Succinct Interaction-Aware Explanations

Sascha Xu°
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
sascha.xu@cispa.de

Joscha Cüppers°
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
joscha.cueppers@cispa.de

Jilles Vreeken
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
vreeken@cispa.de

ABSTRACT

Shapley values (Shap) are a popular approach to explaining decisions of black-box models by revealing the importance of individual features. Shap explanations are easy to interpret, but as they do not incorporate feature interactions, they are also incomplete and potentially misleading. Interaction-aware methods such as *n*Shap report the additive importance of *all* subsets up to *n* features. These explanations are complete, but in practice excessively large and difficult to interpret. In this paper, we combine the best of both worlds. We partition the features into significantly interacting groups, and use these to compose a succinct, interpretable explanation. To determine which partitioning out of super-exponentially many explains a model best, we derive a criterion that weighs the complexity of an explanation against its representativeness for the model's behavior. To be able to find the best partitioning, we show how to prune suboptimal solutions using a statistical test. This not only improves runtime but also helps to avoid explaining spurious interactions. Experiments show that iShap represents underlying modeling more accurately than Shap and *n*Shap, and a user study suggests that iShap is perceived as more interpretable and trustworthy.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Artificial intelligence**; • **Human-centered computing**;

KEYWORDS

Explainability, Shapley Values, Interactions

ACM Reference Format:
Sascha Xu, Joscha Cüppers, and Jilles Vreeken. 2025. Succinct Interaction-Aware Explanations. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3690624.3709175

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.14617305.

°Equal contribution

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1245-6/25/08…$15.00
https://doi.org/10.1145/3690624.3709175

1 INTRODUCTION

Decision processes must be fair and transparent regardless of whether it is driven by a human or an algorithm. Post-hoc explainability methods offer a solution as they can generate explanations that are independent of the underlying model f, and are hence also applicable to black-box machine learning models. One of the most popular approaches is the use of Shapley values [25], which provides intuitive explanations for a decision $f(x)$ of an arbitrary model f for an individual x in terms of how much a specific input value x_i contributes to the outcome $f(x)$. Additive explanations over *single features* are succinct and easily understandable, but, only reliable when the underlying model is indeed additive. Whenever there are interactions between features in the model this can lead to misleading results [9].

Interaction index explanations [24, 39, 41] address this by returning *groups of features* x_S that have a non-additive effect on the prediction $f(x)$. *n*Shap [1] is a recent approach that decomposes a prediction $f(x)$ into a generalized additive model $\sum_{S \subset [d], |S| \leq n} \Phi_S^n$, computing the additive contribution of all subsets of up to n features. Although these explanations are complete up to n^{th}-order interactions, the combinatorial explosion makes these arduous to compute and even harder to interpret.

To illustrate, we consider the explanations of Shap and *n*Shap for bike rental prediction. On the left of Fig. 1 we show that of Shap, which reveals the contributions of the 10 individual features. While easy to understand, it is also counterintuitive: winter and humidity are listed as beneficial, while medium temperature is identified as a negative factor for bike rentals. On the right, we show the *n*Shap explanation for subsets of up to $n = 10$ features. 751 subsets get a non-zero score (see Supplement for all values). These are not just many, but also hard to interpret. `Temperature:0.39` is part of 188 such sets. It is scored negatively individually (-702), positively with `Season:4` (259), negatively with `Month:10` (-152), but positively with both (42). We see similar behavior for many features, making it generally hard to say which interactions are truly important.

In the middle of Fig. 1, we show our proposed explanation, iShap. It partitions the features into groups that significantly interact, and gives an additive explanation over these. It reveals that two interactions responsible for the high predicted demand: it is a dry and relatively warm winter day (`Season:4`, `Hum:0.49` and `Temp:0.39`) and a Saturday with little wind (`Weekday:6` and `Windspeed:0.15`).

In the following, we outline the theory and algorithm behind iShap. We formalize an objective function for the ideal partition of an additive explanation, where we seek to find those coalitions of players, which together best approximate the full game as an additive function. The main hurdle is on the computational side due to the combinatorial explosion of the number of partitions. To reduce the search space, we propose a statistical test to prune insignificant

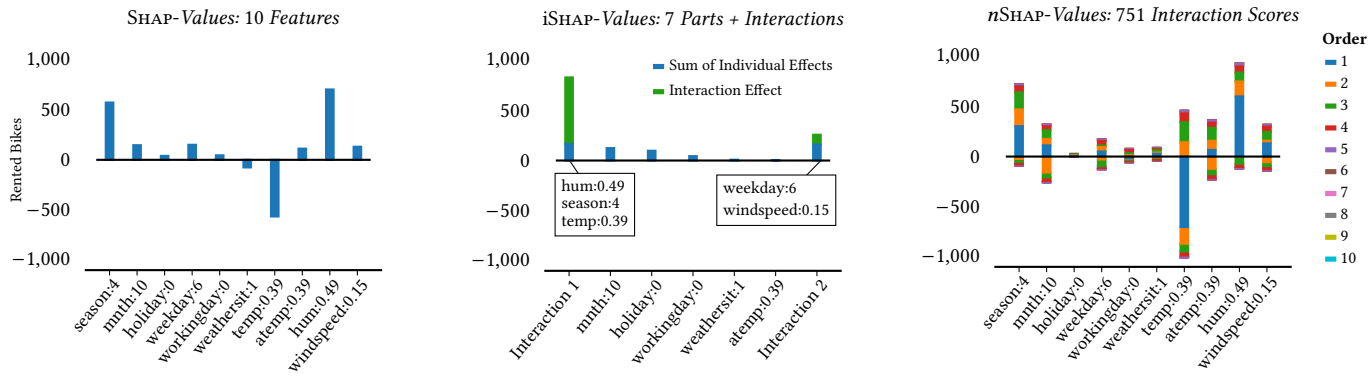

Figure 1: Comparison of SHAP (left), our proposal iSHAP (middle) and *n*SHAP (right) on the Bike Sharing dataset [5]. SHAP does not reveal interactions, *n*SHAP returns non-zero scores for 751 out of 1024 feature sets ($n = d$). iSHAP provides a concise explanation of 2 interactions for the high predicted demand: its is a dry and relatively warm winter day (`Season:4`, `Hum:0.49` and `Temp:0.39`) and a Saturday with little wind (`Weekday:6` and `Windspeed:0.15`).

interactions, searching for the optimal partitioning over the resulting connected components. Over the optimal partition we compute an additive SHAP explanation. We evaluate iSHAP on various benchmarks against both additive and interaction based explanations, including a user study to evaluate the subjective perception of our explanations in terms of understandability and trustworthiness.

2 THEORY

We consider a machine learning model $f : \mathcal{X} \to \mathbb{R}$ with a domain $\mathcal{X}$ over d univariate random variables X_1 to X_d. We denote a subset of input variables by X_S, where S is the index set and denote the set of all indices as $[d] = \{1, ..., d\}$. We define an explanation as a set of tuples $\{(S_i, e_i)\}$ where S_i is an index set with an explanatory value e_i to the prediction $f(x)$. For example, SHAP explains using singletons $S_i = \{i\}$, where e_i are the Shapley values, and *n*SHAP does so over all subsets $S_i \subset [d]$ of cardinality $|S_i| \leq n$ features.

We view the local prediction $f(x)$ as a coalition game, where a coalition x_S is a subset of players receiving a payoff $v(S)$ defined by a value function $v : 2^d \to \mathbb{R}$. W.l.o.g. we assume that the value function is normalized, i.e. $v(\emptyset) = 0$, which can be achieved by pre-processing f so that $E[f(X)] = 0$. In the context of machine learning, two value functions are often used: the observational value function $v(S; f, x) = \mathrm{E}\ [f(X)|X_S = x_S]$ [25], and the interventional value function $v(S; f, x) = \mathrm{E}\ [f(X)|do(X_S = x_S)]$ [16]. Our method is based directly on v and can be instantiated with either. In the following, we therefore omit the specific instance of f and x and refer to the value function simply as $v(S)$.

2.1 Objective

Our goal is to construct an explanation $\{(S_i, e_i)\}$ for $f(x)$ that is succinct, non-redundant, and where the additive interpretation $\sum_i e_i$ approximates the behavior of f best. For example, if f is a linear model, then the value function of a single feature i is the weight w_i times the deviation from the mean, i.e.

$$v(i) = w_i(x_i - \mathrm{E}\,[X_i])\ ,$$

so that the value function of a coalition of features S is the sum of the individual value functions $v(S) = \sum_{i \in S} v(i)$. For complex models such as neural networks however, there generally exists no exact analytic decomposition.

