IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2024, accepted 4 September 2024, date of publication 6 September 2024, date of current version 17 September 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3455563*

RESEARCH ARTICLE

# Interpretable Classifier Models for Decision Support Using High Utility Gain Patterns

**SRIKUMAR KRISHNAMOORTHY, (Member, IEEE)**
Information Systems Area, Indian Institute of Management Ahmedabad, Ahmedabad, Gujarat 380015, India
e-mail: srikumark@iima.ac.in

**ABSTRACT** Ensemble models such as gradient boosting and random forests are proven to offer the best predictive performance on a wide variety of supervised learning problems. The high performance of these black box models, however, comes at a cost of model interpretability. They are also inadequate to meet regulatory demands and explainability needs of organizations. The model interpretability in high performance black-box models is achieved with the help of post-hoc explainable models such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). This paper presents an alternate intrinsic classifier model that extracts a class of higher order patterns and embeds them into an interpretable learning model. More specifically, the proposed model extracts novel High Utility Gain (HUG) patterns that capture higher order interactions, transforms the model input data into a new space, and applies interpretable classifier methods on the transformed space. We conduct rigorous experiments on forty benchmark binary and multi-class classification datasets to evaluate the proposed model against the state-of-the-art ensemble and interpretable classifier models. The proposed model was comprehensively assessed on three key dimensions: 1) quality of predictions using classifier measures such as accuracy, $F_1$, AUC, H-measure, and logistic loss, 2) computational performance on large and high-dimensional data, and 3) interpretability aspects. The HUG-based learning model was found to deliver performance comparable to that of the state-of-the-art ensemble models. Our model was also found to achieve 2-40% (45%) prediction quality (interpretability) improvements with significantly lower computational requirements over other interpretable classifier models. Furthermore, we present case studies in finance and healthcare domains and generate one- and two-dimensional HUG profiles to illustrate the interpretability aspects of our HUG models. The proposed solution offers an alternate approach to build high performance and transparent machine learning classifier models. We hope that our ML solution help organizations meet their growing regulatory and explainability needs.

**INDEX TERMS** Analytics, interpretable machine learning, explainable artificial intelligence, classification, high utility patterns.

## I. INTRODUCTION

Machine learning and advanced analytical models have become an integral part of data-driven decision making in numerous organizations. The decision support capabilities of these models are quite diverse in nature and include delivering personalization services at Netflix, recidivism prediction for criminal justice, credit risk assessment, patient health risk management, delivery route optimization, and autonomous self-driving cars. The decision support capabilites of advanced analytical models have often surpassed humans in numerous decision environments. However, an uncontrolled use of such models have also created problems in several domains. For instance, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) [1] is an automated criminal justice system that is widely used in U.S. for making parole and bail related decisions. The proprietary and black-box nature of the COMPAS system poses the following issues: First, the system takes automated decisions without providing explanations. This makes it hard for

The associate editor coordinating the review of this manuscript and approving it for publication was Loris Belcastro.

defendants and their attorneys to understand the reasons for denial of bail [2]. Second, a high stake decision environment like criminal justice, require effective use of new information, that was unavailable at the time of model building, and adjust the model scores (or weights). A black-box models like COMPAS would make it nearly impossible for a judge to combine the new information to adjust the defendant's risk score and make a fair decision [3].

In the healthcare domain, Zech et al [4], analyze the performance of a black-box model for predicting pneumonia in chest radiographs. Their study observed that the word ''portable'', an equipment type, mentioned in the x-rays was given higher weights by the model than the actual content of the image for prediction. These model level insights are crucial to design robust machine learning systems in high stakes decision making environments.

In the financial services domain, Equal Credit Opportunity Act (ECOA) and the Fair Credit Reporting Act (FCRA) require financial institutions to offer explanations to their customers for denial of credit or other adverse decisions. These regulatory demands often constrains the financial institutions from using advanced machine learning models.

The foregoing discussions reveal that there is a strong need for deciphering how the decisions are being made by advanced machine learning models. The scientific advancements in interpretable and explainable models primarily aim to address this problem, and help meet the diverse needs of end-users, data scientists, developers, researchers, and decision-makers. Du et al [5] present two broad categories of interpretable models: intrinsic and post-hoc. The intrinsic models are self-explanatory models such as decision trees, rule based models, and linear models. On the other hand, the post-hoc models are surrogate models that use local approximations to generate explanations from blackbox models. While post-hoc models such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) have become popular, Rudin et al [6] argue that post-hoc explanations are often incorrect or incomplete. Several prior studies [3], [7], [8], [9] have also reported concerns related to fidelity and false charaterizations of post-hoc models that might create potential harm to organizations and society. Chen et al [10] investigate the use of post-hoc explainable models in the credit lending domain and offer rich illustrations of contradictory explanations generated by these models.

The past research studies in interpretable machine learning show experimental evidence that the accuracy of classifier models need not come at the cost of model interpretability [6]. Many of the real-world datasets are found to exhibit Rashomon effect [6], [11], [12] and have large Rashomon sets i.e. multiple descriptions or models of the same dataset with similar performance results. Although the characteristics of the Rashomon set are unknown [6], an exploration of innovative and inherently interpretable models is needed to identify simpler models within this set.

Recent research attempts on interpretable ML models include: automated feature engineering [13], [14], Supervised Assisted Feature Extraction (SAFE) [9], Penalized Logistic Tree Regression (PLTR) [15], and INterpretable Automated Feature ENgineering (INAFEN) [16]. These approaches either use greedy heuristics to determine the best feature transformations for automated feature engineering or rely on other machine learning models (e.g. decision trees, rules, random forest, boosting) for feature extraction. The identified features are then used as part of another interpretable machine learning model to achieve better interpretability. Our study is distinct from these efforts and aim to build an inherently interpretable model using a new class of features and higher-order interaction patterns. More specifically, our primary motivation is to investigate and answer the following research questions:

1) Are there specific class of higher-order interaction patterns that can be used to model supervised learning problems and achieve better interpretability?
2) Can high-utility itemset mining or its variants identify interesting patterns and support interpretable machine learning and avoid costly exhaustive enumeration that lacks scalability and suffer from overfitting issues?
3) How does the performance, scalability, and interpretability of the models based on a specific class of higher order patterns compare with that of other state-of-the-art ensemble and interpretable classifier models?

Frequent patterns and association rules [17], [18] have been explored in the literature for associative classification tasks. Some of the prominent methods of associative classification include: Classification Based on Associations (CBA) [19], Classification based on Multiple Association Rules (CMAR) [20], Classification based on Predictive Association Rules (CPAR) [21], Association Rule Tree (ART) [22], and SigDirect [23]. These associative classification methods use the occurrence frequency information of patterns and are known to offer high interpretability. But, the predictive performance of these methods is poor compared to advanced state-of-the-art classifier ensembles such as random forest and boosting. The use of frequency as the primary criterion in these methods fail to account for complex feature interactions and imbalanced nature of real-world data. In the recent years, High Utility Itemset (HUI) mining [24], [25], [26] has emerged as an alternative that addresses the common limitations of methods that purely focus on occurrence frequencies.

### A. CONTRIBUTION AND ORGANIZATION

To the best of our knowledge, the use of high utility itemsets or patterns for supervised learning problem has not been studied in the literature. Our objective in this paper is to design a new class of utility patterns and explore its application in the domain of interpretable machine learning. We introduce new data transformation methods, define new approaches for measuring utilities in the context of supervised

learning problems and present a variant of high utility patterns, named, High Utility Gain (HUG) patterns. We then introduce intrinsic interpretable classifier models that utilize the proposed High Utility Gain (HUG) patterns as higher-order features.

The key contributions of the paper are as follows:

- Introduce a new class of utility patterns, named, High Utility Gain (HUG) patterns that exploit the outcome (or label) information available in supervised learning problems.
- Propose intrinsic interpretable classifier models for tabular data that leverage the proposed High Utility Gain (HUG) patterns.
- Present a new algorithm to perform scalable extraction of HUG patterns. The proposed algorithm adapts a top-k HUI mining algorithm for the new pattern mining problem.
- Propose three interpretable classifier models that use the mined HUG patterns as features for supervised learning. A rigorous comparative evaluation of the proposed models against the state-of-the-art ensemble and interpretable classifier models on 40 benchmark datasets is also made.
- Assess the proposed models on three dimensions: (1) prediction quality performance on diverse classifier measures such as accuracy, $F_1$, AUC, H-measure, and logistic loss, (2) computational performance, and (3) interpretability measures.
- Present case studies in finance and healthcare domains to demonstrate the interpretability aspects of the proposed interpretable classifier models.

The rest of this paper is organized as follows. We review the literature in section II and present the key notations and definitions in section III. The proposed high utility gain pattern mining methods and interpretable learning models are presented in section IV. A rigorous experimental evaluation of the proposed ideas on wide variety of supervised learning problem are made in section V. Illustrations of the interpretability aspects of the model are given in section VI. Discussions on theoretical and practical implications are then made in section VI. Finally, the paper concludes with a discussion of the limitations and future research directions.

## II. RELATED WORK

Our work lies at the intersection of two distinct streams of research literature, namely, high utility itemset mining, and interpretable machine learning. We, therefore, present a review of key research works that are related to the proposed study.

