GX-HUI: Global Explanations of AI Models based on High-Utility Itemsets

Davide Napolitano
Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Turin, Italy
0000-0001-9077-4103

Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Turin, Italy
0000-0002-7185-5247

***Abstract*—Shapley Values are established concepts used to explain local and global contribution of individual features to the prediction of AI models. Currently, global Shapley-based explainers do not consider the co-occurrences of feature-value pairs in the analyzed data. This paper proposes a novel approach to leverage the High-Utility Itemset Mining framework to jointly consider Shapley-based feature-level contributions and feature-value pair co-occurrences. The results achieved on benchmark datasets show that the extracted patterns provide actionable knowledge, complementary to those of global Shapley Values.**

***Index Terms*—Explainable AI, Global Explainer, High-Utility Itemset Mining, Model-based Explainability**

I. Introduction

The diffusion of AI models has recently fostered the demand of opening the AI black boxes [1]. Shapley Values [2] (SVs) are concepts from cooperative game theory that are established in explainable AI. They can be used to quantify the contribution of a given feature to both *local* and *global* class label predictions. However, global Shapley-based explainers [3] do not consider the conjunction of feature-value pairs, providing feature-level descriptions rather than pattern-level ones.

High-Utility Itemsets [4] (HUIs) are descriptive patterns representing recurrent conjunctions of highly valued items in a transactional dataset. In this paper we propose to leverage HUIs to provide end-users with global model explanations incorporating instance-level item co-occurrences. To this end, we define the concepts of *item* as a $feature{:}value$ pair and *utility* of a feature within a given instance as the corresponding SV. In such a way, a HUI represents a conjunction of feature-value pairs where the contribution of the involved features is averagely high. Thus, the utility of the HUI incorporates both feature- and pattern-level relevance providing information complementary to global SVs [3].

We propose *GX-HUI*, a *G*lobal AI model e*X*plainer leveraging the *H*igh-*U*tility *I*temset Mining framework [4]. It relies on a model-based SV approximation [5] producing real-time SV estimates based on Neural Network model. The per-class descriptors consist of a shortlist of HUIs capturing both feature- and pattern-level contributions.

Motivating example. Figure 1 shows the top-20 HUIs extracted from the Monks dataset [6] separately for each class. For example, the HUI $\{a5{:}1\}$ indicates that feature $a5$ takes value 1 with a relevantly high contribution to class *positive*. Similarly, HUI $\{a5{:}1,a3{:}2\}$ describes a conjunction of items associated with highly valued features. Notice that, unlike in traditional Shapley-based models, a highly influential feature could be under-represented or even neglected by the HUI model as none of its feature-value combinations is deemed as sufficiently influential. For example, according to the plots shown in Figure 2 feature $a4$ is missing in the top HUIs for class *negative* whereas its global SVs are above zero for both classes and comparable in magnitude to those of feature $a6$. Conversely, for the class *positive* feature $a4$ appears to be as much relevant as all the other features because most of the top-k itemsets include it. Therefore, the pattern-level analysis provides information complementary to the established global Shapley-based models as could reveal interesting occurrence-level correlations neglected by global explainers. The main paper contributions are summarized below.

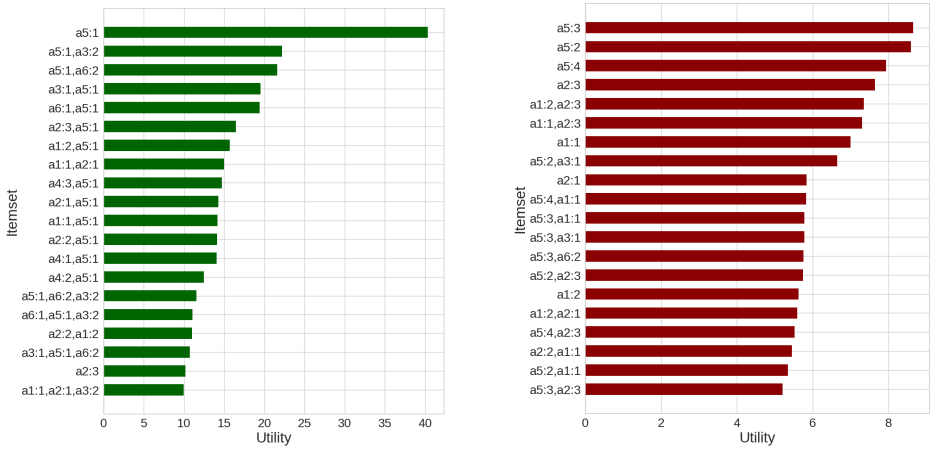
- We leverage the HUI Mining framework to generate global Shapley-based AI model explanations.
- We present GX-HUI, a global explainer integrating both efficient SV approximation and HUI mining.
- We evaluate the proposed approach on 4 UCI [6] benchmark datasets, highlighting patterns that cannot be easily identified by relying on Global SVs [3] solely.