Instead, we focus on finding a partition Π of the features space $[d]$ such that each feature j is contained in only one set S_i. Each feature $j \in S_i$ is hence associated with only one explanatory value e_i. In particular, we want to find that partition Π that minimizes $(f(x) - \sum_{S \in \Pi} v(S))^2$, i.e. the partition Π that *approximates* f of x best. The general idea is that if features x_i and x_j strongly interact, they will have a large joint effect on the result, and hence a local surrogate model would make a large mistake if x_i and x_j are not in the same set S.

It is easy to see that the objective is trivially minimized by $\Pi = \{[d]\}$, which would not give any insight into the inner workings of the model. We therefore regularize the complexity of the explanation by penalizing the number of interactions via an L_0 norm. Each set $S \in \Pi$ represents $\binom{|S|}{2}$ possible pairwise interactions, by which the L_0 norm of a partition Π is $\mathcal{R}(\Pi) = \sum_{S \in \Pi} \binom{|S|}{2}$. We define the optimal partition Π^* with regard to value function v of an algorithmic decision $f(x)$ as the partition Π minimizing

$$\Pi^* = \arg\min_{\Pi} \left(f(x) - \sum_{S_i \in \Pi} v(S_i) \right)^2 + \lambda \cdot \sum_{S_i \in \Pi} \binom{|S_i|}{2} \quad . \tag{1}$$

We next describe how to find Π^*, and then how we can construct an explanation $\{(S_i, e_i)\}$ from it.

2.2 Partitioning

Objective (1) poses a challenging optimization problem. Finding the best partition is a constrained variant of the subset-sum problem and hence NP-hard. The number of partitions for a set of d features is given by Bell number B_d, which grows super-exponentially with d, and hence exhaustive search is not an option. Approximate solutions can be computed in pseudo-polynomial time [30], but require

the value function to be computed for all elements of the power set, of which there are exponentially many.

We observe that when a pair of variables x_i and x_j do not have a significant non-additive effect on the prediction $f(x)$ in the context of *any* set of other variables, our regularizer will ensure they will not be grouped together in the optimal partitioning. This allows us to drastically prune the search space, while still picking up non-linear higher-order effects (e.g. XOR).

Definition 1. *(Eq. (6) Lundberg et al. [24]) Given a value function v, the interaction $\mathcal{I}$ between x_i and x_j in the context of x_S is*

$$\mathcal{I}(i,j,S) = v(S \cup i) + v(S \cup j) - v(S \cup \{i,j\}) - v(S) \ .$$

This definition of interaction measures the effect of setting $X_i = x_i$ and $X_j = x_j$ individually, in contrast to the combined effect, whilst accounting for a covariate set x_S. We now show, that if for any covariate set S, there is no interaction between x_i and x_j, then i and j are not be grouped in the optimal partition with regard to Objective (1). To this end, we begin by showing that the additivity of effects for a pair x_i and x_j is a sufficient criterion to rule out their pairing, and then show in which cases we can use the absence of interaction as an indicator for additivity.

Theorem 1. *Let v be additive for the variables x_i and x_j, so that for all covariates $S \subseteq [d] \setminus \{i,j\}$ there exists a partition $A \cup B = S$ with*

$$v(A \cup i) + v(B \cup j) = v(S \cup \{i,j\}) \ .$$

Then, x_i and x_j do not occur together in the optimal partition Π^* in regards to Objective (1)*, i.e.*

$$\forall S_k \in \Pi^* : i \notin S_k \vee j \notin S_k \ .$$

Proof. Assume the optimal partition Π^* contains a set S where $i, j \in S$. Then, the value function $v(S)$ is decomposable into $v(S) = v(A \cup i) + v(B \cup j)$. Thus, we may construct a partition Π' with $A \cup i$ and $B \cup j$, where the reconstruction error $f(x) - \sum_{S_i \in \Pi'} v(S_i)$ remains the same and its regularization penalty shrinks, i.e. $R(\Pi^*) > R(\Pi')$. It follows that the overall objective of the partition Π' is lower than Π^*, contradicting its optimality. □

Theorem 1 confirms the intuition that if v is additive for two variables x_i and x_j, then they do not occur as part of the same set in the optimal partition. The main challenge lies in the exponential quantity of contexts S to consider, where the effect of x_i and x_j may differ. We show how we can reduce this effort to only a single test per pair x_i and x_j, which under a mild assumption allows to detect an absence of interaction, and hence rule out these being grouped in the optimal solution. The resulting explanations are tractable in real time and contain only significant interactions.

Assumption 1. *If v is additive for a partition A, B of S, i.e. $v(S) = v(A) + v(B)$, then it is also additive for all subsets $A' \subseteq A, B' \subseteq B$, so that*

$$\forall A' \subseteq A, B' \subseteq B : v(A') + v(B') = v(A' \cup B') \ .$$

Assumption 1 requires that the additivity of two sets of features A and B is preserved for all of their subsets. For example, if we find that $v(\{x_1, x_2, x_3\}) = v(\{x_1, x_2\}) + v(\{x_3\})$, then we also assume that $v(\{x_1, x_3\}) = v(\{x_1\}) + v(\{x_3\})$. This holds for many popular value functions, including the interventional value function by Janzing et al. [16] and the original observational value function used by Lundberg and Lee [25] in conjunction with an underlying additive function f, but does not generally hold for Asymmetric Shapley Values [7] (see Appx. B).

We now show how to determine when a pair of variables x_i and x_j is additive using the context dependent interaction $\mathcal{I}(i,j,S)$, and in particular the lack thereof.

Lemma 1. *Under Assumption 1, if the total interaction of a pair of variables i and j is not zero, i.e.*

$$\sum_{S \subseteq [d] \setminus \{i,j\}} \mathcal{I}(i,j,S) \neq 0 \ ,$$

then v is not additive for i and j.

Proof. If there is interaction between i and j, we show that there exists a covariate set S for which v is not additive for i and j. First, we note that

$$\sum_{S \subseteq [d] \setminus \{i,j\}} \mathcal{I}(i,j,S) \neq 0$$
$$\implies \exists S \subseteq [d] \setminus \{i,j\} : \mathcal{I}(i,j,S) \neq 0 \ ,$$

i.e. there exists a covariate set S for which the interaction is not zero. For this set S, it holds that

$$v(S \cup i) + v(S \cup j) \neq v(S \cup \{i,j\}) + v(S) \ . \tag{2}$$

If v indeed was additive for i and j, then for S there exists a partition $A \cup B = S$ so that

$$v(S \cup i, j) = v(A \cup i) + v(B \cup j) \ .$$

By Assumption 1, we know that this decomposition also holds for S, $S \cup i$ and $S \cup j$, so that we can rewrite Equation (3) as

$$v(A \cup i) + v(B) + v(A) + v(B \cup j)$$
$$\neq v(A \cup i) + v(B \cup j) + v(A) + v(B) \ .$$

This statement is a contradiction, and thus shows the claim that v is not additive for i and j. □

With Lemma 1, we show that we can reject the additivity of a pair of variables i and j if their pairwise Banzhaf interaction is non-zero [10]. As per Theorem 1, any pair of variables i and j that is additive is not grouped together in the optimal partition. Thus, to obtain the optimal partition Π^*, we need to consider all interacting pairs of variables i and j. Furthermore, let i and j be non-additive, and let j and k be non-additive too, then we can show that i and k are also non-additive, i.e. potentially grouped together in the optimal partition.