### A. HIGH UTILITY ITEMSET (HUI) MINING

High Utility Itemsets (HUI) [24] are itemsets that use a notion of decision makers' utility in place of occurrence frequencies to mine interesting patterns. The decision makers' utility is defined as a function of the quantity of items purchased and the profitability of items. Unlike frequent itemsets, a high utility itemset does not satisfy downward closure property. Hence, they are computationally harder than frequent itemsets. Numerous algorithms have been proposed in the literature to efficiently mine HUIs: HUI-Miner [24], FHM [29], EFIM [30] HMiner [31]. Several variants of the basic HUI mining problem have also been extensively studied in the literature. These variants include: On-shelf patterns [32], sequential utility patterns [33], high average utility patterns [34], top-k utility patterns [25], [35], and HUIs with negative utilities [36] and multiple utility thresholds [37].

The existing HUI methods and their variants primarily consider unlabeled transaction data as primary input for discovering patterns. They do not consider supervised labels (or outcome variable) as part of the mining process. Therefore, a direct application of the HUIs to a supervised learning problem would not help uncover patterns that are useful for supervised learning. Besides, no prior work has explored the use of HUIs for supervised learning problems. This paper makes a first attempt and presents a novel approach to effectively exploit the supervised label information and then discover a new class of patterns (named, High Utility Gain patterns) that aid the downstream supervised learning task. The proposed approach, as we demonstrate in our experimental results section, offers superior learning performance as well as interpretability.

### B. INTERPRETABLE MACHINE LEARNING

While there is no mathematical definition of interpretability, it is commonly defined as the degree to which a human observer can understand the cause of a decision [38]. Interpretability in ML can be achieved either with inherently interpretable models or via post-hoc explanations generated using surrogate explainable models [15], [39]. A comprehensive survey of explainable models can be referred to in [40], [41]. We limit our review to the inherently interpretable models that are closely related to the current research study.

Table 1 provides a comparative analysis of interpretable machine learning models. The analysis is performed on six aspects: (a) nature and type of benchmark data used, (b) underlying models and data transformations performed as part of feature engineering, (c) nature of interpretable models used (e.g. rule-based, regression-based), (d) outcome variable supported (e.g. binary, multi-class), (e) model assessments considered, and (f) deployment related factors.

Supervised Assisted Feature Extraction (SAFE) [9] performs automated feature engineering by relying on advanced blackbox models such as random forest and gradient boosting. Their approach creates transformations of numeric and categorical variables using Partial Dependence Plot (PDP) profiles and hierarchical clustering. The transformed data is then used to build an interpretable model. SAFE model is tested on 30 different binary classification datasets with no missing values. The performance of the model is assessed using AUC measure. The key limitation of the method is that

**TABLE 1. Comparative analysis of interpretable machine learning models.**

| Description | Interpretable Learning Models | | | | | |
|---|---|---|---|---|---|---|
| | SAFE [9] | PLTR [15] | INAFEN [16] | CBA [19] CMAR [20] CPAR [21] | FRL [27] | HUG-IML* |
| Benchmark Data | | | | | | |
| main source | OpenML | SAS, UCI Kaggle | UCI Kaggle | UCI | UCI Hospital | UCI Kaggle |
| no. of datasets | 30 | 4 | 10 | 26 | 6 | 40 |
| max rows | 45300 | 150000 | 150000 | 5000 | 8000 | 284807 |
| max predictors | 971 | 23 | 20 | 60 | 57 | 93 |
| median rows | 1415 | 17980 | 900 | 529 | 3164.5 | 4388 |
| median predictors | 21 | 13 | 14.5 | 16 | 28 | 11.5 |
| Feature Engineering | | | | | | |
| underlying model | GB | DT | XGB | ARM | Rules | HUG |
| numerical | PDP, PELT | | DT, ARM | | Bayesian | |
| nominal | HC | | ARM | | | |
| Interpretable Model | LR | LR | LR | Association Rules | IF-THEN Rules | LR BNB |
| Outcome variable | | | | | | |
| binary | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| multi-class | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| max no. of classes | ✗ | ✗ | ✗ | 10 | ✗ | 26 |
| Model Assessment | | | | | | |
| ① Prediction quality | | | | | | |
| CV folds | Unspecified | 5 x 2-folds | 5 | 10 | 5 | 10 |
| compare ensembles? | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| multi-dimensional | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ |
| measures | AUC | AUC, BS, KS PGI, PCC | AUC, $F_1$ BS | Acc. | AUC | AUC, $F_1$ Acc., Hm, LL |
| ② Computational | | | | | | |
| compare scalability? (ensembles, IMs) | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| bottlenecks | HC PDP | Exhaustive 2nd order effects | Sensitive to support and confidence thresholds | | SA heuristic | limited with use of top-k |
| ③ Interpretability | | | | | | |
| compare ensembles? | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| compare IMs? | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| objective measures | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| Model Deployment | Computational overhead renders it practically unsafe | Generalizability concerns due to limited assessment | Patterns unoptimized for downstream learning task | | Compromises prediction quality | HUG patterns optimized for prediction task |

DT: Decision Tree LR: Logistic Regression (X)GB: (eXtreme) Gradient Boosting ARM: Association Rule Mining BNB: Binomial Naive Bayes
PDP: Partial Dependence Plot PELT [28]: Pruned Exact Linear Time HC: Hierarchical Clustering HUG: High-Utility Gain patterns
CV: Cross-Validation AUC: Area Under the Curve $F_1$: F-measure Acc.: Accuracy Hm: H-measure LL: Log Loss BS: Brier Score
PGI: Partial Gini Index PCC: Percentage of Correct Classification KS: Kolmogorov-Smirnov statistic
SA: Simulated Annealing IM: Interpretable Methods HUG-IML*: High Utility Gain pattern-based Interpretable Machine Learning (Our work)

it does not consider feature interactions. The method is also computationally expensive as it uses Hierarchical Clustering (HC) as part of feature engineering. The time complexity of the HC is at least quadratic on the size of the data. We also demonstrate through our experimental evaluation that the scalability of the method is poor.

Dumitrescu et al [15] present Penalized Logistic Tree Regression (PLTR). The algorithm primarily uses rules extracted from short-depth decision trees as predictors in a penalized logistic regression model. The authors evaluate their approach on 4 credit scoring datasets and demonstrate that their interpretable model offers performance comparable to that of random forest. The authors also show that PLTR avoids overfitting prevalent in non-linear logistic regression (with quadratic and interaction terms) by capturing only the relevant feature interactions.

Liu et al [16] present INterpretable Automated Feature ENgineering (INAFEN) that extends the core ideas presented in SAFE and PLTR. More specifically, INAFEN transforms features using decision trees and then mines feature interactions using association rules. It also predicts soft target using a black-box model. Subsequently, a logistic regression model is built on the transformed features and the predicted soft target. Their model is evaluated on 10 different binary classification datasets using a variety of performance measures (AUC, $F_1$ measure, and Brier score). One of the key limitations of

the model is its sensitivity to the itemset and rule mining thresholds, and the consequent computational overhead. We support our claims through experimental evaluation of our approach against INAFEN on diverse performance measures.

Associative classifiers such as Classification Based on Associations (CBA) [19], Classification based on Multiple Association Rules (CMAR) [20], and Classification based on Predictive Association Rules (CPAR) [21] apply constrained association rule mining using support-confidence framework. The constraints limit rules with class label as the consequent. These methods suffer from multiple issues: (a) support-confidence framework is limited in capturing interesting rules that influence classifier performance, and (b) generate a large number of rules that affects the readability and interpretability. SigDirect [23] learns statistically dependent rules using Fisher's test as a significance measure. The method also does not require setting minimum support and minimum confidence thresholds. The authors evaluate their approach on 20 different datasets against multiple associative and rule based methods.

Wang and Rudin [27] propose Falling Rule Lists (FRL) that generates an ordered list of if-then rules using a Bayesian approach. The approach is highly interpretable and customized for a healthcare application that predicts patient re-admissions. The method is designed for better interpretability. But, the performance of FRL is not at par with ensemble models.

Our work aims to explore a novel intepretable learning approach that is distinct from decision trees, rule lists, associative classifiers or ensemble model based feature extraction commonly studied in the literature. More specifically, our work investigates the use of a variant of high utility patterns for interpretable machine learning. We introduce the notion of utilities for a supervised learning problem, adapt HUI mining methods for the learning task, and show that our approach can be valuable for interpretable machine learning through rigorous experimental evaluation and case studies. The key differences of our approach against the related interpretable machine learning models are summarized in Table 1.