II. Estimating the Shapley Values

The Shapley value [2] is a solution concept in cooperative game theory that assigns a value to each player in a cooperative game based on the contribution to the total payoff of the group. Formally speaking, the SV for a player i in a cooperative game with a set N of players and a characteristic function $v : 2^N \rightarrow \mathbb{R}$ is defined as follows:

$$\phi(i) = \sum_{S \subseteq N \setminus i} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}[v(S \cup i) - v(S)]$$

where the sum is over all possible subsets S of players that do not contain player i. The term $v(S \cup i) - v(S)$, hereafter denoted by $\Delta(S, i)$, is the marginal contribution of player i to the coalition S.



(a) Class *Positive* (b) Class *Negative*

Fig. 1: Top-20 HUIs extracted from the Monks dataset

The SV satisfies the following axioms, ensuring fairness and a reasonable division of the total payoff among the players:

- *Efficiency*: $\sum_{i \in N} \phi_i = v(N)$.
- *Symmetry*: $\forall S \subseteq N, \Delta(S,i) = \Delta(S,j)$, then $\phi_i = \phi_j$.
- *Linearity*: $\phi_{v_1+v_2}(i) = \phi_{v_1}(i) + \phi_{v_2}(i)$.
- *Dummy player*: $\forall S \subseteq N, v(S \cup i) = v(S)$, then $\phi_i = 0$.

Our goal is to exploit the SVs associated with local predictions to generate global explanations of the predictor. Hence, similar to [3], we leverage the linearity axiom to produce a linear combination of the local per-instance SV estimates.

A. Computational approaches

Computing the SVs in a real-world scenario is challenging. The main approaches to SV computation can be classified as stochastic estimators or model-specific approximations.

a) Stochastic estimators: These models (e.g., [3], [7]) rely either on sampling or generating permutations of the input features. As a drawback, to achieve good accuracy performance they require high computational times.

b) Model-specific approximations: these explainers are mainly based on decision trees (e.g., [8]) or Neural Networks (e.g., [9]–[11]). They are commonly more efficient than stochastic approaches but require ad hoc model evaluations to configure the model parameters. In the present work we rely on a state-of-the-art model-based approach, namely FastSHAP [5].

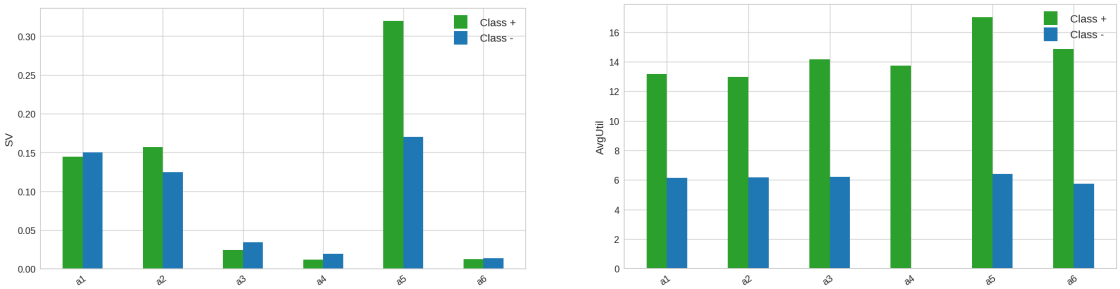
c) FastSHAP: [5] performs real-time estimations of the SVs. It exploits a *surrogate model*, that simulates the original model to be explained by considering different subsets of features, Based on the outputs of the surrogate model, FastSHAP returns the SV approximation in a single-forward pass by minimizing the difference between the surrogate model output and the local normalized output.

B. Evaluation of the Shapley Value estimates

The main strategies to assess the quality of SV estimates are application-grounded, human-grounded, and functional-grounded evaluation [12], our focus as it does not require a human judgment. Explanation models can be further split in:

- *Model-based explanations*: the model itself is used as an explanation (e.g. a decision tree) or a more interpretable model is generated.
- *Attribution-based explanations*: it relies on a measure (e.g. feature importance) of the quality of the explanation.
- *Example-based explanations*: it explains a model by selecting specific data instances.