Lemma 2. *Let v be non-additive for i and j, and let v be non-additive for j and k. Then, v is also not additive for the variables i and k.*

Proof. Let v be non-additive for i and j, and let v be non-additive for j and k, i.e. there exists a set S_1 where

$$\forall A_1, B_1 : v(A_1 \cup i) + v(B_1 \cup j) \neq v(S_1 \cup \{i,j\})$$

and the set S_2 for j and k respectively. Now assume that v is additive for i and k, then it must hold for all covariate sets S_3 that

$$\forall S_3 \subseteq [d] \setminus \{i,k\} : \exists A, B : v(A \cup i) + v(B \cup k) = v(S_3 \cup \{i,k\}) \ .$$

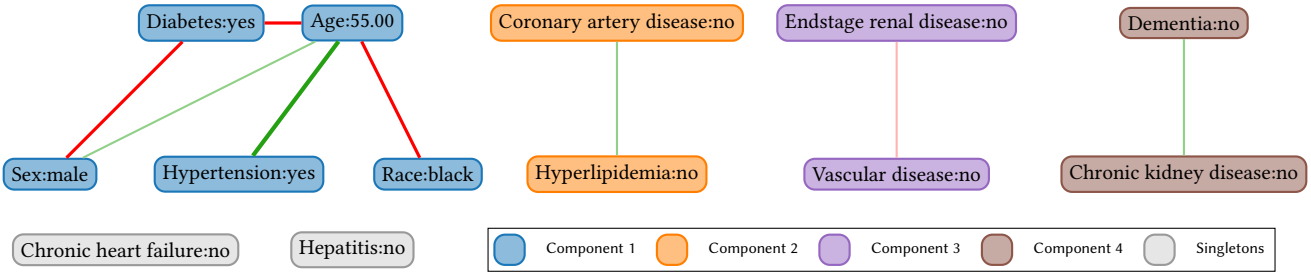

Figure 2: Interaction graph for a COVID-19 survival prediction. Green edges indicate a positive, red edges a negative interaction effect. In this example, the detrimental effect of diabetes and hypertension on survival is alleviated by the relatively young age (55) of the patient. iSHAP uses the connected components of the graph to guide the search for the optimal partition/explanation.

However, consider the set $S_3 = (S_1 \cup S_2 \cup \{j\}) \setminus \{i,k\}$. If v is additive with regard to i and k, there exists a partition A_3, B_3 so that

$$v(A_3 \cup i) + v(B_3 \cup k) = v(S_3 \cup \{i,k\}) \ .$$

Now, we distinguish between two cases: Let $j \in B_3$, then we can construct a new sub-partition $A_1 = S_1 \cap A_3$ and $B_1 = S_1 \cap B_3$. A_1 and B_1 are subsets of $A_3 \cup i$ and $B_3 \cup k$, so that by Assumption 1 additivity is preserved for A_1 and B_1. Therefore, it holds that

$$v(A_1 \cup i) + v((B_1 \setminus j) \cup j) = v(S_1 \cup \{i,j\}) \ ,$$

since $A_1 \cup B_1 = S_1$ as $S_1 \subseteq S_3 \cup k$. This contradicts the fact v is non-additive for i and j, and hence shows that v also is *not additive for i and k*. If $j \in A_3$, we similarly construct a partition $A_2 = S_2 \cap A_3$ and $B_2 = S_2 \cap B_3$, and from which too follows that v is not additive for i and k. □

Lemma 2 allows us to reject the additivity of a pair of variables i and j if they are connected by a chain of interactions. That is, it helps us to reduce the search space to partitions where its elements are connected components with regard to interactions. In practice, we run the risk of falsely eliminating pairs of variables for which the sum of non-additive interaction effects is zero. Our evaluation shows this happens only very rarely in practice.

2.3 Explanation

Once the optimal partition Π^* is obtained, we use the discovered interacting sets $S_i \in \Pi^*$ as building blocks to explain the algorithmic decision $f(x)$. We set the contribution e_i of each feature set S_i to the Shapley values of a new game v', in which each set S_i corresponds to a player. This new game v' allows to only include either all or no features of a feature set S_i, and takes the values of the original value function v. As a result, we return an additive explanation $\{(S_i, e_i)\}_{S_i \in \Pi^*}$, where $\sum_i e_i = f(x)$, based on the optimal partition of an algorithmic decision $f(x)$. Finally, we quantify the amount of interaction in S_i using Definition 1 and extend it onto sets as the difference between their joint contribution and the contribution of grouped features individually, $v(S_i) - \sum_{j \in S_i} v(j)$.

3 ALGORITHM

The number of partitions of a set of d variables is the Bell number, B_d. The amount grows super-exponentially with the number of variables d, making it vital to restrict the search space. To this end, we introduce the iSHAP algorithm for Interaction-aware Shapley Value Explanations. It enables us to find the optimal partition Π^* that minimizes the regularized reconstruction error of Objective 1.

3.1 Search Space Pruning

Based on Theorem 1, we prune suboptimal partitions by identifying all groups of features with significant interactions, and then remove all partitions that do not contain these groups. In particular, as per Lemma 1 we test for all pairs of variables i and j, if there is *any* interaction, i.e. of higher and lower order, through

$$\sum_{S \subseteq [d] \setminus \{i,j\}} \mathcal{I}(i,j,S) \neq 0 \ .$$

To exactly compute this quantity, exponentially many value functions $v(S)$ are required. Instead, we adopt a Monte Carlo approach by sampling uniformly at random coalitions S from $[d] \setminus \{i,j\}$ and computing the expectation of the interaction effect $E_S[\mathcal{I}(i,j,S)]$. This Kernel-SHAP like approach allows us to approximate the interaction effect with fewer samples and use a statistical test with significance level to reject the null hypothesis of no interaction.

For a pair x_i and x_j, we formulate the null-hypothesis as

$$H_0 : \mathrm{E}_S\left[\mathcal{I}(i,j,S)\right] = 0, \text{ where } S \sim \mathit{Uniform}\left(\mathcal{P}\left([d] \setminus \{i,j\}\right)\right) .$$

To this end, we rely on a standard t-test, which allows us to find statistically significant interaction effects given a significance level α. We construct an undirected graph with a node for each feature, and draw an edge wherever H_0 is rejected.

Fig. 2 shows an example of an interaction graph for COVID-19 survival prediction. Pairs with significant non-additive effect (i.e. null hypothesis rejected) are connected by an edge. For example, the positive interaction between `Age` and `Diabetes` indicates that the generally negative effect of diabetes on surviving COVID-19 is less pronounced for non-elderly people. (Note these only explain the interactions in the model, not necessarily those in the data.)

The so-obtained interaction graph is the base of the graph partitioning algorithm to optimize Objective (1). By Theorem 2, we know that any pair of nodes which is connected by a path in the interaction graph is not additive. Therefore, the optimal partition consists at most of *connected components* of the interaction graph, and their subsets. In this much reduced search space are contained only those partitions the sets show statistically significant interactions of any order, determined through our interaction test $\mathrm{E}_S\left[\mathcal{I}(i,j,S)\right] \neq 0$

that considers all possible contexts S. We provide the pseudocode of FindInteractions in Appx. C.

3.2 Partitioning Algorithm

Given the interaction graph we can derive all valid candidate partitions. A partition is valid if for each component S and all pairwise features $x_i, x_j \in S$ there exist a path between x_i and x_j in the interaction graph. Importantly, this does not mean the features have to interact directly, so that iShap can naturally detect higher order interactions, e.g. a three way XOR.

We propose two variants of iShap-Exact and iShap-Greedy aimed at small and large datasets, respectively. In iShap-Exact we test all valid partitions, and select the one which minimizes our objective. We evaluate all eligible partitions Π by sampling the value function $v(S)$ for each subset of features $S \in \Pi$, computing the reconstruction loss $L(\Pi) = (f(x) - \sum_{S\in\Pi} v(S))^2$ and adding the regularization penalty $\mathcal{R}(\Pi)$. To avoid recomputing the value function for the same subset many times, we additionally buffer the value function $v(S)$ for re-use. By searching over all valid partitions, we can guarantee to find the optimal partition Π^* that minimizes the objective function, with the downside that iShap-Exact still comes to its limit for many variables.

Hence, we introduce a greedy search variant iShap-Greedy, which is suitable for large datasets and fast inference. We show the pseudocode of iShap-Greedy in Algorithm 1. We start by computing the interaction graph G (line 1), and initialize the partition Π with the all singleton partition $\{S_i|S_i = \{i\}\}_{i=1}^d$ (line 3). Next, we start the search in a bottom-up approach where, iteratively, the two sets S_i and S_j that yield the highest gain in the objective are merged (line 7 - 12). Here, only merges are considered which do not violate the interaction graph, i.e. S_i and S_j are connected by a path (line 8). We continue this until no further merge improves the score (line 13). Finally, the Shapley values are computed for each set $S_i \in \Pi$ and the explanation of $f(x)$ is returned (line 17 - 19). Naturally, the greedy approach is not guaranteed to be optimal, but as we will see in the evaluation, achieves near-optimal results in practice. We explain iShap-Exact and give a complexity analysis of both variants in Appx. C.