**TABLE 2.** Notations.

| Symbol | Descriptions |
|---|---|
| $\mathscr{D}$ | Input database |
| $\ddot{\mathscr{D}}$ | Database with generated utilities |
| $\widetilde{\mathscr{D}}$ | HUG transformed representation of $\ddot{\mathscr{D}}$ |
| $\mathscr{D}^{\mathrm{tr}}(\mathscr{D}^{\mathrm{tst}})$ | Train (test) data from $\mathscr{D}$ |
| $\ddot{\mathscr{D}}^{\mathrm{tr}}(\ddot{\mathscr{D}}^{\mathrm{tst}})$ | Train (test) data from $\ddot{\mathscr{D}}$ |
| $\widetilde{\mathscr{D}}^{\mathrm{tr}}(\widetilde{\mathscr{D}}^{\mathrm{tst}})$ | Train (test) data from $\widetilde{\mathscr{D}}$ |
| C | Distinct set of class labels $\{0, 1, ...\}$ |
| $\mathrm{y}^{(\mathrm{i})}$ | Class label of instance i |
| $\mathrm{y}^{\mathrm{i,tr}}(\mathrm{y}^{\mathrm{i,tst}})$ | Label of instance i in train (test) data |
| $\mathrm{x}^{(\mathrm{i})}$ | Predictors of instance i |
| $\mathrm{x}^{\mathrm{i,tr}}(\mathrm{x}^{\mathrm{i,tst}})$ | Predictors of instance i in train (test) data |
| f | Predictor variable f |
| $\mathrm{x}^{\mathrm{i}}_{\mathrm{f}}$ | Predictor f for instance $\mathrm{x}^{\mathrm{i}}$ |
| d | Dimensionality of predictors or features |
| $\mathscr{B}$ | Number of bins used for discretization |
| $\mathrm{br}_{\mathrm{f}}$ | Range of bin values for feature f |
| $\mathrm{cat}[\mathrm{x}^{(\mathrm{i})}_{\mathrm{f}}]$ | Encoded category of f for instance i |
| $\mathrm{x}_{\mathrm{fk}}$ | Predictor f assigned to bin or category k |
| I | Set of all items $\mathrm{x}_{11}, \mathrm{x}_{12}...$ |
| $\mathrm{T}_{\mathrm{i}}$ | Transaction $\mathrm{T}_{\mathrm{i}} \subseteq \mathrm{I}$ |
| $\sigma_{\mathrm{fy}}$ (or $\sigma_{\mathrm{f}}$) | Correlation between f and target y |
| NMI | Normalized Mutual Information |
| $\mathrm{NMI}(\mathrm{x_f}, \mathrm{y})$ | NMI between predictor f and target y |
| $\mathrm{w_c}$ | Weights for class $\mathrm{c} \in \mathrm{C}$ |
| $\mathrm{EU}(\mathrm{x_f})$ | External utility of predictor f |
| $\mathrm{IU}(\mathrm{x_f}, \mathrm{T_i})$ | Internal utility of predictor f in $\mathrm{T_i}$ |
| $\mathrm{U}(\mathrm{x_f}, \mathrm{T_i})$ | Utility of an item $\mathrm{x_f}$ in $\mathrm{T_i}$ |
| $\mathrm{U}(\mathrm{X}, \mathrm{T_i})$ | Utility of an itemset X in $\mathrm{T_i}$ |
| $\mathrm{Z_c}$ | Normalizing constant |
| U(X) | Utility of an itemset X |
| IG(X) | Information gain of an itemset X |
| minU | Minimum utility threhsold value |
| HUI | Set of High Utility Itemsets |
| $\mathrm{HUI_l}$ | Set of HUIs of specific length l |
| $\mathscr{L}$ | Maximum length of pattern in HUI |
| $\mathscr{G}$ | Information gain threshold value |
| topkHUI | Set of top-k HUIs |
| $\mathrm{topkHUI}_{\mathscr{L}}$ | Length constrained set of top-k HUIs |
| HUG | Set of High Uility Gain patterns |
| $\lambda$ | Regularization parameter |
| $\beta_{\mathrm{lp}}$ | Coefficients of HUG pattern p of length l |

## III. NOTATIONS AND DEFINITIONS

We first describe the key notations used throughout this paper. We also introduce several definitions that are related to data transformation, utility patterns, and supervised learning process. The summary of the notations can be referred to in Table 2.

Let $\mathscr{D}$ represent a database that contains the input supervised learning examples. Each observation or instance in $\mathscr{D}$ contains predictors and a known class label, $(x^{(i)}, y^{(i)})$, where $i$ refers to a specific example and its value ranges from $1, 2, \ldots m$. Let $C = \{0, 1, \ldots\}$ denote the distinct set of class labels. Then, $y^{(i)} \in C$ is the target or label of a particular data instance $i$. Let the training and test instances of $\mathscr{D}$, $x^i$, and $y^i$ be denoted as $\mathscr{D}^{tr}$, $\mathscr{D}^{tst}$, $x^{i,tr}$, $x^{i,tst}$, $y^{i,tr}$, $y^{i,tst}$. Each $x^{(i)}$ has a set of $d$ predictors or features. Let $x^i_f$ denote the feature $f$ for a given instance $x^i$.

The examples in $\mathscr{D}$ can be viewed as a set comprising $X$ matrix of size $m \times d$ and a vector of class labels $y$. Without loss of generality, we apply a label encoding transformation of the target variable and assign integer labels in descending order of the size of examples in each class. Intuitively, we assign

**TABLE 3.** Sample supervised learning database ($\mathscr{D}$).

| | $\mathrm{x}_1$ | $\mathrm{x}_2$ | $\mathrm{x}_3$ | $\mathrm{x}_4$ | $\mathrm{x}_5$ | y |
|---|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 2.5 | M | 1 |
| 2 | 4.9 | 3.0 | 1.4 | 1.2 | M | 1 |
| 3 | 6.4 | 2.9 | 4.4 | 0.4 | F | 1 |
| 4 | 5.4 | 2.4 | 1.5 | 0.2 | M | 1 |
| 5 | 6.1 | 2.2 | 5.0 | 2.5 | F | 2 |
| 6 | 7.9 | 2.5 | 6.4 | 2.0 | F | 2 |
| 7 | 8.1 | 2.3 | 6.9 | 1.2 | M | 2 |
| 8 | 4.1 | 2.2 | 6.9 | 2.2 | F | 1 |
| 9 | 8.1 | 2.3 | 6.9 | 1.2 | F | 2 |
| 10 | 6.1 | 2.1 | 1.5 | 3.2 | F | 1 |

**TABLE 4.** Database with encoded features.

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5 = F$ | $x_5 = M$ | y |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | 1 | 4 | 0 | 1 | 1 |
| 2 | 1 | 4 | 1 | 2 | 0 | 1 | 1 |
| 3 | 3 | 4 | 2 | 1 | 1 | 0 | 1 |
| 4 | 2 | 3 | 2 | 1 | 0 | 1 | 1 |
| 5 | 3 | 1 | 3 | 4 | 1 | 0 | 2 |
| 6 | 4 | 3 | 3 | 3 | 1 | 0 | 2 |
| 7 | 4 | 2 | 4 | 2 | 0 | 1 | 2 |
| 8 | 1 | 1 | 4 | 3 | 1 | 0 | 1 |
| 9 | 4 | 2 | 4 | 2 | 1 | 0 | 2 |
| 10 | 3 | 1 | 2 | 4 | 1 | 0 | 1 |

greater importance to minority classes. This is in line with most real-world problems where the data is often highly imbalanced (e.g. fraud detection) and deciphering patterns from minority classes have a greater practical significance.

For the illustration, let us consider a simple credit lending decision problem. In this problem, the predictors for a specific customer ($x^{(i)}$) could include variables such as age, income, gender, number of personal loans, and number of credit cards. The target variable ($y^{(i)}$) is whether the selected customer has a good or bad credit (1 or 0). A sample classifier learning database $\mathscr{D}$ is given in Table 3.

The proposed HUG model performs quantile discretization (of the min-max normalized numerical) and one-hot encoding (of the categorical) predictor variables as part of initial data preparation. Let the number of bins used for discretization be denoted as $\mathscr{B}$. Let $cat[x_f^{(i)}]$ denote the specific encoded bin or category of feature ($f$) for example $i$. Let $br_f$ denote the bin ranges for feature $f$. For the running example with $\mathscr{B} = 4$, the bin ranges of the min-max normalized variables $x_1$ and $x_2$ are: $br_1 = [(0, 0.269), (0.269, 0.5), (0.5, 0.856), (0.856, 1.0)]$, $br_2 = [(0, 0.0893), (0.0893, 0.179), (0.179, 0.5), (0.5, 1.0)]$.

The discretized and one-hot encoded version of the database $D$ is shown in Table 4.

Let the correlation coefficient between the predictors and target variable ($y$) be denoted as $\sigma_{fy}$. For the running example, the correlation values (computed from Table 4) are $\sigma_{1y} = 0.7824, \sigma_{2y} = -0.3390, \sigma_{3y} = 0.6599, \sigma_{4y} = 0.1099, \sigma_{5y(F)} = 0.25, \sigma_{5y(M)} = -0.25$.

Normalized Mutual Information (NMI), a commonly used information theoretic concept, measures the mutual dependence between two variables and ranges between 0 (no mutual dependence) and 1 (perfect association). Let the NMI between the categorical predictors and the target variable $y$ be denoted as $NMI(x_f, y)$. For the running example, $NMI(x_5, y) = 0.0478$.

*Definition 1:* Each predictor $x_f$ is assigned an external utility value, referred to as $EU(x_f)$. In the supervised learning context, we propose the use of feature correlations ($\sigma_{fy}$) and Normalized Mutual Information ($NMI$) as proxies for predictor utilities. Additionally, the class weights, denoted as $w_c$, are considered. More specifically, the external utilities are computed for each feature and class $c \in C$ as:

$$\begin{aligned} EU(x_f, c) &= abs(\sigma_{fy}) * w_c, \quad if \ \ f \ \ is \ \ numerical \\ &= NMI(x_f, y) * w_c, \quad o.w. \end{aligned} \tag{1}$$

The default class weights ($w$) for each class is assumed as 1. Our model implicitly captures class level variations through ordinal transformation of the target variable based on the size of examples in each class. This obviates the need for explicit class imbalance treatments with sampling or class weighting schemes. Our model incorporates the class weighting scheme to offer additional flexibility to handle unique class specific requirements. For the running example, the computed $EU$ values are shown in Table 5.

*Definition 2:* The discretized (or encoded) item $x_f \in T_i$ is assigned an internal utility value, referred to as $IU(x_f, T_i)$. The internal utilities for numeric predictors are computed as:

$$\begin{aligned} IU(x_f, T_i) &= br_f[cat[x_f^{(i)}]].right, \quad if \ \ \sigma_f > 0 \\ &= br_f[b - cat[x_f^{(i)}] + 1].right, \quad o.w. \end{aligned} \tag{2}$$

$IU(x_1, T_1) = br_1[cat[x_1^{(1)}]].right = 0.269$ for the running example. Similarly, $IU(x_2, T_5) = br_2[4 - cat[x_2^{(5)}] + 1].right = br_2[4 - 1 + 1].right = br_2[4].right = 1.0$.