In this work we define specific model-based explainers using the HUI mining framework (see Section IV for further details).



(a) Estimated Shapley Values (b) Average utility

Fig. 2: Comparison between per Average utility and Shapley Value estimates for each feature and class of the Monks dataset

We also report a comparison with the *Mutual Information*, which is commonly used to explain model fidelity [13].

III. High-Utility Itemset Mining

High-utility Itemset miners are unsupervised algorithms aimed at extracting highly valued patterns, called High-Utility Itemsets, from a transactional database [14].

a) Data model: Let $\mathcal{D} = \{T_1, T_2, \ldots, T_n\}$ be a transactional database, where each transaction T_i is a set of distinct items $T_i = \{x_1, x_2, \ldots, x_k, \ldots, x_m\} \in \mathcal{X}$. From now on, we will focus on structured data where each item x_k consists of a feature-value pair $feature{:}value$. Given a transaction T_i, each item $x_k \in T_i$ is characterized a *internal utility* $Q(x_k, T_i)$ quantifying its local value. Each item can be also associated with an *external utility* $P(x_k, \mathcal{D})$, indicating a global influence value. The *absolute utility* of an item x_k in a transaction T_i is denoted as $u(x_k, T_i) = P(x_k, \mathcal{D}) \cdot Q(x_k, T_i)$.

b) HUI mining: An itemset I is a set of arbitrary items in $\mathcal{X}$ such that each item in I corresponds to a different feature. The utility u_I of itemset I in a given transaction $T_i \in \mathcal{D}$ is defined as the sum of the absolute utilities of its items $x_k \in I$:

$$u(I, T_i) = \sum_{x_k \in I} u(x_k, T_i)$$

Analogously, the total utility of I in the entire dataset $\mathcal{D}$, $u(I, \mathcal{D}) = \sum_{T_i \in \mathcal{D}} u(I, T_i)$, is the sum of the corresponding per-transaction utilities. If the total utility of I is above an user-specified treshold thr (i.e., $u(I, \mathcal{D}) > thr$) then I is a HUI. Given a transactional database $\mathcal{D}$ and an absolute threshold thr, HUI mining entails extracting *all* the HUIs in $\mathcal{D}$.

c) Implementation: To efficiently extract HUIs we apply the Enhanced Frequent Itemset Mining algorithm (EFIM) [15].

IV. GX-HUI

The presented method, namely *Global eXplainer based on High-Utility Itemsets* (GX-HUI) produces global explanations consisting of HUIs. Algorithm 1 enumerates the main steps.

Let $\mathcal{D}$ be the dataset under consideration and M be the AI model to be explained. Our goal is to explain the predictions made by M on transactions $T_i \in \mathcal{D}$. Rather than explaining each local prediction, we aim at collecting a global descriptions of the AI model consisting of the HUIs extracted separately from each class. For the sake of simplicity, we will focus on explaining a binary classification model M predicting either the *positive* $(+)$ or the *negative* class $(-)$. The multiclass case can be straightforwardly modeled by picking each class.

The GX-HUI method extracts for each class the top-k HUI in order of decreasing total utility (where k is a user-specified parameter). To leverage the SV approximations, we compare the class labels assigned by M to each transaction $T_i \in \mathcal{D}$ with those available in the training set (i.e., the ground truth). Then, we split the transactions in $\mathcal{D}$ into the *positive* and *negative* subsets $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively, and define the utilities of an item x_k as the corresponding SV estimates:

$$u(x_k, \mathcal{D}^+) = \phi_k^+; \quad u(x_k, \mathcal{D}^-) = \phi_k^- \tag{1}$$

Algorithm 1 GX-HUI pseudo-code

Require: Dataset $\mathcal{D}$, Model M, predictions Pr, Mininum utility threshold thr

```
Φ = Shapley-Value-Estimate(D, M, Pr)
S = ComputeMutualInformation(Φ, Pr)
for class in D do
    D^class, Φ^class = Data-Model(D, Φ)
    HUI^class = HUI-Mining(D^class, Φ^class, thr)
    S = S ∪ ComputeHUIMetrics(HUI^class, Φ^class)
end for
return Statistics S
```

a) HUI model descriptors: We describe the GX-HUI explanations according to the following statistics derived from the HUI model.