Algorithm 1: iShap-Greedy $(f, x, \hat{X}, n_S, v, \alpha, \lambda)$

Input: Data point x, model f, sample $\hat{X}$, number of samples n_S, value function v, significance level α, regularizer parameter λ

1 $G \leftarrow$ FindInteractions$(f, x, \hat{X}, n_S, \alpha)$
2 d $\leftarrow$ number of nodes in G
3 $\Pi \leftarrow \{S_i|S_i = \{i\}\}_{i=1}^d$
4 **while** *True* **do**
5 Best Candidate Score $\leftarrow L(\Pi) + \lambda\mathcal{R}(\Pi)$
6 $\Pi* \leftarrow \Pi$
7 **for** $S_i, S_j \in \Pi$ **do**
8 **if** $\exists k \in S_i, l \in S_j : (k, l) \in G$ **then**
9 $\Pi' \leftarrow \Pi \cup \{S_i \cup S_j\} \setminus \{S_i, S_j\}$
10 **if** $L(\Pi') + \lambda\mathcal{R}(\Pi')$ <*Best Candidate Score* **then**
11 Best Candidate Score $\leftarrow L(\Pi') + \lambda\mathcal{R}(\Pi')$
12 $\Pi* \leftarrow \Pi'$
13 **if** $\Pi \neq \Pi*$ **then**
14 $\Pi \leftarrow \Pi*$
15 **else**
16 break
17 Value function $v'(S)$ of game with Π as players, $v'(S) = v(S)$ if $S \in \mathcal{P}(\Pi)$
18 Compute Shapley values ϕ_i for $i = 1, \ldots, m$ using $S_i \in \Pi$ as players using v'
19 **return** Explanation $\{(S_i, \phi_i)\}_{S_i\in\Pi}$, Interaction Graph G

4 RELATED WORK

We focus on post-hoc, model-agnostic explainability approaches [34] that treat the model f as a black-box and generate explanations by perturbing the input and analyzing the output. Here, explanations can be categorized into global explanations of f and local explanations of a particular decision $f(x)$.

Friedman [6] introduced the partial dependence plot (PD), a global explanation that visualizes the relationship between a variable X_i and the predicted output $f(X)$. PD plots are well suited for an injective relationship between X_i and $f(X)$, but not for cases with interaction effects between more than two variables. Functional ANOVA (analysis of variance) [14, 15] is another global explanation approach which aims at discovering non-additive interactions between input variables. Sivill and Flach [38] propose to discover interacting feature sets for a given model, one step further Herbinger et al. [12] aim to partition the feature space into subspaces by minimizing feature interactions. Overall, global model explanations can give a good overview over a model f, but produce non-conclusive explanations when faced with complex, highly interactive functions where it is hard to create a single summary.

Local explanations on the other hand aim to explain the decision of a model $f(x)$ for a particular instance x. Local Interpretable Model Agnostic Explanations (Lime) [35], is perhaps the most influential post-hoc explainability method. Lime explains a prediction $f(x)$ by constructing a local surrogate model f' that is interpretable, for example a Linear Regression or a Decision Tree. Lime is model agnostic and generates simple, intuitive explanations, but has a fuzzy data sampling process and no guarantees as it is based on purely heuristics. As Lime fits a local model, the resulting explanations can be misleading [36] or non informative for cases where x is *far* from any decision boundary.

A different style of explanation are approaches which explain a decision $f(x)$ through combinations of features x_i [2, 36]. In essence both method find a sufficient set of variables such that the prediction $f(x)$ does not change. In contrast to our method they can not provide explanations for regression models and they do not explain which combinations of variables has which effect on the prediction. Counterfactual explanations [20, 26, 32, 42, 43] are another type of local explanations that bring together causality and explainability. Algorithmic Recourse [17, 21] goes one step further and wants to find the best set of changes to reverse a models decision. These methods are most appropriate if the user seeks

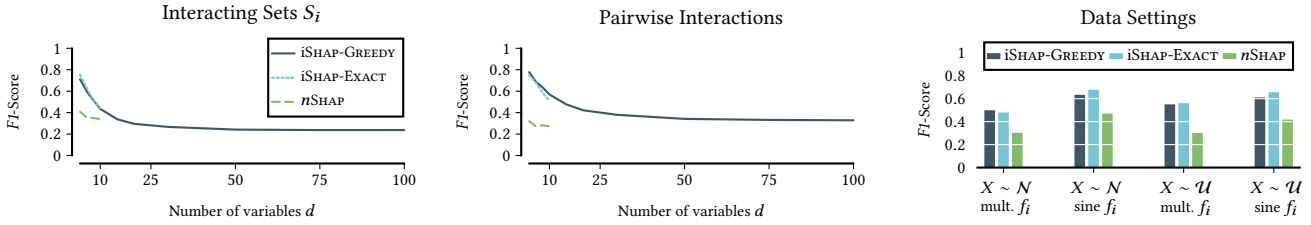

Figure 3: [Higher is better] Average *F1* score of recovered interactions in GAMs, in terms of exactly recovered sets of interacting features (3a) resp. pairwise interactions (3b). In Fig. 3c we show average performance up to 10 features. We observe that iShap outperforms *n*Shap on all function classes and iShap-Greedy performs closely to iShap-Exact.

actionable insights, i.e. how to reverse a decision, and, in contrast, do not explain a prediction.

One of the most widely used approaches for explainable AI are Shapley values [25]. Shapley values where originally introduced [37] in game theory to measure the contributions of individual players. Recently different variants have been proposed like asymmetric [7] and causal Shapley values [13]. Unlike our method they provide per feature attributions and no further insight into which features interact. Describing interaction is getting more attention in recent years. Jung et al. [19] propose to quantify the effect of a group of causes through *do*-interventions, but focus on estimating those effects from non-experimental data. Jullum et al. [18] propose to compute Shapley values on predefined feature sets, where they suggest to either group semantically related features or correlated features. Our approach discovers the feature sets automatically.

Most closely related is a recent class of so called *interaction index* explanations. The work by Lundberg et al. [24] introduces *SHAP interaction values* extending SHAP explanations to all pairwise interactions. Sundararajan et al. [39], Harris et al. [11] and Tsai et al. [41] both derive Shapley interaction indices for binary features that cover the entire power set. Most recently, Bordt and von Luxburg [1] introduced *n*Shap, which extends Shapley interaction values to feature sets up to degree n. Similar to our method, *n*Shap explains a decision $f(x)$ through a generalized additive model $\sum_{S \subset [d], |S| \leq n} \Phi_S^n$. The main difference is that *n*Shap provides a value for all of the power set of features, whereas our method chooses a succinct representation selecting interacting components.

5 EXPERIMENTS

In this section we empirically evaluate iShap. We compare it to Shap [25], Lime [35], and *n*Shap [1]. We implemented iShap in Python, we used the original Python implementation of Shap, Lime, and *n*Shap ($n = 4$). We allow each method up to 10 minutes per explanation. We provide the code and data generators in the Supplementary Material[1]. All experiments were conducted on a consumer-grade laptop.

5.1 Discovering Interactions

First, we examine whether iShap recovers truly interacting sets of variables. To this end we generate generalized additive models f (GAMs) for which we determine the ground truth sets of interacting features S_i. We sample d feature variables X_j, either from a normal distribution: $P(X_j) = N(\mu, \sigma^2), \mu \in [0, 3], \sigma \in [0.5, 1.5]$ or a uniform distribution: $X_j \in \mathcal{U}(0, 3)$. We construct the ground truth partition Π by sampling sets S_i of arbitrary size from a Poisson distribution. Next, we define f as $f(x; \Pi) = \sum_{S_i \in \Pi} f_i(X_{S_i})$, with a non-additive inner function f_i, where we consider

(1) $f_i\left(X_{S_i}\right) = \prod_{j \in S_i} a_{i,j} \cdot X_j$, $a_{i,j} \in \pm[0.5, 1.5]$

(2) $f_i\left(X_{S_i}\right) = \sin\left(\sum_{j \in S_i} a_{i,j} \cdot X_j\right)$, $a_{i,j} \in \pm[0.5, 1.5]$.

We generate data with all combinations of inner function and feature distribution, that is *multiplicative inner function* (1) and normal distribution ($X \sim \mathcal{N}$), *sine inner function* (2) and ($X \sim \mathcal{N}$), etc. For more details we refer to Appx. D.

We compute the iShap and the *n*Shap explanations for a random data point x. For *n*Shap we construct a $\hat{\Pi}$ by iteratively taking the strongest interacting set without overlap to already chosen sets, until all features are covered. We measure how well the explanations $\hat{\Pi}$ compare to the ground truth Π by the *F1* score between the sets using as precision (*Pr.*) and recall (*Re.*)

$$Re.(\Pi, \hat{\Pi}) = \frac{1}{|\Pi|} \sum_{\hat{S}_i \in \hat{\Pi}} \mathbb{1}(\hat{S}_i \in \Pi) \ , Pr.(\Pi, \hat{\Pi}) = \frac{1}{|\hat{\Pi}|} \sum_{S_i \in \Pi} \mathbb{1}(S_i \in \hat{\Pi}) \ .$$

In addition, we compute the F1-score on pairs $j, k \in S_i$ and $j', k' \in \hat{S}_i$ to assess how accurately pairwise interactions are recovered.