Note that the bin edges are chosen in reverse order for negatively correlated features. This heuristic primarily enables identification of the right set of patterns through utility mining for the downstream learning task.

The internal utilities for the one-hot encoded categorical predictors are computed as:

$$\begin{aligned} IU(x_f, T_i) &= 1, \quad if \ \ \sigma_{fy} > 0 \\ &= 0.05, \quad o.w. \end{aligned} \tag{3}$$

The utility weighting scheme in equation 3 gives higher weights to the positively correlated feature. This allows extraction of relevant utility patterns for the downstream learning task. For the running example, the $x_5 = F$ has a positive correlation, and hence its internal utility will be set to 1. On the other hand, the transactions with $x_5 = M$ will be assigned an internal utility value of 0.05.

*Definition 3:* The utility of an item $x_f \in T_i$, denoted as $U(x_f, T_i)$, is computed as the normalized product of external and internal utilities of items in the transaction, $T_i$. That is,

$$U(x_f, T_i) = \frac{1}{Z_c} \ EU(x_f, c) \cdot IU(x_f, T_i) \tag{4}$$

$c$ is the class label of the transaction $T_i$. The denominator $Z_c$ is the normalizing constant and is computed as the maximum of the numerator values over all the features for each class $c$ ($\in C$).

The discretized database $D$ is transformed to a transaction list format with associated utility information for mining utility patterns. The items in the transaction list are the encoded

**TABLE 5.** Item external utilities.

| Item | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| EU | 0.7824 | 0.3390 | 0.6599 | 0.109 |

**TABLE 6. Transaction database with internal utilites.**

| TID | Transaction | IU |
|---|---|---|
| $T_1$ | $x_{11}, x_{24}, x_{31}, x_{44}$, M | 0.269,0.089,0.018,1.000,0.05 |
| $T_2$ | $x_{11}, x_{24}, x_{31}, x_{42}$, M | 0.269,0.089,0.018,0.467,0.05 |
| $T_3$ | $x_{13}, x_{24}, x_{32}, x_{41}$, F | 0.856,0.089,0.600,0.333,1.00 |
| $T_4$ | $x_{12}, x_{23}, x_{32}, x_{41}$, M | 0.500,0.179,0.600,0.333,0.05 |
| $T_5$ | $x_{13}, x_{21}, x_{33}, x_{44}$, F | 0.856,1.000,0.977,1.000,1.00 |
| $T_6$ | $x_{14}, x_{23}, x_{33}, x_{43}$, F | 1.000,0.179,0.977,0.742,1.00 |
| $T_7$ | $x_{14}, x_{22}, x_{34}, x_{42}$, M | 1.000,0.500,1.000,0.467,0.05 |
| $T_8$ | $x_{11}, x_{21}, x_{34}, x_{43}$, F | 0.269,1.000,1.000,0.742,1.00 |
| $T_9$ | $x_{14}, x_{22}, x_{34}, x_{42}$, F | 1.000,0.500,1.000,0.467,1.00 |
| $T_{10}$ | $x_{13}, x_{21}, x_{32}, x_{44}$, F | 0.856,1.000,0.600,1.000,1.00 |

features in our model. That is, $I = \{x_{11}, x_{12} \ldots x_{d\mathscr{B}}\}$, where $\mathscr{B}$ is the number of discretized bins or categories of individual features $(1..d)$. A transaction $T_i = \{t_l | l = 1, 2 \ldots n_i, \forall t_l \in I\}$, where $n_i$ is the number of items in transaction $T_i$. The transactional data for the running example is shown in Table 6. Using the computed internal and the external utilities, overall utilities of each item in each transaction are then computed by applying equation 4. The resulting transaction database with utility information (denoted as $\ddot{\mathscr{D}}$) is used for utility mining.

*Definition 4:* The utility of an itemset $X$ in transaction $T_i$ $(X \subseteq T_i)$ is denoted as $U(X, T_i)$, and is defined as:

$$U(X, T_i) = \sum_{x_f \in X} U(x_f, T_i) \tag{5}$$

*Definition 5:* The utility of an itemset is denoted as $U(X)$, and is defined as:

$$U(X) = \sum_{X \subseteq T_i} U(X, T_i) \tag{6}$$

*Definition 6:* Let us denote the information gain of an itemset as $IG(X)$. This is a standard measure used in information theory and machine learning. It is computed as the difference in entropy at the parent node (i.e. the itemset $X$'s immediate ancestor) and the current node $(X)$.

*Definition 7:* High Utility Itemsets ($HUI$) are set of itemsets whose utility threshold values are above an user-defined utility threshold $minU$. More formally,

$$HUI = \{X : U(X) | X \subseteq I, U(X) \geq minU\} \tag{7}$$

*Definition 8:* Top-K High Utility Itemsets are the set of all k-$HUI$s with the highest utility values, and denoted as $topkHUI$.

*Definition 9:* Let the pattern length and the user-defined maximum $HUI$ pattern length be denoted respectively as $HUI_l$ and $\mathscr{L}$.

If a user-defined maximum pattern length is set, our model generates only the $HUI$s whose lengths are less than or equal to $\mathscr{L}$. Let the length constrained set of top-k $HUI$s be denoted as $topkHUI_{\mathscr{L}}$.

*Definition 10:* High Utility Gain ($HUG$) patterns are the set of itemsets in top$kHUI_{\mathscr{L}}$ that satisfy the user-defined information gain threshold value, $\mathscr{G}$. That is,

$$HUG = \{X | X \in topkHUI_{\mathscr{L}} \; and \; IG(X) \geq \mathscr{G}\} \tag{8}$$

The original database $\mathscr{D}$ is transformed using the mined $HUG$ patterns, denoted as $\widetilde{\mathscr{D}}$, for interpretable machine learning. We describe our complete HUG modeling steps next using the key notations and definitions introduced in this section.

## IV. HUG-IML: AN INTERPRETABLE CLASSIFIER MODEL

The proposed High Utility Gain (HUG) Interpretable Machine Learning (IML) model consists of seven broad stages. The overall process workflow of the proposed model is shown in Figure 1. The individual stages of our model are described in the following pages.

For the running example, let us assume $\mathscr{B} = 4, \mathscr{L} = 1, \mathscr{G} = 0.2$ for the three parameters used in our model.

### A. STAGE 1: PREPARE DATA

Our model takes input learning examples $\mathscr{D}$ as input. Basic data preparation operations performed on $\mathscr{D}$ include: deletion of rows with a large number of missing values, removal of duplicate rows, missing value replacement, and label encoding of target variable. Missing values are replaced using median (numerical features) and mode (categorical features). For the target variable $y$, labels or integer values are assigned in descending order of the size of the examples in each class, $c \in C$.

After the basic data preparation, a stratified sample of train and test data are generated. In our experiments, we used a stratified, ten-fold cross-validation to ensure robustness of our model results.

**TABLE 7. Transaction list data with utility information.**

| TID | Transaction | Utility (U) |
|---|---|---|
| $T_1$ | $x_{11}, x_{24}, x_{31}, x_{44}$, M | 0.269,0.038,0.015,0.140,0.003 |
| $T_2$ | $x_{11}, x_{24}, x_{31}, x_{42}$, M | 0.269,0.038,0.015,0.065,0.003 |
| $T_3$ | $x_{13}, x_{24}, x_{32}, x_{41}$, F | 0.856,0.038,0.506,0.046,0.061 |
| $T_4$ | $x_{12}, x_{23}, x_{32}, x_{41}$, M | 0.500,0.077,0.506,0.046,0.003 |
| $T_5$ | $x_{13}, x_{21}, x_{33}, x_{44}$, F | 0.856,0.433,0.823,0.140,0.061 |
| $T_6$ | $x_{14}, x_{23}, x_{33}, x_{43}$, F | 1.000,0.077,0.823,0.104,0.061 |
| $T_7$ | $x_{14}, x_{22}, x_{34}, x_{42}$, M | 1.000,0.216,0.843,0.065,0.003 |
| $T_8$ | $x_{11}, x_{21}, x_{34}, x_{43}$, F | 0.269,0.433,0.843,0.104,0.061 |
| $T_9$ | $x_{14}, x_{22}, x_{34}, x_{42}$, F | 1.000,0.216,0.843,0.065,0.061 |
| $T_{10}$ | $x_{13}, x_{21}, x_{32}, x_{44}$, F | 0.856,0.433,0.506,0.140,0.061 |

### B. STAGE 2: CONSTRUCT TRANSACTIONS WITH UTILITY FOR PATTERN MINING

In the second stage, our model applies discretization and one-hot encoding of features. For discretiztion, a quantile binning is applied using the user-defined number of bins $\mathscr{B}$ without considering the labels $y$. Future work may consider alternate discretization methods designed for supervised learning task [42]. For the sample database shown in Table 3, the discretized and one-hot encoded output is given in Table 4.

The internal, external, and overall utilities of items (discretized or one-hot encoded features) are computed using equations 1 to 4. For the running example, the computed internal utility($IU$) values at the transaction level are shown in Table 6.