- *Feature Coverage* (FC_i): percentage of top-k HUIs including feature i, i.e. $FC_i = \frac{frequency(i)}{k}$. It quantifies the *pattern-level* relevance of feature i.
- *Top-k Average utility* ($AvgUtil_i^k$): average of the absolute of the estimated SVs[1] over the top-k HUIs. Let $\text{HUI}_1, \ldots, \text{HUI}_k$ be the the top-k HUIs and let $\mathcal{H}^i$ be the set of top-k HUIs containing feature i, the top-k Average utility is defined by $AvgUtil_i^k = \frac{\sum_{HUI_j \in \mathcal{H}^i} u(HUI_j, \mathcal{D})}{|\mathcal{H}^i|}$. It indicates the pattern-based importance of feature i in the top-k HUIs involving both single items and item conjunctions.
- *Item Average utility* ($AvgUtil_i^{item}$): average of the absolute of the estimated SVs[1] over the high-utility single items. It indicates the pattern-based importance of feature i in the top-k HUIs involving only single items (thus disregarding any item conjunction). Let $\text{HUI}_1, \ldots, \text{HUI}_n$ be the all HUIs and let $\mathcal{H}^i$ be the set of HUIs containing feature i and with size one, the item average utility is defined by $AvgUtil_i^{item} = \frac{\sum_{HUI_j \in \mathcal{H}^i} u(HUI_j, \mathcal{D})}{|\mathcal{H}^i|}$.
- *Domain Coverage* (DC_i): given a feature i, it indicates the percentage of items x_i that occur in the top-k HUIs.

Hereafter, we will also consider the *Mutual Information* (*MI*) between SVs and the class [13] as a baseline indicator.

V. Experiments

In this section we analyze the outcomes of GX-HUI on 4 UCI [6] tabular datasets suitable for HUI applications: Monks, WBC, Hearth and Census. We compare the derived HUI statistics with the established Global SVs [3].

A. Experimental design

The experiments were run on machines equipped with an Intel Xeon Gold 6140 and Nvidia Tesla T4.

To estimate the SVs we used the implementation of FastSHAP provided by [5]. To extract HUIs we used the implementation of the EFIM algorithm [15] available in the SPMF library [16]. To make integer utility values and real SVs compatible, we rounded the latter up to the 5^{th} decimal digit

[1] We consider the magnitude of the SV and neglect the sign.

and then multiplied by $\alpha = 10^5$. All the computed itemset utilities were normalized accordingly.

We tested various configurations for the k parameter indicating the number of top-k HUIs included in the global explanations. As representative k values, we tested 20, 50, 100 and 1000. Considering too low or too large numbers of HUIs is deemed as weakly informative and hard to manage by human experts, respectively. To extract a comparable number of itemsets across different datasets and classes and to limit computational time we set the utility threshold to 1 for Monks and WBC, $5 \cdot 10^4$ for Hearth and $4 \cdot 10^5$ for Census.

Feature	MI	AU$^+$	Pv$^+$	FC$^+$	AU$^-$	Pv$^-$	FC$^-$
MONKS							
a1	↑	↑	×	0.25±0.0	↑	✓	0.45±0.0
a2	↑	↑	×	0.35±0.0	↑	✓	0.45±0.0
a3	∼	∼	×	0.3±0.0	∼	×	0.1±0.0
a4	∼	∼	×	0.15±0.0	∼	-	0±0
a5	↑	↑	✓	0.8±0.0	↑	✓	0.59±0.03
a6	∼	∼	×	0.25±0.0	∼	-	0.05±0.0
WBC							
ClThick	∼	∼	-	0±0	↑	×	0.21±0.02
UCSize	↑	∼	×	1.0±0.0	∼	×	0.19±0.02
UCShape	↑	↑	×	0.79±0.07	↑	×	0.2±0.06
MargAdh	↑	∼	×	0.52±0.11	∼	-	0.08±0.03
EpCSize	∼	∼	×	0.57±0.12	∼	-	0±0
BareNucl	↑	↑	×	0.89±0.05	↑	×	0.61±0.05
BlChrom	∼	∼	-	0±0	↑	×	0.18±0.05
NormNucl	↑	∼	×	0.7±0.06	↑	×	0.14±0.02
Mitoses	↑	∼	×	0.65±0.09	∼	×	0.26±0.02
HEART							
age	∼	∼	-	0±0	∼	-	0±0
sex	↑	∼	×	0.47±0.03	∼	×	0.39±0.02
cstPain	∼	↑	×	1.0±0.0	↑	-	0±0
bldPress	↑	∼	-	0±0	∼	-	0±0
serChol	∼	∼	-	0±0	∼	-	0±0
fBldSug	∼	∼	-	0.39±0.02	∼	-	0.45±0.04
rstEleRes	∼	∼	×	0.16±0.02	∼	-	0±0
maxHRate	∼	↑	-	0±0	∼	-	0±0
exIndAng	↑	↑	×	0.64±0.02	∼	×	0.45±0.04
oldpeak	∼	∼	-	0±0	∼	-	0±0
slope	↑	∼	-	0.1±0.04	∼	×	0.21±0.02
majVess	∼	↑	-	0±0	↑	×	0.91±0.07
thal	↑	↑	×	0.44±0.02	↑	×	0.69±0.06
CENSUS							
Age	∼	∼	-	0±0	∼	-	0±0
Wclass	∼	∼	-	0±0	∼	-	0±0
Ed-Num	∼	↑	-	0±0	↑	-	0±0
MarStat	∼	∼	×	1.0±0.0	↑	×	1.0±0.0
Occup	∼	∼	-	0±0	∼	-	0±0
Rel	∼	↑	×	1.0±0.0	↑	×	1.0±0.0
Race	∼	∼	×	0.45±0.0	∼	×	0.5±0.0
Sex	↑	∼	×	0.8±0.0	∼	×	0.7±0.0
CapGain	↑	↑	×	0.6±0.0	∼	×	0.7±0.0
CapLoss	↑	∼	×	0.45±0.0	∼	×	0.6±0.0
Hweek	∼	∼	-	0±0	∼	-	0±0
Country	↑	∼	×	0.45±0.0	∼	×	0.4±0.0

