithms on the *German Credit* (Hofmann 1994), *Bank* (Moro, Cortez, and Laureano 2011), *Adult Census* (Kohavi 1996), *Bike* (Fanaee-T and Gama 2014), *COMPAS* (Angwin et al. 2016), *Titanic* (Dawson 1995), and *California* (Kelley Pace and Barry 1997) datasets, see Table 1. For further experimental results, including a run-time analysis and detailed information on the datasets, models, and pre-processing steps, we refer to Appendix C. We compute additive interactions for single predictions using TreeSHAP-IQ with n-SII of different order.

[2] All experimental code and the technical appendix can be found at: github.com/mmschlk/TreeSHAP-IQ.

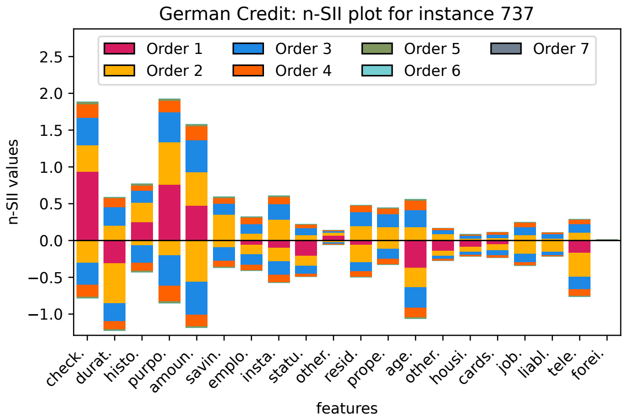

Figure 5: Visualization of positive and negative n-SII scores per feature with $s_0 = 7$ for an observation in *German Credit*.

TreeSHAP-IQ Reveals Intricate Feature Interactions. Using TreeSHAP-IQ, we examine the model's prediction based on higher order interaction effects. We distinguish n-SII scores that positively (red) and negatively (blue) impact the prediction. In Figure 1, we visualize n-SII with $s_0 = 2$. The width of the network vertices (order 1) and the network edges (order 2) describes the absolute value of the corresponding n-SII scores. We observe that there exist features that strongly impact the prediction individually, such as the information about a non-existing *checking account*. However, the present *credit amount* strongly impacts the prediction only in interaction with the given *installment rate* (positively) and *duration* (negatively).

The force plots in Figure 2 illustrate how the additive local explanations change, if higher order interactions are considered. We consider the n-SII scores for $s_0 = 1, 2, 3$ for the *California* housing dataset and an XGBoost regressor. The force plot displays the positive and negative interaction scores starting from the predicted value to the left and right, respectively, sorted by their absolute value. We observe that individual feature effects, such as *Longitude*, reduce when higher order interactions are considered. The interaction of *Longitude* and *Latitude* reveals the importance of the geographic location of this instance.

The waterfall chart in Figure 4 displays the explanations of n-SII with order $s_0 = 3$ for an instance in the *bike* dataset. For this instance, it can be seen that the interaction of the evening *hour* with a non-*working* day affects the prediction negatively, whereas the interaction with both *temperature* features contribute positively.

n-SII Plots Quantify Interactions of Each Feature. To assess the strength of interaction per individual feature, we utilize the visualization of n-SII values presented by Bordt and von Luxburg (2023). We compute exact n-SII scores up to order $s_0 = 7$ for the *German Credit* dataset. The positive and negative interactions are distributed equally onto each feature in the subset and displayed on the positive and negative axes, respectively. The sum of all stacked bars results in the SV of each feature (Bordt and von Luxburg 2023). In Figure 5, we observe that the interaction effects diminish at order 5, with interactions of orders 6 and 7 being virtually absent. Assuming that interactions decay with higher order, this visualization can be used to find the maximum order to explain the prediction (i.e. $s_0 = 5$ from Figure 5).

Limitations

TreeSHAP-IQ applies to the broad class of CIIs, provided that its representation in terms of a weighted sum of discrete derivatives is known. For FSI, this representation is only explicitly known for top-order interactions (Tsai, Yeh, and Ravikumar 2023), as FSI is motivated as a solution to a constrained weighted least square problem. Similar to the SV, Shapley interactions strongly rely on how absent features are modeled. In our work, we considered the *path dependent* feature perturbation (Lundberg et al. 2020), which is linked to the *observational* approach (Chen et al. 2020). The *interventional* approach (Lundberg et al. 2020) can be computed with TreeSHAP-IQ, akin to TreeSHAP, but similarly increases the computational complexity by the number of samples used in the background dataset. In this case, more efficient variants should be used instead (Zern, Broelemann, and Kasneci 2023). Both paradigms yield different explanations, where the appropriate choice should be carefully done depending on the application (Chen et al. 2020).