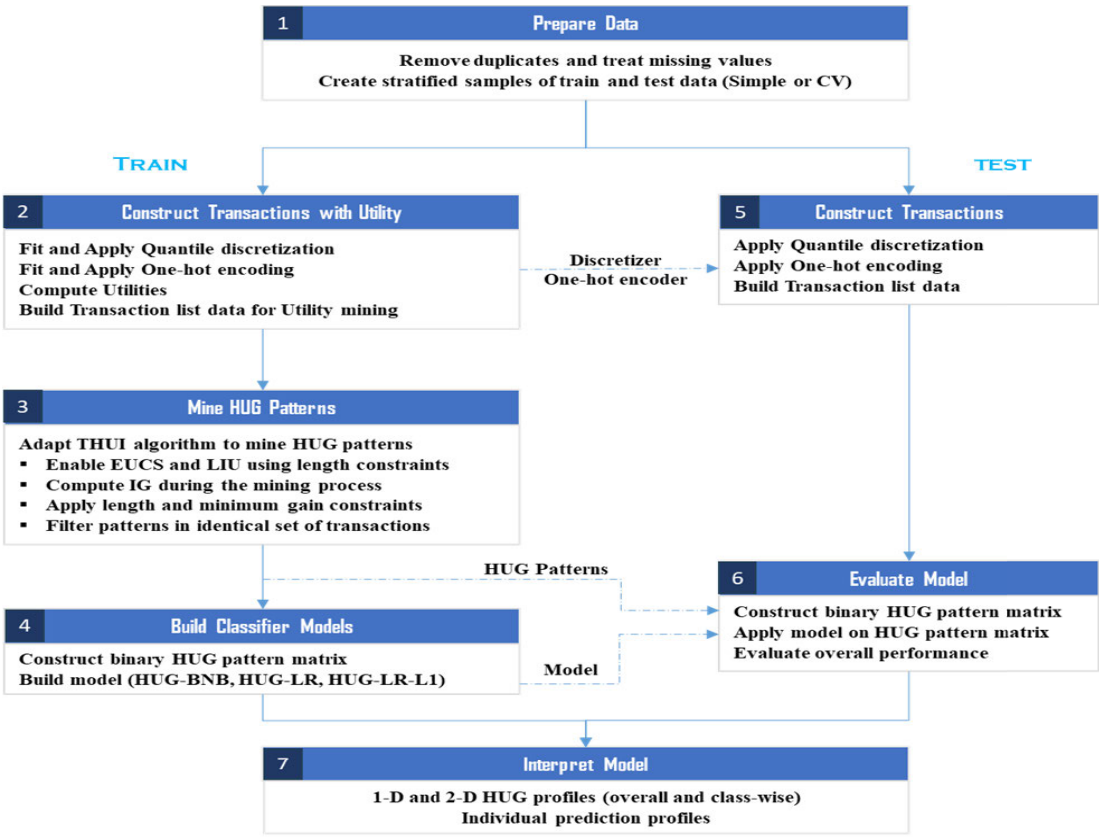


**FIGURE 1.** HUG-IML process Workflow.

The transaction level overall utility information ($\ddot{\mathscr{D}}^{tr}$) is computed using equation 4 and an illustration for the running example is provided in Table 7.

### C. STAGE 3: MINE HUG PATTERNS

We mine the HUG patterns from the training data and then transform the test data using the discovered patterns in subsequent steps. The proposed algorithm for mining HUG patterns is described in the following pages.

High utility itemset mining requires specification of the minimum utility ($minU$) values. However, the minimum utility values ($minU$) are often difficult to determine for individual datasets. Hence, we use a top-k HUI mining algorithm (THUI [25]) and adapt it to mine *HUG* patterns. The k value in our model is specified as the function of the pattern length $\mathscr{L}$ (default value is set as $\binom{|I|}{\mathscr{L}}$). We applied the following major changes to the THUI algorithm for mining *HUG* patterns introduced in this paper:

1) *Threshold raising strategies.* The proposed *HUG* model imposes pattern length constraints ($\mathscr{L}$). Our model handles its impact on threshold raising strategies used in THUI (EUCS and LIU), by applying suitable constraints. More specifically, EUCS strategy is enabled if and only if $\mathscr{L} >= 2$. Similarly, the LIU strategy is enabled if and only if the length is unconstrained.
2) *Information gain.* The proposed model computes information gain measure during the mining process. We first compute entropy at each node during the depth first traversal of the THUI algorithm. The information gain value is then computed as the difference in entropy at the parent and the child nodes. It is to be noted that at each level of the THUI search tree exploration, only binary splits are made (given the binary nature of the items). This allows inferring entropy values of both the branches of the child node by visiting just one branch of the search tree.
3) *Additional pruning rules.* During the search tree exploration, the pattern length ($\mathscr{L}$) and minimum gain constraints ($\mathscr{G}$) are applied to limit the search space.
4) *Duplicate patterns.* The proposed HUG mining algorithm filter patterns that occur in identical set of transactions. It retains just one of the patterns with higher information gain. The dropped patterns are likely to be part of the Rashomon set [11] and may potentially be used for generating alternate model explanations.

For the running example with $\mathscr{L} = 1$ and $\mathscr{G} = 0.2$, the HUG mining algorithm generates three *HUG* patterns of size 1: (1) $x_{14}$, (2) $x_{33}$, and (3) $x_{22}$. In the original data space, the generated patterns are equivalent to: $x_1 = [7.524 - 8.1]$, $x_3 = [4.7 - 6.773]$, $x_2 = [2.225 - 2.351]$.

**TABLE 8.** Binary HUG pattern matrix.

| TID | $x_{14}$ | $x_{33}$ | $x_{22}$ | y | TID | $x_{14}$ | $x_{33}$ | $x_{22}$ | y |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 6 | 1 | 1 | 0 | 2 |
| 2 | 0 | 0 | 0 | 1 | 7 | 1 | 0 | 1 | 2 |
| 3 | 0 | 0 | 0 | 1 | 8 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 1 | 9 | 1 | 0 | 1 | 2 |
| 5 | 0 | 1 | 0 | 2 | 10 | 0 | 0 | 0 | 1 |

### D. STAGE 4: BUILD CLASSIFIER MODEL

The mined *HUG* patterns are used to create a binary HUG pattern matrix ($\widetilde{\mathscr{D}}^{tr}$). The resulting binary matrix is shown in Table 8. In the actual implementation, the HUG pattern matrix is stored in sparse binary matrix format. The pseudo-code of the key steps of our interpretable classifier model is given in Algorithm 1.

The generated binary HUG pattern matrix is used as an input for classifier modeling. We used three simple models, namely, Binomial Naive Bayes (BNB), Logistic Regression (LR), and $L_1$ regularized Logistic Regression (LR-L1). We refer to the logistic regression model built on HUG pattern matrix as HUG-LR. The HUG-LR model primarily fits an additive model over the HUG patterns of different lengths. Mathematically,

$$y = \beta_0 + \sum_{l=1}^{\mathscr{L}} \sum_{p=1}^{|HUG_l|} \beta_{lp} * X_{lp} + \lambda * ||\beta||_1 \qquad (9)$$

where $HUG_l \in HUG | \forall X_p \in HUG, |X_p| = l$ and $X_{lp} \in HUG_l$. $\beta_{lp}$ is the learned coefficient for a *HUG* pattern of specific length $l$ ($X_{lp}$). The second term in the above expression is the $L_1$ regularization that is used to select sparse set of patterns and control overfitting. The default value of the regularization parameter ($\lambda$) is set to 0 (1) for the HUG-LR (HUG-LR-L1) models.

The size of $\mathscr{L}$, as we show through rigorous experimental evaluation, is very small ($\leq 2$) for most real-world problems. Besides, the size of *HUG* is likely to be small in comparison to standard non-linear LR models that require exhaustive enumeration of higher order terms. This is due to the fact that our model can effectively capture piece-wise linear relationships across predictors and target variables through novel application of utilities. It also makes our HUG model less prone to overfitting and help achieve high performance, scalability, and interpretability.

### E. STAGE 5: CONSTRUCT TRANSACTIONS FOR TEST DATA

The transaction construction process for the test data is similar to stage 2 of our model (refer to section IV-B). The key differences are: (1) the discretizer and one-hot encoding models fitted on training data are used for transformation, and (2) no utility information is computed. The output generated will resemble Table 7 with just columns 1 and 2.

### F. STAGE 6: EVALUATE MODEL ON TEST DATA

The stage six of our model takes three inputs (refer to Figure 1): the transaction list data generated from the test data, HUG patterns mined from the training data, and the classifier model. The first two inputs are used to construct the binary HUG pattern matrix ($\widetilde{\mathscr{D}}^{tst}$). The classifier model is then applied on $\widetilde{\mathscr{D}}^{tst}$ to predict outcomes. The model performance is comprehensively assessed on three dimensions: (1) quality of predictions using five diverse measures used in the literature: (2) computational performance, and (3) quantitative interpretability measures.

### G. STAGE 7: INTERPRET MODEL

The model interpretation is carried out at three levels. First, the overall model performance is assessed by analyzing probability score distributions, margin distributions [44], and HUG profiles. Second, class-wise analysis is conducted to assess class-level HUG patterns to generate model descriptions. One and two dimensional HUG profile analysis is introduced for generating model descriptions. Third, the individual instances are examined to understand the factors that drive prediction outcomes. Counterfactual analysis may also be conducted to study the factors that need to be changed for realizing alternate (or desired) outcomes. We illustrate the interpretability aspects of the proposed model with the help of three case studies in section VI.