We show the performance of iShap-Greedy, iShap-Exact and *n*Shap in Fig. 3, in (3a) and (3b) we report the average over all four settings, in (3c) we show average $F1$ score for each function class up to 10 variables, *n*Shap times out for settings with more than 10 variables. We see that iShap outperforms *n*Shap both in terms of retrieving interacting sets (Fig. 3a) and pairwise interactions (Fig. 3b), and that this advantage persists across all tested combinations of feature distributions $P(X)$ and classes of inner functions (Fig. 3c). More interestingly, we see that iShap-Greedy performs almost as well as iShap-Exact, showing the effectiveness of our interaction test in restricting the search space.

5.2 Surrogate Model Accuracy

Next, we evaluate how well the iShap, Shap, *n*Shap and Lime explanations can serve as *surrogate models*. Given a model f, Poursabzi-Sangdeh et al. [31] proposed to present a user with an explanation for a data point x, and ask them to use this to predict the output

[1] https://github.com/Schascha1/iSHAP

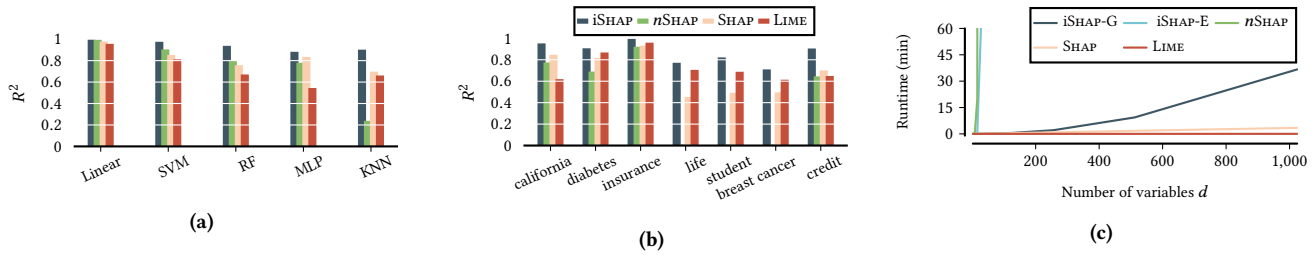

Figure 4: R^2 for surrogate models of iSHAP, *n*SHAP, SHAP and LIME across different model classes (a) and datasets (b). iSHAP provides the most accurate surrogate model across all classes and datasets, whilst scaling to more dimensions than *n*SHAP (c).

of the model on new, unseen data. We adopt this approach and test each method using seven datasets[2] with increasingly complex, non-additive models f: linear models, SVMs, random forests (RF), multi-layer perceptrons (MLP) and k-nearest neighbors (KNN).

We start by selecting two data points, denoted as $x^{(1)}$ and $x^{(2)}$. We then create a new data point, x', by randomly selecting features from either $x^{(1)}$ or $x^{(2)}$ with equal probability. For each feature x'_i, or sets of features in case of iSHAP and *n*SHAP, we take its resp. additive contribution, and sum them up to obtain the implied prediction $\hat{f}(x')$. We then calculate the mean squared error between this implied prediction $\hat{f}(x')$ and the true prediction $f(x')$ and report the overall R^2 in Fig. 4.

In Figure 4b we show R^2 of different model, averaged over all datasets. For linear models f the additivity assumption holds fully, which is reflected in the near perfect accuracy of all methods. When using SVMs and random forests, the overall accuracy of all methods is decreased. Here, iSHAP emerges as the most accurate surrogate model with an R^2 over 0.9. LIME struggles to model the local decision surface of an MLP as a linear model, and *n*SHAP struggles with k-nearest neighbors. SHAP does not take into account any interactions.

From the performance across different datasets, shown in Fig. 4b, we see that iSHAP outperforms all other methods. On datasets with more features and interactions between these, such as the *Credit* dataset, we find that iSHAP achieves a R^2 score of 0.9, while SHAP, *n*SHAP and LIME obtain R^2 scores of 0.6-0.7. On the *Life Expectancy*, *Student* and *Breast Cancer* datasets where SHAP and LIME struggle and *n*SHAP times out, iSHAP provides by far the best surrogate models. This increase in performance comes with an increased computational effort for iSHAP.Still, in contrast to *n*SHAP's limit of 16 features, iSHAP-EXACT can explain up to 32 variables within an hour, and iSHAP-GREEDY scales up to hundreds of features (Fig. 4c).

5.3 Case Study: Covid-19

Next, we conduct a qualitative comparison between the Shapley value based explanations provided by iSHAP, SHAP and *n*SHAP. For this, we consider a Covid-19 dataset containing survival data of 1,540 hospitalized patients [22]. We train a random forest classifier to predict the likelihood of survival, based on diverse biomarkers such as age and pre-existing conditions. For each patient we provide the respective iSHAP, SHAP and *n*SHAP explanation in the Supplementary Material. We show in Table 1 the SHAP (left), *n*SHAP (middle) and iSHAP (right) explanations for a patient which was hospitalized, and for which the model correctly predicted survival.

We discuss the explanations in turn. The features that SHAP identifies as key to survival seem to be reasonable at first glance, but upon closer inspection are at least partly counterintuitive. `Hypertension:1` is marked as a positive factor and `Diabetes:1` as having only a slight negative effect, despite both are known risk factors for Covid-19 [28].

To obtain further insight, we move on to the *n*SHAP explanation shown in the middle. We show the top 13 out of 2 516 interaction coefficients, the actual values of which we provide in Supplementary Material. Like SHAP, *n*SHAP also identifies `Age` as a positive factor, but additionally shows that it is included in many higher-order sets. The amount of redundancy and inconclusive values makes a clear interpretation of them hard, for example (`Age`, `Hypertension`, `Diabetes`, `Race`) is given a contribution of -9% to survival chances, while (`Age`, `Hypertension`) alone improve odds by 7.8% supposedly.

Thus, we inspect the iSHAP explanation shown on the right. It partitions the feature set into six parts, and clearly identifies there are two main interacting sets to consider. Firstly the combination of `Age:55` in conjunction with `Hypertension:1` and `Diabetes:1` is marked with a strong positive effect and is explained as follows: While diabetes and hypertension are negative marginal factors for survival across the entire data set, their effect is significantly reduced for a patient of only 55 years old compared to the on-average much older patients in the dataset. The high SHAP value of the feature `Age:55` reflects this as it is the *sum* of the marginal and the interaction effects. Second, we see that `Hyperlipidemia:0` and `Coronary Artery Disease:0` are positively interacting factors for survival. `Coronary artery disease` is known to be a risk factor for Covid-19 patients [40], and thus its absence is positive. Hyperlipidemia is strongly associated with CAD [8], and its absence validates the `CAD:0` feature, thus interacting positively with it.

Overall, we see that all three explanations describe the same phenomena, but do so in different levels of detail. The SHAP explanation is arguably the most compact, but also the least detailed as it cannot explain the interactions that are important to understand the underlying mechanisms. The *n*SHAP explanation is arguably the most detailed, but, also the hardest to interpret, as it lists all interaction coefficients. The iSHAP explanation offers the best of both worlds: it is as interpretable as SHAP, and includes the main interactions as found by *n*SHAP, so to succinctly explain the models decision and making the user aware of the most important interactions.

[2] *California* [27], *Diabetes* [29], *Insurance* [23], *Life* [33], *Student* [3] and *Credit* [4]

(a) SHAP values

Feature	Effect (%)
age:55	28.1
race:black	0.9
sex:male	-0.1
hypertension:1	7.1
hyperlipidemia:0	-0.2
diabetes:1	-3.6
CAD:0	7.3
chf:0	0.0
CeVD:0	0.0

(b) Top-k *n*SHAP values (2516 total)

Feature Set	Effect (%)
age	15.1
age, hypertension, race, diabetes	-9.0
age, hypertension	7.8
age, race, hypertension	7.5
age, race, diabetes	6.5
age, diabetes	4.8
diabetes	-4.7
CAD	4.7
hypertension, diabetes	4.5
sex	-4.3
age, race, diabetes, CAD	4.2
age,race	-4.1
age, CAD, hyperlipidemia	3.6
⋮	⋮

(c) iSHAP values

Feature Set	Individual Effect+ Interaction Effect(%)
diabetes:1, age:55, hypertension:1	10.2 + 18.6
hyperlipidemia:0, CAD:0	4.5 + 1.9
sex	-3.5 + 0
race	4.0 + 0
copd	0.2 + 0
chf	0.0 + 0

Table 1: Explanations for predicted Covid-19 survival. In (a) we show feature-wise SHAP values, in (b) *n*SHAP values for all feature subsets, and in (c) iSHAP values, attributed to partitioned features.