TABLE I: Feature- and pattern-level statistics. Significance t-test level: above 95%. − means not applicable/not succeeded ↑ means high value, ∼ indicates other values.

B. Comparison between Mutual Information and Average Utility statistics

Table I reports the main statistics collected on the datasets. The MI is one the indicators that are most established for evaluating the consistency of the SV estimates with the class labels [13]. For example, for Monks features $a1$, $a2$, and $a5$ have maximal MI as they are the most discriminating ones to assign the class label. The per-class average utilities computed on top of the HUIs confirm the prior knowledge.

Columns *Pv*$^-$ and *Pv*$^+$ respectively report the average p-values[2] of the two-tailed t-test of independence between the top-20 Average utility and Item Average utility. Overall, all tests are executed on $AvgUtil_i^k$ and $AvgUtil_i^{item}$. Their correlation shows how highly contributing features are indirectly influenced by the presence of other features in the top-k HUIs. Such an insight is not directly derivable from classical Shapley models as they neglect item co-occurrences. The outcomes of t-test achieved on Monks indicate the presence of a correlation for the feature $a5$ and the *positive* class. This is confirmed by a deeper exploration of the Global SVs (see Figure 2), where the predominance of $a5$ compared to $a1$ and $a2$ is evident. By construction, in the Monks dataset when feature $a5$ takes value one then the class is always *positive*. Feature $a5$ is the only feature directly influencing the class *positive* regardless of the presence of other associated items. Conversely, when features $a1$ and $a2$ take the same value they jointly determine the class label *positive* but they are discriminating while considered one at a time. Thus, the latter can be deemed as a weaker relationship between a specific feature and a class label. While setting k to 20, the t-tests succeed only on Monks, likely because the dataset contains relatively few rows and features and considering just a small value of HUIs is adequate. Conversely, as discussed in Section V-C, on larger datasets a larger number of HUIs is required.

C. Effect of K

The number k of selected HUIs has a relevant impact on the selectivity of the global explainer. Specifically, Table II reports the feature coverage and p-values separately for each dataset and for various values of k. While considering relatively small itemset-based models (e.g., 20) the selectivity of the model is quite high thus identifying only a limited subset of high-valued features. Conversely, by increasing the value of k the statistics also include less valued patterns thus the outcomes are much similar to those of existing global explainers (e.g., [3]). The correlation between Item Average utility and Average utility is also influenced by the number of considered HUIs. The larger k the higher the influence of relatively low-utility items contributing to low-ranked HUIs. For example, by setting k to 50 on the Monks dataset the trends of the considered statistics appear to be divergent, indicating an anomalous trend likely due to an improper setting of parameter k. We recommend end-users to set the value of k to 20 on relatively small datasets (i.e., few hundreds of rows), whereas setting k to

[2]computed over several repeated runs to get reliable estimates.