## V. EXPERIMENTAL RESULTS

In this section, we first explain the details of our experiments in terms of the datasets used, model implementations, comparative ensemble models, related interpretable models, and performance measures. Subsequently, we assess the performance of our model and compare it with other related

**Algorithm 1** HUG-IML Classifier

**Input:** $\mathscr{D}$ : input transactional database
$\mathscr{B}$ : number of bins (computed from $\mathscr{D}^{tr}$)
$\mathscr{L}$ : maximum pattern length (default: 1)
$\mathscr{G}$ : information gain threshold (default: 1e-4)
**Output:** IML classifier model

Scan $\mathscr{D}$ and prepare data //stage 1
Generate train and test data $\mathscr{D}^{tr}$, $\mathscr{D}^{tst}$
$qdModel \leftarrow$ Quantile-discretizer($\mathscr{D}^{tr}$, $\mathscr{B}$) //stage 2
$oheModel \leftarrow$ One-hot-encoder($\mathscr{D}^{tr}$)
Compute $EU$, $IU$, and $U$
Build transaction list with utility $\ddot{\mathscr{D}}^{tr}$ (e.g. Table 7)
$HUG \leftarrow$ Mine-HUG($\ddot{\mathscr{D}}^{tr}$, $\mathscr{L}$, $\mathscr{G}$) //stage 3
$\widetilde{\mathscr{D}}^{tr} \leftarrow$ construct($\ddot{\mathscr{D}}^{tr}$, $HUG$) //stage 4
$hugModel \leftarrow$ Build-Model($\widetilde{\mathscr{D}}^{tr}$)
Discretize test data: $qdModel(\mathscr{D}^{tst})$ //stage 5
One hot encode test data: $oheModel(\mathscr{D}^{tst})$
Build transaction list $\ddot{\mathscr{D}}^{tst}$
$\widetilde{\mathscr{D}}^{tst} \leftarrow$ construct($\ddot{\mathscr{D}}^{tst}$, $HUG$) //stage 6
Apply model on test data: $hugModel(\widetilde{\mathscr{D}}^{tst})$
Evaluate model performance
**return** $hugModel$

**TABLE 9.** Description of benchmark datasets.

| Dno | Name | \|R\| | \|F\| (\|n\|) | \|C\| | $\|C_m\|$ | Dno | Name | \|R\| | \|F\| (\|n\|) | \|C\| | $\|C_m\|$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| B1 | sonar | 208 | 60 (0) | 2 | 97 | B21 | lending club | 9578 | 13 (3) | 2 | 1533 |
| B2 | breast cancer | 286 | 9 (9) | 2 | 85 | B22 | heloc | 10459 | 23 (0) | 2 | 5000 |
| B3 | congressional voting | 435 | 14 (14) | 2 | 168 | B23 | vehicle insurance | 15420 | 31 (29) | 2 | 923 |
| B4 | breast cancer (wisc.) | 569 | 30 (0) | 2 | 212 | B24 | magic GT | 19020 | 10 (0) | 2 | 6688 |
| B5 | ILPD | 583 | 10 (1) | 2 | 167 | B25 | car fraud claim | 17998 | 21 (12) | 2 | 2816 |
| B6 | credit approval | 690 | 15 (9) | 2 | 307 | B26 | default credit card | 30000 | 23 (9) | 2 | 6636 |
| B7 | blood transfusion | 748 | 4 (0) | 2 | 178 | B27 | adult income | 30718 | 9 (6) | 2 | 7650 |
| B8 | pima | 768 | 8 (0) | 2 | 268 | B28 | bank marketing | 41188 | 15 (6) | 2 | 4640 |
| B9 | titanic | 891 | 6 (3) | 2 | 342 | B29 | give me credit | 150000 | 10 (0) | 2 | 10026 |
| B10 | tictactoe | 958 | 9 (9) | 2 | 332 | B30 | cc wordline ULB | 284807 | 29 (0) | 2 | 492 |
| B11 | german credit | 1000 | 30 (27) | 2 | 300 | M31 | lung cancer | 32 | 56 (56) | 3 | 9 |
| B12 | bank note | 1372 | 4 (0) | 2 | 610 | M32 | iris | 150 | 4 (0) | 3 | 50 |
| B13 | telecom iranian | 3150 | 12 (4) | 2 | 495 | M33 | wine | 178 | 13 (0) | 3 | 48 |
| B14 | abalone | 4175 | 8 (1) | 2 | 1447 | M34 | wheat seeds | 210 | 7 (0) | 3 | 70 |
| B15 | spambase | 4601 | 57 (0) | 2 | 1813 | M35 | yeast | 1484 | 8 (2) | 10 | 5 |
| B16 | telecom churn | 5000 | 9 (4) | 2 | 707 | M36 | car | 1728 | 6 (6) | 4 | 65 |
| B17 | universal bank | 5000 | 11 (6) | 2 | 480 | M37 | wine quality | 4898 | 11 (0) | 7 | 5 |
| B18 | COMPAS | 6172 | 11 (10) | 2 | 2809 | M38 | digits | 5620 | 64 (0) | 10 | 554 |
| B19 | bankruptcy | 6819 | 93 (0) | 2 | 220 | M39 | nursery | 12960 | 8 (8) | 5 | 2 |
| B20 | mushroom | 8124 | 20 (20) | 2 | 3916 | M40 | letter | 20000 | 16 (0) | 26 | 734 |

|R|: Number of records |C|: Number of classes |F| (|n|): Total (nominal) features $|C_m|$: Size of the minority class

**TABLE 10.** HUG-model parameters.

| Dno | $\mathscr{B}$ | $\mathscr{L}$ | $\mathscr{G}$ | Dno | $\mathscr{B}$ | $\mathscr{L}$ | $\mathscr{G}$ |
|---|---|---|---|---|---|---|---|
| B1 | 4 | 2 | 5e-3 | B2 | - | 1 | 1e-3 |
| B3 | - | 1 | 1e-4 | B4 | 6 | 1 | 1e-2 |
| B5 | 10 | 1 | 1e-3 | B6 | 6 | 1 | 1e-3 |
| B7 | 4 | 1 | 1e-4 | B8 | 7 | 1 | 5e-3 |
| B9 | 15 | 2 | 6e-4 | B10 | - | 3 | 1.5e-3 |
| B11 | 15 | 1 | 1e-3 | B12 | 11 | 1 | 1e-7 |
| B13 | 9 | 2 | 1e-6 | B14 | 14 | 1 | 1e-4 |
| B15 | 40 | 1 | 1e-5 | B16 | 8 | 2 | 1e-5 |
| B17 | 11 | 2 | 1e-4 | B18 | 7 | 1 | 1e-4 |
| B19 | 7 | 1 | 3e-3 | B20 | - | 1 | 1e-4 |
| B21 | 13 | 1 | 1e-5 | B22 | 8 | 1 | 1e-3 |
| B23 | 10 | 2 | 1e-4 | B24 | 18 | 1 | 1e-6 |
| B25 | 10 | 1 | 1e-4 | B26 | 10 | 1 | 1e-4 |
| B27 | 6 | 2 | 1e-4 | B28 | 11 | 2 | 3e-3 |
| B29 | 30 | 2 | 1e-4 | B30 | 10 | 2 | 1e-4 |
| M31 | - | 2 | 2e-1 | M32 | 14 | 1 | 1e-2 |
| M33 | 4 | 1 | 1e-2 | M34 | 6 | 2 | 3e-2 |
| M35 | 9 | 1 | 1e-4 | M36 | - | 3 | 1e-4 |
| M37 | 21 | 2 | 9e-4 | M38 | 4 | 2 | 5e-2 |
| M39 | - | 3 | 1e-4 | M40 | 11 | 2 | 1e-3 |

ensemble and interpretable models. We then share our key observations and insights.

### A. EXPERIMENTAL DESIGN

In our experiments, we used forty benchmark datasets with varying characteristics in terms of its size, number of predictors, and the mixture of numerical and categorical attributes. Thirty of these datasets pertain to binary classification tasks. The imbalanced ratio of the datasets range from a low of 2 (mushroom dataset) to 579 (for cc wordline ULB dataset). The remaining ten datasets are multi-class datasets with up to 26 distinct classes. The datasets were obtained from the UCI machine learning repository [45] and Kaggle.[1]

The characteristics of the benchmark datasets are shown in Table 9. The fourth column in the table gives the total features and the count of nominal or categorical features. The last column gives the size of the class with the least number of examples.

The proposed model was implemented using python scripts.[2] We also used standard open source machine learning packages (sklearn,[3] xgboost[4]) for the implementation and comparison with the state-of-the-art ensemble models.

The HUG patterns were mined by adapting the THUI [25] algorithm written in Java programming language. The mined HUG patterns were used to transform the train and test data as described in section IV. The transformed data matrices (i.e. $\widetilde{\mathscr{D}}^{tr}$, $\widetilde{\mathscr{D}}^{tst}$)) were highly sparse. More specifically, the sparsity of the HUG pattern transformed train (test) matrices ($\widetilde{\mathscr{D}}$) were found to be about 13% (12%) with a standard deviation of about 12% (11%) across all benchmark datasets. Therefore, to optimize memory utilization, a sparse matrix representation was used in our implementation though the running example (section IV) was illustrated with a dense matrix format (Table 8).

The comparative evaluation of the proposed three HUG models (HUG-BNB, HUG-LR, and HUG-LR-L1) were made against both the baseline (including ensembles) and interpretable classifier models. For the baseline models, we used: Logistic Regression (LR), Random Forest (RF) and eXtreme Gradient Boosting (XGB). For the interpretable models, we used: SAFE [9], and INAFEN [16]. The publicly available code shared by the authors were used for the comparative evaluation.