Conclusion and Future Work

We presented TreeSHAP-IQ, an efficient method to compute any-order additive Shapley interactions that locally explain single predictions for general ensembles of trees. Akin to SOTA Linear TreeSHAP (Yu et al. 2022), our algorithm is based on a solid theoretical foundation that exploits polynomial arithmetic. We applied TreeSHAP-IQ on SOTA ML models, such as XGBoost (Chen and Guestrin 2016), and several benchmark datasets. We demonstrated that TreeSHAP-IQ reveals intricate feature interactions, which enrich Shapley-based feature attribution.

Utilizing well-known visualization and aggregation techniques from machine learning (Lundberg and Lee 2017; Bordt and von Luxburg 2023) and statistics (Inglis, Parnell, and Hurley 2022) we presented these scores in a manner that is easily understandable and interpretable. While interactions are widely studied in statistics, explaining local predictions using interaction scores, in particular with Shapley-based interactions, is an emerging line of research in the field of XAI. Due to the exponentially increasing number of interactions, we provided intuitive visualizations to present TreeSHAP-IQ scores to practitioners. Nevertheless, it would be beneficial to explore further human-centered post-processing techniques and visualizations, as well as rigorously evaluate the explanatory capabilities of TreeSHAP-IQ with user studies, especially to validate quantitatively that the user's understanding increases when higher order explanations are presented. Additionally, the n-SII scores define a local generalized additive model (GAM) (Bordt and von Luxburg 2023) that could be further linked to functional decomposition (Hiabu, Meyer, and Wright 2023).

Acknowledgements

We sincerely thank the anonymous reviewers for their work and helpful comments. We gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824.

References

Adadi, A.; and Berrada, M. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6: 52138–52160.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine Bias. There's software used across the country to predict future criminals. And it's biased against blacks.

Bordt, S.; and von Luxburg, U. 2023. From Shapley Values to Generalized Additive Models and back. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, volume 206 of *Proceedings of Machine Learning Research*, 709–745. PMLR.

Boulesteix, A.-L.; Bender, A.; Lorenzo Bermejo, J.; and Strobl, C. 2012. Random forest Gini importance favours SNPs with large minor allele frequency: impact, sources and recommendations. *Briefings in Bioinformatics*, 13(3): 292–304.

Casalicchio, G.; Molnar, C.; and Bischl, B. 2018. Visualizing the Feature Importance for Black Box Models. In *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML PKDD 2018)*, volume 11051 of *Lecture Notes in Computer Science*, 655–670. Springer.

Castro, J.; Gómez, D.; Molina, E.; and Tejada, J. 2017. Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Computers & Operations Research*, 82: 180–188.

Castro, J.; Gómez, D.; and Tejada, J. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, 36(5): 1726–1730.

Chen, H.; Covert, I.; Lundberg, S.; et al. 2023. Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence*, 5: 590–601.

Chen, H.; Janizek, J. D.; Lundberg, S. M.; and Lee, S. 2020. True to the Model or True to the Data? arXiv:2006.16234.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2016)*, 785–794. ACM.

Covert, I.; and Lee, S. 2021. Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression. In *The 24th International Conference on Artificial Intelligence and Statistics, (AISTATS 2021)*, volume 130 of *Proceedings of Machine Learning Research*, 3457–3465. PMLR.

Covert, I.; Lundberg, S. M.; and Lee, S. 2020. Understanding Global Feature Contributions With Additive Importance Measures. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 (NeurIPS 2020)*.

Dawson, R. J. M. 1995. The "Unusual Episode" Data Revisited. *Journal of Statistics Education*, 3(3).

Fanaee-T, H.; and Gama, J. 2014. Event Labeling Combining Ensemble Detectors and Background Knowledge. *Progress in Artificial Intelligence*, 2(2): 113–127.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5): 1189–1232.

Fumagalli, F.; Muschalik, M.; Kolpaczki, P.; Hüllermeier, E.; and Hammer, B. 2023. SHAP-IQ: Unified Approximation of any-order Shapley Interactions. arXiv:2303.01179.

Grabisch, M.; and Roubens, M. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4): 547–565.

Harris, C.; Pymar, R.; and Rowat, C. 2022. Joint Shapley values: a measure of joint feature importance. In *The Tenth International Conference on Learning Representations, (ICLR 2022)*. OpenReview.net.

Herbinger, J.; Bischl, B.; and Casalicchio, G. 2023. Decomposing Global Feature Effects Based on Feature Interactions. arXiv:2306.00541.

Hiabu, M.; Meyer, J. T.; and Wright, M. N. 2023. Unifying local and global model explanations by functional decomposition of low dimensional structures. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, volume 206 of *Proceedings of Machine Learning Research*, 7040–7060. PMLR.

Hofmann, H. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. DOI: 10.24432/C5NC77.