Feature	Pv^+	FC^+	Pv^-	FC^-	Pv^+	FC^+	Pv^-	FC^-	Pv^+	FC^+	Pv^-	FC^-	Pv^+	FC^+	Pv^-	FC^-
Monks																
	TopK=20				TopK=50				TopK=100				TopK=1000			
a1	×	0.25±0.0	✓	0.45±0.0	✓	0.35±0.01	✓	0.46±0.01	✓	0.42±0.01	×	0.47±0.01	×	0.57±0.006	×	0.54±0.002
a2	×	0.35±0.0	✓	0.45±0.0	✓	0.42±0.0	✓	0.52±0.01	✓	0.55±0.01	×	0.56±0.02	×	0.67±0.004	×	0.65±0.002
a3	×	0.3±0.0	×	0.10±0.0	×	0.34±0.0	×	0.26±0.02	×	0.37±0.01	×	0.33±0.01	✓	0.59±0.002	✓	0.60±0.002
a4	×	0.15±0.0	-	0±0	×	0.18±0.01	-	0±0	×	0.31±0.01	×	0.07±0.01	✓	0.63±0.001	✓	0.65±0.000
a5	✓	0.8±0.0	✓	0.59±0.03	✓	0.82±0.01	✓	0.58±0.01	✓	0.73±0.0	×	0.58±0.02	×	0.75±0.004	×	0.70±0.001
a6	×	0.25±0.0	-	0.05±0.0	×	0.40±0.01	×	0.18±0.0	×	0.36±0.01	×	0.28±0.01	✓	0.57±0.002	✓	0.58±0.001
WBC																
	TopK=20				TopK=50				TopK=100				TopK=1000			
ClumpThick	-	0±0	×	0.22±0.03	-	0±0	×	0.31±0.01	-	0±0	×	0.29±0.01	×	0.48±0.01	✓	0.41±0.03
UCellSize	×	1.0±0.0	×	0.19±0.02	×	0.88±0.05	×	0.30±0.02	×	0.59±0.02	×	0.32±0.01	×	0.56±0.01	✓	0.46±0.01
UCellShape	×	0.80±0.08	×	0.2±0.06	×	0.69±0.04	×	0.32±0.02	×	0.58±0.01	×	0.33±0.01	×	0.54±0.01	✓	0.45±0.01
MargAdhes	×	0.52±0.11	-	0.09±0.02	×	0.53±0.05	×	0.20±0.02	×	0.53±0.02	×	0.22±0.01	×	0.50±0.01	✓	0.37±0.01
EpCellSize	×	0.57±0.12	-	0±0	×	0.56±0.07	-	0.02±0.03	×	0.54±0.02	×	0.08±0.01	×	0.51±0.01	×	0.31±0.01
BareNuclei	×	0.89±0.05	×	0.63±0.02	×	0.71±0.02	×	0.58±0.02	×	0.60±0.01	✓	0.69±0.01	×	0.55±0.01	✓	0.69±0.03
BlandChrom	-	0±0	×	0.17±0.03	-	0±0	×	0.20±0.01	-	0±0	×	0.22±0.01	×	0.58±0.01	✓	0.45±0.03
NormalNucl	×	0.7±0.06	×	0.13±0.03	×	0.61±0.07	×	0.17±0.03	×	0.55±0.01	×	0.17±0.01	×	0.52±0.01	✓	0.37±0.01
Mitoses	×	0.66±0.10	×	0.22±0.03	×	0.56±0.06	×	0.22±0.0	×	0.54±0.02	×	0.26±0.01	×	0.52±0.01	×	0.41±0.01
Heart																
	TopK=20				TopK=50				TopK=100				TopK=1000			
age	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0.008±0.003	-	0±0
sex	×	0.47±0.03	×	0.39±0.03	×	0.49±0.01	×	0.34±0.01	×	0.53±0.02	×	0.36±0.02	✓	0.51±0.007	✓	0.51±0.005
chestPain	×	1.0±0.0	-	0±0	×	0.88±0.02	×	0.12±0.02	×	0.72±0.02	×	0.14±0.01	✓	0.62±0.01	✓	0.47±0.01
bloodPress	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0.04±0.002	-	0.001±0.001
serumCholl	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0
fBloodSug	-	0.39±0.03	-	0.45±0.04	-	0.40±0.02	-	0.47±0.04	-	0.44±0.01	-	0.47±0.01	-	0.48±0.01	-	0.46±0.004
ElectrRe	×	0.16±0.03	-	0±0	×	0.37±0.01	×	0.09±0.03	×	0.40±0.01	×	0.14±0.01	✓	0.43±0.01	×	0.42±0.005
MHeartRate	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0	-	0±0
exerIndAng	×	0.64±0.03	×	0.45±0.04	-	0.54±0.02	×	0.38±0.02	×	0.55±0.02	×	0.43±0.02	✓	0.49±0.005	✓	0.49±0.005
oldpeak	-	0±0	-	0±0	-	0±0	-	0.03±0.01	-	0±0	-	0.10±0.01	-	0.10±0.008	-	0.28±0.001
slope	×	0.13±0.04	×	0.21±0.03	×	0.30±0.02	×	0.24±0.07	×	0.44±0.02	×	0.36±0.01	✓	0.47±0.005	✓	0.48±0.01
majorVess	-	0±0	×	0.92±0.07	-	0±0	×	0.81±0.08	×	0.09±0.02	×	0.80±0.03	✓	0.55±0.002	✓	0.61±0.01
thal	×	0.44±0.03	×	0.69±0.07	×	0.42±0.02	×	0.59±0.07	×	0.51±0.02	✓	0.67±0.04	✓	0.58±0.004	✓	0.58±0.01