A k-fold stratified cross-validation was performed to ensure robustness of the results. The k value was set to

[1] https://kaggle.com

[2] https://github.com/srikumar2050/hugiml

[3] https://scikit-learn.org/stable/

[4] https://xgboost.readthedocs.io/en/stable/

10 or the size of the class with the least number of examples ($|C_m|$). The learning algorithms such as Naive Bayes, Boosting, and Random forest do not generate well-calibrated probabilities [46] that are of importance to human decision making [47]. We, therefore, applied calibration methods to obtain well-calibrated probabilities. A sigmoid (isotonic) calibration method is applied to the small (large) datasets. It is to be noted that the isotonic calibration method is known to overfit on smaller datasets [46] and hence we applied sigmoid calibration for training data with less than 500 records.

The optuna library[5] was used for hyperparameter tuning and optimization. Our HUG models require specification of three parameters: (1) number of bins ($\mathscr{B}$), (2) maximum HUG pattern length ($\mathscr{L}$), and (3) information gain threshold value ($\mathscr{G}$). A grid search over a range of these parameter values were performed using the optuna library. The final parameters selected for our experiments are given in Table 10. The hyperparamter tuning was also done for the related interpretable methods. For example, INAFEN [16] requires specification of support and confidence threshold values. A grid search was performed for support (0.005-0.15) and confidence (0.5-0.9) thresholds and the best values were chosen based on validation performance.

The overall classifier model performance was assessed on three dimensions: (1) quality of predictions using five diverse classifier measures: Accuracy, $F_1$, AUC, H-measure [43], and log loss, (2) computational performance, and (3) interpretability measures. In line with past research studies [9], [16], we measure the interpretability as the inverse of the number of model parameters (complexity). In the case of ensemble models such as random forest and gradient boosting, the number of parameters are based on the number and depth of trees, threshold for each of the nodes (2) and the weights of child nodes (2) [16]. A standard RF (XGB) model with 100 (10) trees and max depth of 3 (6) will have 3200 (2560) parameters. A more complex model with a large number of parameters are difficult to interpret by humans. The highly interpretable models, therefore, are expected to use a lower number of parameters without degrading quality of predictions and computational performance.

### B. PERFORMANCE COMPARISON WITH ENSEMBLE MODELS

In the first set of experiments, we compare our models (HUG-BNB, HUG-LR, and HUG-LR-L1) against three baseline methods: LR, RF and XGB. We assess the average rankings obtained by each of these six models on diverse classifier measures (Accuracy, $F_1$, AUC, H-measure, and logistic loss) on forty benchmark datasets. The results of our experiments are shown in Table 11. The best values are marked with a † symbol. The results reveal that the proposed HUG-LR model offers the best overall average rankings. The XGB and RF obtained results that are closer to the HUG-LR model. The detailed performance results of the proposed HUG-LR and ensemble models (RF and XGB) are shown in Tables 12 and 13. It is evident that the proposed HUG model offers identical performance results (and superior results in a few cases) compared to a more complex ensemble models. We also conducted Wilcoxon rank sum tests for equality of performance measures against HUG-LR (HLR) and ensemble models. We did not find sufficient evidence to reject the null hypothesis at a significance level of 0.05. This indicates that the proposed HLR model offer performance comparable to that of the ensemble models.

[5]https://optuna.org/

**TABLE 11. Mean (std deviation) of rankings across all benchmark datasets.**

| Method | Acc. | $F_1$ | AUC | Hm | LL |
|---|---|---|---|---|---|
| LR | 2.65 | 3.60 | 2.40 | 3.10 | 3.35 |
| | (1.51) | (1.59) | (1.20) | (1.34) | (1.64) |
| RF | 1.74 | 2.59 | 1.74 | 1.95 | 2.26 |
| | (0.74) | (1.21) | (0.95) | (1.01) | (0.95) |
| XGB | 1.70 | 2.50 | 1.90 | 2.05 | 2.53 |
| | (1.14) | (1.58) | (1.26) | (1.24) | (1.43) |
| HUG-BNB | 2.80 | 3.33 | 2.68 | 3.58 | 3.80 |
| | (1.23) | (1.35) | (1.06) | (1.20) | (1.27) |
| HUG-LR | 1.55† | 1.90† | 1.48† | 1.78† | 1.78 |
| | (0.74) | (0.83) | (0.67) | (0.94) | (1.04) |
| HUG-LR-L1 | 1.63 | 1.95 | 1.63 | 1.98 | 1.73† |
| | (0.89) | (1.09) | (0.94) | (1.08) | (0.87) |

Acc.: Accuracy; Hm: H-measure; LL - Log loss

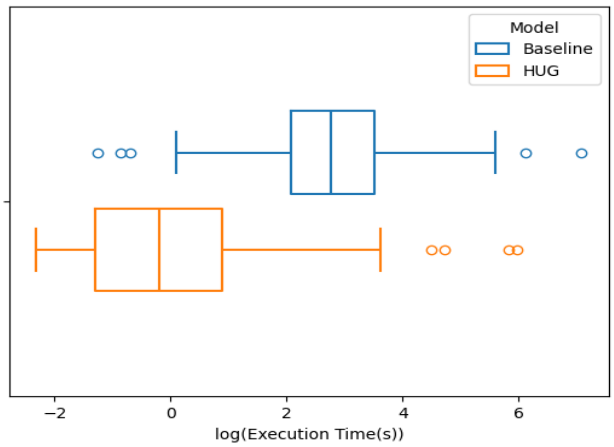


**FIGURE 2. Execution time performance across all benchmark datasets.**

We also assessed the computational performance of our model against the best performing ensemble model. The results of our experiments are shown in Figure 2. The boxplot shows the overall distribution of the execution times (in seconds) across forty benchmark datasets. The results clearly reveal that our HUG-based model do not incur additional execution overhead in comparison to advanced ensemble models across the forty benchmark datasets studied.

### C. PERFORMANCE COMPARISON WITH INTERPRETABLE MACHINE LEARNING MODELS

The performance of our HUG-based model was then compared against the state-of-the-art interpretable classifier

**TABLE 12.** Classifier performance evaluation of HUG-LR, RF, and XGB models (Accuracy, $F_1$ and AUC measures).

| Name (Dno) | Accuracy | | | $F_1$ | | | AUC | | |
|---|---|---|---|---|---|---|---|---|---|
| | HLR | RF | XGB | HLR | RF | XGB | HLR | RF | XGB |
| sonar (B1) | 0.86† | 0.81 | 0.85 | 0.84† | 0.76 | 0.83 | 0.93 | 0.94† | 0.92 |
| breast cancer (B2) | 0.72† | 0.72† | 0.71 | 0.42† | 0.14 | 0.02 | 0.69† | 0.67 | 0.63 |
| congressional voting (B3) | 0.96† | 0.96† | 0.96† | 0.95† | 0.95† | 0.94 | 0.99† | 0.99† | 0.99† |
| breast cancer (wisc.) (B4) | 0.98† | 0.96 | 0.97 | 0.97† | 0.95 | 0.96 | 1.00† | 0.99 | 0.99 |
| ILPD (B5) | 0.73† | 0.72 | 0.72 | 0.43† | 0.34 | 0.29 | 0.75† | 0.74 | 0.74 |
| credit approval (B6) | 0.88† | 0.88† | 0.88† | 0.87† | 0.87† | 0.86 | 0.94† | 0.93 | 0.93 |
| blood transfusion (B7) | 0.79† | 0.76 | 0.77 | 0.32† | 0.06 | 0.11 | 0.75† | 0.69 | 0.71 |
| pima (B8) | 0.77† | 0.76 | 0.75 | 0.63 | 0.64† | 0.60 | 0.83† | 0.82 | 0.81 |
| titanic (B9) | 0.82 | 0.82 | 0.83† | 0.75† | 0.75† | 0.75† | 0.87† | 0.87† | 0.87† |
| tictactoe (B10) | 0.98 | 0.99† | 0.99† | 0.97 | 0.98 | 0.99† | 1.00† | 1.00† | 0.99 |
| german credit (B11) | 0.76 | 0.77† | 0.77† | 0.54† | 0.51 | 0.51 | 0.80† | 0.79 | 0.80† |
| bank note (B12) | 0.99 | 0.99 | 1.00† | 0.98 | 0.99 | 1.00† | 1.00† | 1.00† | 1.00† |
| telecom iranian (B13) | 0.95 | 0.95 | 0.96† | 0.84 | 0.86 | 0.87† | 0.98 | 0.98 | 0.99† |
| abalone (B14) | 0.78 | 0.79† | 0.77 | 0.66† | 0.66† | 0.64 | 0.85 | 0.86† | 0.85 |
| spambase (B15) | 0.95 | 0.95 | 0.96† | 0.93 | 0.94† | 0.94† | 0.98 | 0.99† | 0.99† |
| telecom churn (B16) | 0.91 | 0.93 | 0.94† | 0.66 | 0.72 | 0.76† | 0.89 | 0.91† | 0.90 |
| universal bank (B17) | 0.99† | 0.99† | 0.99† | 0.93 | 0.94† | 0.94† | 0.99 | 1.00† | 1.00† |
| COMPAS (B18) | 0.68† | 0.67 | 0.68† | 0.64† | 0.61 | 0.62 | 0.74† | 0.71 | 0.73 |
| bankruptcy (B19) | 0.97† | 0.97† | 0.97† | 0.33† | 0.25 | 0.21 | 0.93 | 0.93 | 0.94† |
| mushroom (B20) | 1.00† | 1.00† | 1.00† | 1.00† | 1.00† | 1.00† | 1.00† | 1.00† | 1.00† |
| lending club (B21) | 0.84† | 0.84† | 0.84† | 0.07† | 0.00 | 0.00 | 0.67† | 0.67† | 0.65 |
| heloc (B22) | 0.73† | 0.73† | 0.73† | 0.72† | 0.71 | 0.71 | 0.80† | 0.80† | 0.79 |
| vehicle insurance (B23) | 0.94† | 0.94† | 0.94† | 0.11† | 0.05 | 0.06 | 0.83† | 0.83† | 0.82 |
| magic GT (B24) | 0.85 | 0.88 | 0.89† | 0.78 | 0.82 | 0.83† | 0.90 | 0.94 | 0.94† |
| car fraud claim (B25) | 0.84† | 0.84† | 0.84† | 0.06† | 0.02 | 0.01 | 0.71† | 0.70 | 0.69 |
| default credit card (B26) | 0.82† | 0.82† | 0.82† | 0.47† | 0.47† | 0.45 | 0.78† | 0.77 | 0.78† |
| adult income (B27) | 0.84† | 0.82 | 0.84† | 0.64† | 0.59 | 0.64† | 0.89† | 0.87 | 0.89† |
| bank marketing (B28) | 0.89† | 0.89† | 0.89† | 0.29† | 0.00 | 0.00 | 0.79† | 0.75 | 0.72 |
| give me credit (B29) | 0.93 | 0.94† | 0.94† | 0.16 | 0.21 | 0.23† | 0.84 | 0.85 | 0.86† |
| cc worldline ULB (B30)* | 1.00† | - | 1.00† | 0.79 | - | 0.87† | 0.98† | - | 0.98† |
| lung cancer (M31) | 0.47 | 0.56† | 0.32 | 0.40 | 0.46† | 0.20 | 0.62 | 0.75† | 0.54 |
| iris (M32) | 0.96† | 0.94 | 0.92 | 0.96† | 0.94 | 0.92 | 0.99 | 1.00† | 0.99 |
| wine (M33) | 0.99† | 0.98 | 0.96 | 0.99† | 0.98 | 0.96 | 1.00† | 1.00† | 1.00† |
| wheat seeds (M34) | 0.94† | 0.93 | 0.94† | 0.94† | 0.93 | 0.94† | 0.99† | 0.99† | 0.99† |
| yeast (M35) | 0.59 | 0.61† | 0.59 | 0.51 | 0.55† | 0.55† | 0.87† | 0.86 | 0.86 |
| car (M36) | 0.99† | 0.96 | 0.99† | 0.96† | 0.87 | 0.94 | 1.00† | 1.00† | 1.00† |
| wine quality (M37) | 0.66† | 0.65 | 0.62 | 0.40† | 0.34 | 0.29 | 0.79 | 0.78 | 0.80† |
| digits (M38) | 0.98† | 0.98† | 0.98† | 0.98† | 0.98† | 0.98† | 1.00† | 1.00† | 1.00† |
| nursery (M39) | 1.00† | 0.98 | 1.00† | 0.79 | 0.76 | 0.80† | 1.00† | 1.00† | 1.00† |
| letter (M40) | 0.94 | 0.97† | 0.97† | 0.94 | 0.96 | 0.97† | 1.00† | 1.00† | 1.00† |