Hooker, G. 2004. Discovering additive structure in black box functions. In Kim, W.; Kohavi, R.; Gehrke, J.; and DuMouchel, W., eds., *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2004)*, 575–580. ACM.

Hooker, G. 2007. Generalized Functional ANOVA Diagnostics for High-Dimensional Functions of Dependent Variables. *Journal of Computational and Graphical Statistics*, 16(3): 709–732.

Inglis, A.; Parnell, A.; and Hurley, C. B. 2022. Visualizing Variable Importance and Variable Interaction Effects in Machine Learning Models. *Journal of Computational and Graphical Statistics*, 31(3): 766–778.

Jethani, N.; Sudarshan, M.; Covert, I. C.; Lee, S.; and Ranganath, R. 2022. FastSHAP: Real-Time Shapley Value Estimation. In *The Tenth International Conference on Learning Representations (ICLR 2022)*. OpenReview.net.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017 (NeurIPS 2017)*, 3146–3154.

Kelley Pace, R.; and Barry, R. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297.

Kohavi, R. 1996. Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, 202–207.

Kolpaczki, P.; Bengs, V.; Muschalik, M.; and Hüllermeier, E. 2023. Approximating the Shapley Value without Marginal Contributions. arXiv:2302.00736.

Kumar, I.; Scheidegger, C.; Venkatasubramanian, S.; and Friedler, S. A. 2021. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021 NeurIPS 2021*, 26598–26608.

Kumar, I. E.; Venkatasubramanian, S.; Scheidegger, C.; and Friedler, S. A. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, 5491–5500. PMLR.

Lengerich, B. J.; Tan, S.; Chang, C.; Hooker, G.; and Caruana, R. 2020. Purifying Interaction Effects with the Functional ANOVA: An Efficient Algorithm for Recovering Identifiable Additive Models. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, volume 108 of *Proceedings of Machine Learning Research*, 2402–2412. PMLR.

Lundberg, S. M.; Erion, G. G.; Chen, H.; DeGrave, A. J.; Prutkin, J. M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; and Lee, S. 2020. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1): 56–67.

Lundberg, S. M.; and Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, NeurIPS 2017*, 4765–4774.

Lunetta, K. L.; Hayward, L. B.; Segal, J.; and Van Eerdewegh, P. 2004. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics*, 5: 32.

Moro, S.; Cortez, P.; and Laureano, R. 2011. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In *Proceedings of the European Simulation and Modelling Conference (ESM 2011)*.

Patel, N.; Strobel, M.; and Zick, Y. 2021. High Dimensional Model Explanations: An Axiomatic Approach. In *2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event (FAccT 2021)*, 401–411. ACM.

Shapley, L. S. 1953. A Value for n-Person Games. In *Contributions to the Theory of Games (AM-28), Volume II*, 307–318. Princeton University Press.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90.

Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; and Lakkaraju, H. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES 2020)*, 180–186. ACM.

Strumbelj, E.; and Kononenko, I. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3): 647–665.

Sundararajan, M.; Dhamdhere, K.; and Agarwal, A. 2020. The Shapley Taylor Interaction Index. In *Proceedings of the 37th International Conference on Machine Learning, (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, 9259–9268. PMLR.

Sundararajan, M.; and Najmi, A. 2020. The Many Shapley Values for Model Explanation. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, 9269–9278. PMLR.

Tsai, C.; Yeh, C.; and Ravikumar, P. 2023. Faith-Shap: The Faithful Shapley Interaction Index. *Journal of Machine Learning Research*, 24(94): 1–42.

Tsang, M.; Rambhatla, S.; and Liu, Y. 2020. How does This Interaction Affect Me? Interpretable Attribution for Feature Interactions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 6147–6159.

Winham, S. J.; Colby, C. L.; Freimuth, R. R.; Wang, X.; de Andrade, M.; Huebner, M.; and Biernacka, J. M. 2012. SNP interaction detection with Random Forests in high-dimensional genetic data. *BMC Bioinformatics*, 13: 164.

Wright, M. N.; Ziegler, A.; and König, I. R. 2016. Do little interactions get lost in dark random forests? *BMC Bioinform.*, 17: 145.

Yang, J. 2021. Fast TreeSHAP: Accelerating SHAP Value Computation for Trees. arXiv:2109.09847.

Yu, P.; Bifet, A.; Read, J.; and Xu, C. 2022. Linear tree shap. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, (NeurIPS 2022)*.

Zern, A.; Broelemann, K.; and Kasneci, G. 2023. Interventional SHAP Values and Interaction Values for Piecewise Linear Regression Trees. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, (AAAI 2023)*, 11164–11173. AAAI Press.

Zhang, H.; Xie, Y.; Zheng, L.; Zhang, D.; and Zhang, Q. 2021. Interpreting Multivariate Shapley Interactions in DNNs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, (AAAI 2021)*, 10877–10886. AAAI Press.