TABLE II: P-values varying k. Monks, WBC, and Hearth datasets. − means not applicable/ succeeded.

1000 on datasets consisting of thousands of rows. Due to space limitations and similar results w.r.t. Heart, results on Census are neglected in Table II and Figure 3.

D. Comparison between feature and domain coverage

Figure 3 shows how the domain coverage values vary across datasets, features, and number k of shortlisted HUIs. On most of the analyzed datasets few items are predominant and tend to over-influence the pattern utility. For example, on most of the features of the Heart dataset the feature coverage is quite high whereas the domain coverage quite low (see Figure 3). This is due to the fact that some item combinations are biased by the presence of (few) highly valued items. More specifically, when the feature coverage is high, that specific feature likely appears in most of the HUIs. Whether the domain coverage of the same feature is low it is a strong clue of the presence of highly valued items in the HUI shortlist.

VI. Conclusions and Future Work

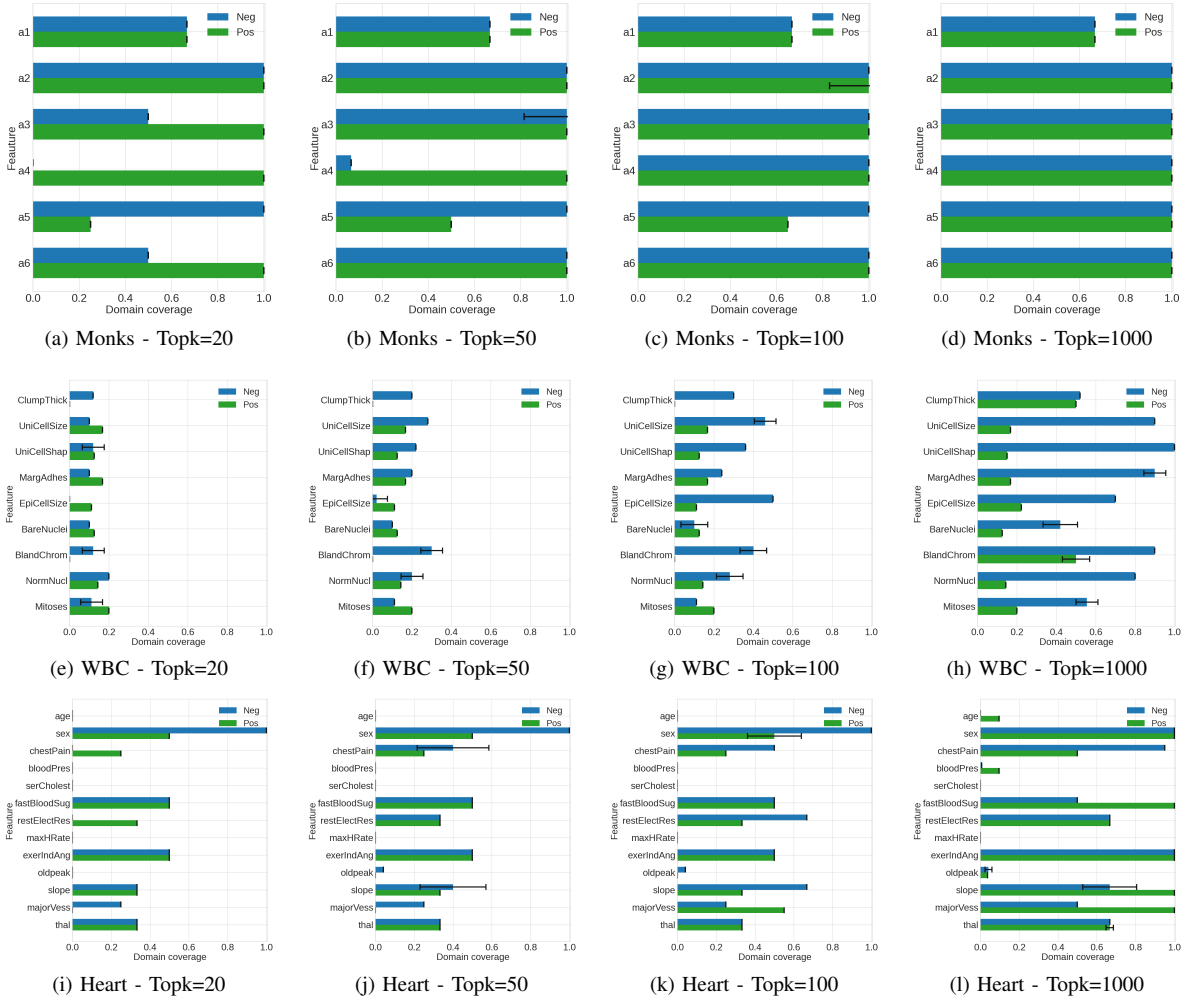
The paper presented a global AI model explainer that combines the SVs and the HUI Mining framework. The global model consists of HUIs providing pattern-level view of the most contributing features and items. The experiments show that:

- The feature coverage values are in line with the expectation (as long as proper values of k and thr are used).
- The average item and feature utilties can be divergent when either the contributions of other high-valued items are significant or a particular item is linked with many low-valued items.
- The comparison between pattern-based models allows us to identify peculiar cases in which the contributions of various feature combinations are diversified.
- The selection of an excessive number of top-k HUIs can be prevented via statistical tests (t-tests).

As future work, we plan to explore to use HUIs for local explainability, a quantitative metric to assess the performance, the integration of the concepts of Coalition Interval and the development of more efficient SV estimators.

Acknowledgments

This study was partially carried out within the FAIR - Future Artificial Intelligence Research - and received funding from the European Union Next-GenerationEU (PNRR



(a) Monks - Topk=20 (b) Monks - Topk=50 (c) Monks - Topk=100 (d) Monks - Topk=1000

(e) WBC - Topk=20 (f) WBC - Topk=50 (g) WBC - Topk=100 (h) WBC - Topk=1000

(i) Heart - Topk=20 (j) Heart - Topk=50 (k) Heart - Topk=100 (l) Heart - Topk=1000

Fig. 3: Domain Coverage Analysis

MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 D.D. 1555 11/10/2022, PE00000013). This paper reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, *Explainable AI Methods - A Brief Overview*, 2022, pp. 13–38.

[2] L. S. Shapley, *Notes on the N-Person Game II: The Value of an N-Person Game*. Santa Monica, CA: RAND Corporation, 1951.

[3] I. Covert, S. M. Lundberg, and S.-I. Lee, "Understanding global feature contributions with additive importance measures," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 17 212–17 223.

[4] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2012, p. 55–64.

[5] N. Jethani, M. Sudarshan, I. C. Covert, S. Lee, and R. Ranganath, "Fastshap: Real-time shapley value estimation," in *ICLR 2022*.

[6] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4765–4774.

[8] S. M. Lundberg, G. G. Erion, H. Chen, A. J. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S. Lee, "Explainable AI for trees: From local explanations to global understanding," *CoRR*, vol. abs/1905.04610, 2019.

[9] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "L-shapley and c-shapley: Efficient model interpretation for structured data," *CoRR*, 2018.

[10] M. Ancona, C. Oztireli, and M. Gross, "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 09–15 Jun 2019, pp. 272–281.

[11] R. Wang, X. Wang, and D. I. Inouye, "Shapley explanation networks," *CoRR*, vol. abs/2104.02297, 2021.

[12] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, 2021.

[13] A. Nguyen and M. R. Martínez, "On quantitative aspects of model interpretability," *CoRR*, vol. abs/2007.07584, 2020.

[14] V. S. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient algorithms for mining top-k high utility itemsets," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 54–67, 2016.

[15] S. Zida, P. Fournier-Viger, C.-W. Lin, C.-W. Wu, and V. S. Tseng, "Efim: A highly efficient algorithm for high-utility itemset mining," in *Mexican International Conference on Artificial Intelligence*, 2015.

[16] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C. Wu., and V. S. Tseng, "SPMF: a Java Open-Source Pattern Mining Library," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 3389–3393.