5.4 User Study

Lastly, we evaluate the perception of human users on the explanations provided by iSHAP, SHAP and *n*SHAP. We conducted a user study with 24 participants, who were shown explanations for a Covid-19 patient as in Table 1. Their task mirrored the simulation experiment from Section 5.2, where given two explanations the tasks is to infer the models output for a mixed data point x'. We provide the survey handed out to participants in the Supplementary Material. Afterwards, we asked all participants to rank all methods in terms of interpretability, trustworthiness and reasonability of the explanation and give general feedback.

In terms of accuracy, on average predictions for SHAP were off by 8%, for nSHAP by 6% and for iSHAP by 5%. As only one instance per method was evaluated, we refrain from drawing any conclusions. We show the results of the post-study survey in Fig. 5. In terms of interpretability, 12 participants preferred SHAP, whilst 11 preferred iSHAP and only 1 preferred *n*SHAP, which a general preference for less detailed, more succinct explanations. On the aspect of trustworthiness, the majority of votes were cast for iSHAP, with feedback citing iSHAP's reporting of interactions in an interpretable manner as the deciding factor. This sentiment is also reflected in the perceived reasonability of the explanations, where 16 participants preferred iSHAP, compared to 8 preferences of *n*SHAP. Interestingly so, SHAP was not chosen once due to the lack of interactions.

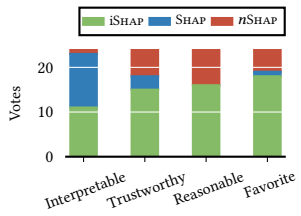

Figure 5: User study.

Lastly, when asked which method should be presented to a doctor, the overall highest vote went to iSHAP with 18 votes. Overall, the response by the participants suggests that our idea and execution of succinct, interaction-aware explanations resonates well with many users and provides a valuable approach to post-hoc decision explanation in addition to existing methods.

6 LIMITATIONS

The main trade-off of the succinctness of iSHAP explanations is their restriction to a single interaction set per feature in the partition Π. This limitation was raised multiple times in the feedback of the user study we conducted, where participants were concerned that the explanation might miss important interactions. This was offset by the succinctness and therewith interpretability of the explanation, which stood in contrast to the more complex explanations by *n*SHAP.

Assumption 1 holds for truly additive sets of features of a function f, but does not extend to those sets which are non-additive. This means that iSHAP's power to rule out interactions is reduced for highly interactive functions f. However, whilst a function f fitted through a ensemble of models or a deep learning model is rarely completely additive for a combination of features, we observe that the actual interaction is not statistically significant and thus allows the test to nevertheless prune it.

Lastly, the impact of sample size to estimate individual value functions $v(S)$ is another potential limitation for all Shapley value based methods, including SHAP, *n*SHAP, and iSHAP. It is not clear yet, what sample size is appropriate to estimate the value function $v(S)$ accurately. Furthermore, exact computation requires exponentially many value function $v(S)$, which is infeasible for large d. Our sampling based approach empirically performs well, but further investigations into the importance of sample size on the quality of explanations, for all methods, are necessary.

7 CONCLUSION

In this paper, we proposed a model agnostic, post-hoc explanation method. In contrast to existing explanations, we directly integrate significant interactions between sets of features x_{S_i} into a succinct, additive explanation. We showed how to use a statistical test to guaranteed find the underlying optimal partitioning of features and avoid fitting spurious interactions. Our algorithm iSHAP is an effective and fast procedure that takes the theoretical results into practice. On synthetic data we have shown that iSHAP returns accurate, ground truth interactions, and on real world data, we find that iSHAP is a more accurate surrogate than the state-of-the-art.

For future work, we plan to extend iSHAP to allow multiple interactions per variable. Furthermore, we want to extend the definition of interaction to be able to differentiate between any distribution of interaction effects.

REFERENCES

[1] Sebastian Bordt and Ulrike von Luxburg. 2023. From shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 709–745.

[2] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. 2019. What made you do this? understanding black-box decisions with sufficient input subsets. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 567–576.

[3] Paulo Cortez and Alice Maria Gonçalves Silva. 2008. Using data mining to predict secondary school student performance. (2008).

[4] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[5] Hadi Fanaee-T and Joao Gama. 2013. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* (2013), 1–15. https://doi.org/10.1007/s13748-013-0040-3

[6] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* (2001), 1189–1232.

[7] Christopher Frye, Colin Rowat, and Ilya Feige. 2020. Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems* 33 (2020), 1229–1239.

[8] Joseph L Goldstein, William R Hazzard, Helmut G Schrott, Edwin L Bierman, Arno G Motulsky, et al. 1973. Hyperlipidemia in coronary heart disease I. Lipid levels in 500 survivors of myocardial infarction. *The Journal of Clinical Investigation* 52, 7 (1973), 1533–1543.

[9] Alicja Gosiewska and Przemyslaw Biecek. 2019. Do not trust additive explanations. *arXiv preprint arXiv:1903.11420* (2019).

[10] Michel Grabisch and Marc Roubens. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory* 28 (1999), 547–565.

[11] C Harris, Richard Pymar, and Colin Rowat. 2022. Joint Shapley Values: a measure of joint feature importance. In *International Conference on Learning Representations*.

[12] Julia Herbinger, Bernd Bischl, and Giuseppe Casalicchio. 2023. Decomposing Global Feature Effects Based on Feature Interactions. *arXiv preprint arXiv:2306.00541* (2023).

[13] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. 2020. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in Neural Information Processing Systems* 33 (2020), 4778–4789.

[14] Giles Hooker. 2004. Discovering additive structure in black box functions. In *Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Seattle, WA*. 575–580.

[15] Giles Hooker. 2007. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* 16, 3 (2007), 709–732.

[16] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. 2020. Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2907–2916.

[17] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615* (2019).

[18] Martin Jullum, Annabelle Redelmeier, and Kjersti Aas. 2021. groupShapley: Efficient prediction explanation with Shapley values for feature groups. *arXiv preprint arXiv:2106.12228* (2021).

[19] Yonghan Jung, Shiva Kasiviswanathan, Jin Tian, Dominik Janzing, Patrick Blöbaum, and Elias Bareinboim. 2022. On measuring causal contributions via do-interventions. In *International Conference on Machine Learning*. PMLR, 10476–10501.

[20] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 895–905.

[21] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 353–362.

[22] Ben Lambert, Isaac J Stopard, Amir Momeni-Boroujeni, Rachelle Mendoza, and Alejandro Zuretti. 2022. Using patient biomarker time series to determine mortality risk in hospitalised COVID-19 patients: a comparative analysis across two New York hospitals. *Plos one* 17, 8 (2022), e0272442.

[23] Brett Lantz. 2019. *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd.

[24] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* 2, 1 (2020), 56–67.

[25] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30 (2017).

[26] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[27] R Kelley Pace and Ronald Barry. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters* 33, 3 (1997), 291–297.

[28] Rizwana Parveen, Nouroz Sehar, Ram Bajpai, and Nidhi Bharal Agarwal. 2020. Association of diabetes and hypertension with disease severity in covid-19 patients: A systematic literature review and exploratory meta-analysis. *Diabetes research and clinical practice* 166 (2020), 108295.

[29] Pedregosa et al. 2011. Scikit-learn: Machine Learning in Python. Dataset available at https://scikit-learn.org/stable/datasets.html.

[30] David Pisinger. 1999. Linear time algorithms for knapsack problems with bounded weights. *Journal of Algorithms* 33, 1 (1999), 1–14.

[31] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. 2021. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–52.

[32] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 344–350.

[33] Kumar Rajarshi, Deeksha Russell, and Duan Wang. 2019. Life Expectancy (WHO). https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who. Kaggle: Online Data Science Community.

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-Agnostic Interpretability of Machine Learning. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*.

[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Francisco, CA*. 1135–1144.

[36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[37] Lloyd S Shapley. 1953. A value for n-person games. In *Contributions to the Theory of Games (AM-28)*. Princeton University Press, 307–317.

[38] Torty Sivill and Peter Flach. 2023. Shapley Sets: Feature Attribution via Recursive Function Decomposition. *arXiv preprint arXiv:2307.01777* (2023).

[39] Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. 2020. The shapley taylor interaction index. In *International conference on machine learning*. PMLR, 9259–9268.

[40] Lukasz Szarpak, Malgorzata Mierzejewska, Jonasz Jurek, Anna Kochanowska, Aleksandra Gasecka, Zenon Truszewski, Michal Pruc, Natasza Blek, Zubaid Rafique, Krzysztof J Filipiak, et al. 2022. Effect of Coronary Artery Disease on COVID-19—Prognosis and Risk Assessment: A Systematic Review and Meta-Analysis. *Biology* 11, 2 (2022), 221.

[41] Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. 2023. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research* 24, 94 (2023), 1–42.

[42] Arnaud Van Looveren and Janis Klaise. 2021. Interpretable counterfactual explanations guided by prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*. Springer, 650–665.

[43] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

A PROOFS

A.1 Proof of Theorem 1

Proof. Assume the optimal partition Π^* contains a set S where $i, j \in S$. Then, the value function $v(S)$ is decomposable into

$$v(S) = v(A \cup i) + v(B \cup j) \ .$$

Thus, we may construct a partition Π' with $A \cup i$ and $B \cup j$. Let $\mathrm{E}\left[f(X)\right] = 0$ (achievable by pre-processing), then the reconstruction error of Π' is

$$f(x) - \sum_{T' \in \Pi'} v(T') = \left(f(x) - \sum_{T \in \Pi^*} v(T)\right) - v(S) + v(A \cup i) + v(B \cup j) \ .$$

As per the assumption, $v(S) = v(A \cup i) + v(B \cup j)$, so the reconstruction error of Π' is equal to the reconstruction error of Π^*. For the regularization penalty of the new partition elements $A' = A \cup i$ and $B' = B \cup j$ in Π' it holds that

$$|A'|(|A'|-1)/2 + |B'|(|B'|-1)/2 < (|A'|+|B'|)(|A'|+|B'|-1)/2 \ ,$$

which shows that it is smaller than the regularization penalty of Π^* for $S \cup i, j$. Thus, the overall objective of Π' is lower than Π^*, contradicting its optimality. □

A.2 Proof of Lemma 1

Proof. If there is interaction between i and j, we show that there exists a covariate set S for which v is not additive for i and j. First, we note that

$$\begin{aligned} &\sum_{S \subseteq [d] \setminus \{i,j\}} \mathcal{I}(i,j,S) \neq 0 \\ \implies &\exists S \subseteq [d] \setminus \{i,j\} : \mathcal{I}(i,j,S) \neq 0 \ , \end{aligned}$$

i.e. there exists a covariate set S for which the interaction is not zero. For this set S, it holds that

$$v(S \cup i) + v(S \cup j) \neq v(S \cup i, j) + v(S) \ . \tag{3}$$

If v indeed was additive for i and j, then for S there exists a partition $A \cup B = S$ so that

$$v(S \cup i, j) = v(A \cup i) + v(B \cup j) \ .$$

By Assumption 1, we know that this decomposition also holds for S, $S \cup i$ and $S \cup j$, so that we can rewrite Equation (3) as

$$\begin{aligned} &v(A \cup i) + v(B) + v(A) + v(B \cup j) \\ \neq\ &v(A \cup i) + v(B \cup j) + v(A) + v(B) \ . \end{aligned}$$

This statement is a contradiction, and thus proves that v is not additive for i and j. □

A.3 Proof of Lemma 2

Proof. Assume that v is additive for i and k, i.e. $\forall S_3 \subseteq [d] \setminus \{i,k\} : \exists A, B : v(A \cup i) + v(B \cup k) = v(S_3 \cup i, k)$.

Now consider a set S_1 for which v is not additive in regards to i and j, i.e. $\forall A_1, B_1 : v(A_1 \cup i) + v(B_1 \cup j) \neq v(S_1 \cup i, j)$ and the set S_2 for j and k respectively. We construct a set $S_3 = (S_1 \cup S_2 \cup \{j\}) \setminus \{i,k\}$, for which v now has to be additive with regard to i and k, i.e. there exists a partition A_3, B_3 so that $v(A_3 \cup i) + v(B_3 \cup k) = v(S_3 \cup i, k)$.

There are two cases to consider, either $j \in A_3$ or $j \in B_3$. Let $j \in B_3$, then we can construct a new sub-partition $A_1 = S_1 \cap A_3$ and $B_1 = S_1 \cap (B_3 \cup k)$. A_1 and B_1 are subsets of $A_3 \cup i$ and $B_3 \cup k$, so that by Assumption 1 additivity is preserved for A_1 and B_1. Therefore, it holds that $v(A_1 \cup i) + v((B_1 \setminus j) \cup j) = v(S_1 \cup i, j)$, since $A_1 \cup B_1 = S_1$ as $S_1 \subseteq S_3 \cup k$. This contradicts the assumption that v is non-additive for i and j.

Similarly, we can show that $j \in A_3$ violates the assumption that v is non-additive for j and k, and conclude that v must in fact be non-additive for i and k. In the interaction graph, this allows us to reject the additivity of a pair of variables i and j if they are connected by a path, and justifies the use of connected components over cliques as search space. □

B ADDITIVITY OF VALUE FUNCTIONS

We consider two value functions: the observational value function

$$v(S; f, x) = \mathrm{E}\ \left[f(X) | X_S = x_S\right]$$

by [25], where it is assumed that all variables are independent of each other, i.e. $\forall i \neq j : X_i \perp X_j$, and the interventional value function

$$v(S; f, x) = \mathrm{E}\ \left[f(X') | do(X'_S = x_S)\right]$$

by [16], where we consider as features variables the model inputs X'_i that are purely determined by the real world counterpart X_i. Hence, intervening on the model input as $do(X'_S = x_S)$ only has an effect on the input X'_S.

Let f now be additive for two sets A and B, so that $f(x) = g(x_A) + h(x_B)$, then the value function $v(A; f, x) + v(B; f, x)$ is transformed into

$$\begin{aligned}&\mathrm{E}\ [f(X)|X_A = x_A] + \mathrm{E}\ [f(X)|X_B = x_B]\\ =&\mathrm{E}\ [g(X)|X_A = x_A] + \mathrm{E}\ [h(X)|X_A = x_A] + \mathrm{E}\ [g(X)|X_B = x_B] + \mathrm{E}\ [h(X)|X_B = x_B]\ .\end{aligned}$$

We can drop the conditioning of $X_A = x_A$ where there is only h and vice versa for $X_B = x_B$ and g. This is possible both with the independence assumption in the observational Shapley values by [25], and the causal model as postulated by [16]. This leaves us with

$$\begin{aligned}&\mathrm{E}\ [g(X)|X_A = x_A] + \mathrm{E}\ [h(X)] + \mathrm{E}\ [g(X)] + \mathrm{E}\ [h(X)|X_B = x_B]\\ &\qquad = \mathrm{E}\ [g(X)|X_A = x_A] + \mathrm{E}\ [h(X)|X_B = x_B] + \mu\ .\end{aligned}$$

By convention, we preprocess $f(X)$ so that $\mu = \mathrm{E}\,[f(X)] = 0$, and μ hence can be dropped. Now, we similarly decompose $v(S; f, x)$ as

$$\mathrm{E}\,[f(X)|X_S = x_S] = \mathrm{E}\,[g(X) + h(X)|X_S = x_S] = \mathrm{E}\,[g(X)|X_S = x_S] + \mathrm{E}\,[h(X)|X_S = x_S]\ .$$

Now we again drop X_B from g and X_A from h and are left with

$$\mathrm{E}\,[g(X)|X_A = x_A] + \mathrm{E}\,[h(X)|X_B = x_B]\ ,$$

which shows the additivity for any set A and B for which f is decomposable.

C ALGORITHM

iShap consists of two main subroutines: find_interactions and find_partition. The first subroutine is the same for both the greedy and the optimal algorithm. iShap-Greedy uses a greedy, bottom-up approach to find the best partition, while iShap-Exact uses a exhaustive search over all valid partitions from the interaction graph.

find_interactions. As input, find_interactions receives a data point x, a model f and a sample $\hat{X}$ of $P(X)$. It returns all pairwise interactions between features that are statistically significant, encoded as a graph G. We initialize the interaction graph G with d nodes, where each node represents a single feature. We then sample n_s new data points $x^{(j)'}$ from the empiric data distribution, either marginally, conditionally or interventional as required by the value function v. For each data point $x^{(j)'}$, we sample a random intervention $z \in \{0,1\}^d$ with $p = 0.5$. We then intervene on the i-th feature on the j-th data point $x^{(j)'}$, i.e. $x_i^{(j)'} = x_i$, if $z_i^{(j)} = 1$.

Now, for each pair of features i, j, we test the hypothesis

$$H_0 : \sum_{S\subseteq[d]\setminus\{i,j\}} v(S\cup i) - v(S) + v(S\cup j) - v(S) = \sum_{S\subseteq[d]\setminus\{i,j\}} v(S\cup i, j) - v(S)$$

by taking dividing up the sample $\{f(x^{(j)'})\}$ into four subsets according to the intervention $z^{(j)}$:

- $v(S\cup\{i,j\}) \leftarrow \{f(x^{(j)'})|i, j \in z^{(j)}\ \}$
- $v(S\cup\{i\}) \leftarrow \{f(x^{(j)'})|i \in z^{(j)}, j \notin z^{(j)}\ \}$
- $v(S\cup\{j\}) \leftarrow \{f(x^{(j)'})|j \in z^{(j)}, i \notin z^{(j)}\ \}$
- $v(S) \leftarrow \{f(x^{(j)'})|i, j \notin z^{(j)}\ \}$

Now, we can test the hypothesis using a two-sided t-test with significance level α with unequal variances, also known as Welch's t-test. If the hypothesis is rejected, we add an edge between the nodes i and j to the graph G. By splitting up the sample into four subsets, we can test the hypothesis for each pair of features i, j on the same sample, which is more efficient than testing each pair on a separate sample as it reduces the number of required samples and thus evaluations of f by the amount of pairwise interactions, i.e. $O(d^2)$.

C.1 Complexity

The complexity of the greedy search is cubic. For a fully connected graph we have to evaluate $d(d-1)/2$ merges, where we can take at most d steps before arriving at the complete set $[d]$. In each step, we have to estimate the value function v with n samples, whose complexity we denote by $O(v(n,d))$. Hence, its complexity is $O(d^3)O(v(n,d))$.

find_partition. The find_partition subroutine takes in addition the graph G from find_interactions and uses the same data point x, the model f and the sample $\hat{X}$ of $P(X)$. find_partition returns the best scored partition Π in regards to Objective 1.

For the greedy approach, we initialize Π with d singleton sets, i.e. $\Pi = \{S_i|S_i = \{i\}\}_{i=1}^d$. We merge all eligible pairs of sets $S_i, S_j \in \Pi$ into a new set $S_i \cup S_j$ and score the new partition Π'. Eligibility is given if the graph G contains an edge between an element of S_i and an element of S_j. Each step, we merge the pair of sets that yields the best score and terminate once no more improvement is possible.

The exhaustive approach is a brute-force search over all possible partitions Π. We restrict the search space by only considering partitions Π, were all elements $S_i \in \Pi$ are connected in the graph G. Then, we score each partition Π and return the best scored partition.

Explanation. Once we have found the best scored partition Π, we can explain the prediction $f(x)$ by computing the Shapley values for a new game v' which has as its players the elements of Π instead of the features $\{1, \ldots, d\}$. The value function v' is defined only for those sets S which are elements of the power set of Π. On that set, the value function v' is defined as $v'(S) = v(S)$ as the underlying model f is the same.

On this new game, we can compute the Shapley values ϕ_i for each element $S_i \in \Pi$. The Shapley value ϕ_i is the average marginal contribution of the element S_i to the value of the game v'. iSHAP returns the Shapley values ϕ_i as the explanation for the prediction $f(x)$, in addition to the interaction graph G and the partition Π.

D EXPERIMENTS

D.1 Interaction Experiments

We first verify whether iSHAP accurately recovers the correct sets of variables for a generalized additive model. To this end, we generate a random function f over $d \in \{4, 6, 8, 10, 15, 20, 30, 50, 75, 100\}$ variables. To obtain a function with additive components, we partition the variables into sets S_i, where we iteratively sample the size of each set from a Poisson distribution with $\lambda = 1.5$, over which we define function $f(X) = \sum_{S_i \in \Pi} f_i(X_{S_i})$, whereas inner functions f_i we consider

- $f_i\left(X_{S_i}\right) = \prod_{j \in S_i} a_{i,j} \cdot X_j$, $a_{i,j} \in \pm[0.5, 1.5]$
- $f_i\left(X_{S_i}\right) = \sin\left(\sum_{j \in S_i} a_{i,j} \cdot X_j\right)$, $a_{i,j} \in \pm[0.5, 1.5]$

We sample all the underlying d variables either from a normal distribution: $P(X_j) = N(\mu, \sigma^2)$, $\mu \in [0, 3]$, $\sigma \in [0.5, 1.5]$ or a uniform distribution: $X_j \in \mathcal{U}(0, 3)$. In total, for we test the accuracy of 100 explanations for each combination of d, inner function and sampling distribution. Each time, we sample a random function f and dataset X of 10 000 points. We use the observational value function $v(S) = E[f(X)|X_S = x_S]$, where we sample X_i individually as they are independently generated. For a random $x \in X$, we generate the partition $\hat{P}$ with iSHAP, using as significance level $\alpha = 0.01$ and as regularization coefficient $\lambda = 5\text{e}{-3}$.

D.2 Accuracy Experiments

We consider five regression and two classification dataset:*California* [27], *Diabetes* [29], *Insurance* [23], *Life* [33], *Student* [3] with increasingly complex, non-additive models f: linear models, support vector machines (SVM), random forests (RF), multi-layer perceptrons (MLP) and k-nearest neighbors (KNN). This experiment is for each dataset repeated 100 times and goes as follows: we pick two instances $x^{(1)}$ and $x^{(2)}$ from a dataset. We use these to construct a we construct a new data point out of, randomly selected, $x^{(1)}$ and $x^{(2)}$, e.g. $x' = x_1^{(2)}, x_2^{(1)}, x_3^{(2)}, x_4^{(2)}, x_5^{(1)}$, by randomly selecting features from $x^{(1)}$ and $x^{(2)}$, where $x^{(1)} = x_1^{(1)}, \ldots, x_5^{(1)}$ and $x^{(2)} = x_1^{(2)}, \ldots, x_5^{(2)}$. To construct a surrogate prediction of SHAP, we compute the Shapley values for $x^{(1)}$ and $x^{(2)}$ and then compute the surrogate prediction by taking the sum over the respective feature importance values. For the example above, that is:

$$\phi_1^{(2)} + \phi_2^{(1)} + \phi_3^{(2)} + \phi_4^{(1)} + \phi_5^{(1)}$$

By design, this scheme is not applicable for all random instances with partitions $\Pi^{(k)}$ and $\Pi^{(l)}$, in which case simply resample until we have an admissible input. We compute the implied prediction for the new data point x' in the following way for each method:

- iSHAP: $f(x') = \sum_{S_i \in \Pi_1, S_i \subset I_1} e_i^{(1)} + \sum_{S_j \in \Pi_2, S_j \subseteq I_2} e_j^{(2)}$
- SHAP: $f(x') = \sum_{i \in I_1} \phi_i^{(1)} + \sum_{j \in I_2} \phi_j^{(2)}$
- *n*SHAP: $f(x') = \sum_{S \subseteq I_1} \Phi_S^{(1)} + \sum_{S \subseteq I_2} \Phi_S^{(2)}$
- LIME: For each datapoint we obtain a surrogate model $\hat{f}_1$ and $\hat{f}_2$. We take $\hat{f}_1(x')$ and $\hat{f}_2(x')$ and use whichever is closer to $f(x')$.

E USER STUDY

We conducted a user study with a total of 24 test subjects. Of these 24 participants, there were 5 female, 3 diverse and 16 male participants. The majority of participants were university students (23) and had some degree of technical knowledge in machine learning (21).

The study was self-supervised with an estimated duration of 30 minutes. The participants were distributed a document via Google Forms that included all instructions and tasks. The document is provided in the Supplementary Material[3].

The participants had to consider a total of 3 cases with SHAP, iSHAP and *n*SHAP explanations respectively, disguised as separate XAI models for a Covid-19 prediction task. For each method, we provided two explanations and asked the participants to infer the models prediction for a third patient who is a mixture of the two patients in the explanations. We asked the participants to evaluate the explanations in terms of informativeness, trustworthiness, understandability, and overall preference. The participants were also asked to provide feedback on the explanations and the study itself. We provide the spreadsheet containing all answers and feedback in the Supplementary Material.

[3]https://github.com/Schascha1/iSHAP