HLR: HUG-LR model * RF was not run as it took more than an hour

models: SAFE [9] and INAFEN [16]. It is to be noted that INAFEN [16] extends the basic ideas proposed in SAFE [9] and PLTR [15] and is the most recent and state-of-the-art interpretable model.

The SAFE [9] was found to be computationally expensive. Hence, we first analyzed the performance of all three interpretable methods (HLR, SAFE, and INAFEN) on a subset of 15 smaller benchmark datasets. The results of our experiments on prediction quality and computational performance are shown in Figure 3. It is evident that the HLR offers superior performance on both prediction quality and computation against the other two related methods. The prediction quality performance (except log loss) of INAFEN was much closer to that of HLR but the computational performance was observed to be poor. The higher computational overhead in INAFEN is due to the use of occurrence frequencies and the sensitive nature of support and confidence thresholds. The proposed HLR method use the notion of utilities and explores patterns that are useful for downstream supervised learning task. The experimental results clearly reveal the value of the utility based approach.

We also compared the most recent INAFEN model against our method on all thirty binary classification datasets. It is to be noted that INAFEN and SAFE supports only binary classification tasks. The results of our experiments are shown in Figure 4. It is evident that the HLR method offer better performance over INAFEN.

The Wilcoxon rank sum tests for the performance of HLR and other interpretable models are shown in Table 14. The results reveal that the performance improvement of HLR over SAFE was statistically significant at 5% significance level across all the performance measures. While HLR also shows better performance over INAFEN on Accuracy, $F_1$, AUC, and H-measure, the results were not found to be statistically significant. On other performance measures such as logistic loss, HLR's improvement over INAFEN was statistically significant.

**TABLE 13.** Classifier performance evaluation of HUG-LR, RF, and XGB models (H-measure and Log loss).

| Name (Dno) | H-measure | | | Log loss | | |
|---|---|---|---|---|---|---|
| | HLR | RF | XGB | HLR | RF | XGB |
| sonar (B1) | 0.75 | 0.76† | 0.74 | 0.37† | 0.45 | 0.45 |
| breast cancer (B2) | 0.29 | 0.31† | 0.23 | 0.58 | 0.57† | 0.59 |
| congressional voting (B3) | 0.95† | 0.94 | 0.93 | 0.11† | 0.13 | 0.14 |
| breast cancer (wisc.) (B4) | 0.96† | 0.93 | 0.95 | 0.09† | 0.16 | 0.14 |
| ILPD (B5) | 0.34† | 0.32 | 0.31 | 0.53† | 0.57 | 0.64 |
| credit approval (B6) | 0.73† | 0.72 | 0.73† | 0.32† | 0.38 | 0.37 |
| blood transfusion (B7) | 0.30† | 0.24 | 0.26 | 0.48† | 0.51 | 0.51 |
| pima (B8) | 0.44† | 0.43 | 0.39 | 0.48† | 0.50 | 0.51 |
| titanic (B9) | 0.55 | 0.56† | 0.56† | 0.43† | 0.43† | 0.43† |
| tictactoe (B10) | 0.97 | 0.98† | 0.98† | 0.09† | 0.14 | 0.18 |
| german credit (B11) | 0.38† | 0.36 | 0.36 | 0.5† | 0.53 | 0.52 |
| bank note (B12) | 0.98 | 0.99 | 1.00† | 0.10 | 0.01† | 0.01† |
| telecom iranian (B13) | 0.84 | 0.86 | 0.88† | 0.12† | 0.13 | 0.12† |
| abalone (B14) | 0.42 | 0.45† | 0.43 | 0.46 | 0.44† | 0.45 |
| spambase (B15) | 0.86 | 0.89† | 0.89† | 0.16 | 0.14 | 0.13† |
| telecom churn (B16) | 0.60 | 0.67† | 0.67† | 0.24 | 0.20 | 0.19† |
| universal bank (B17) | 0.92 | 0.95† | 0.95† | 0.05 | 0.04 | 0.03† |
| COMPAS (B18) | 0.22† | 0.18 | 0.20 | 0.60† | 0.62 | 0.62 |
| bankruptcy (B19) | 0.64 | 0.67 | 0.68† | 0.09† | 0.09† | 0.09† |
| mushroom (B20) | 1.00† | 1.00† | 1.00† | 0.01 | 0.00† | 0.00† |
| lending club (B21) | 0.13† | 0.11 | 0.10 | 0.41† | 0.42 | 0.42 |
| heloc (B22) | 0.33† | 0.32 | 0.31 | 0.54† | 0.55 | 0.56 |
| vehicle insurance (B23) | 0.34† | 0.34† | 0.29 | 0.19† | 0.19† | 0.20 |
| magic GT (B24) | 0.57 | 0.65 | 0.67† | 0.36 | 0.29 | 0.28† |
| car fraud claim (B25) | 0.16† | 0.14 | 0.13 | 0.40† | 0.40† | 0.40† |
| default credit card (B26) | 0.29† | 0.29† | 0.29† | 0.43† | 0.43† | 0.43† |
| adult income (B27) | 0.48 | 0.44 | 0.49† | 0.35† | 0.38 | 0.35† |
| bank marketing (B28) | 0.33† | 0.29 | 0.19 | 0.28† | 0.31 | 0.35 |
| give me credit (B29) | 0.37 | 0.40 | 0.43† | 0.19 | 0.18† | 0.18† |
| cc worldline ULB (B30)* | 0.88 | - | 0.89† | 0.00† | - | 0.00† |
| lung cancer (M31) | 0.50 | 0.65† | 0.46 | 1.17 | 0.92† | 1.06 |
| iris (M32) | 0.99† | 0.98 | 0.96 | 0.23 | 0.22† | 0.28 |
| wine (M33) | 0.99† | 0.99† | 0.98 | 0.13† | 0.14 | 0.20 |
| wheat seeds (M34) | 0.95† | 0.94 | 0.95† | 0.20† | 0.27 | 0.29 |
| yeast (M35) | 0.60† | 0.59 | 0.58 | 1.10† | 1.13 | 1.23 |
| car (M36) | 0.99† | 0.97 | 0.99† | 0.07† | 0.16 | 0.14 |
| wine quality (M37) | 0.43† | 0.43† | 0.43† | 0.98 | 0.93† | 1.02 |
| digits (M38) | 0.99† | 0.99† | 0.99† | 0.07† | 0.08 | 0.12 |
| nursery (M39) | 1.00† | 0.98 | 0.99 | 0.02 | 0.10 | 0.01† |
| letter (M40) | 0.97 | 0.99† | 0.99† | 0.20 | 0.13† | 0.13† |

HLR: HUG-LR model * RF was not run as it took more than an hour

**TABLE 14.** Wilcoxon rank sum test: HLR vs other interpretable models.

| Performance Measure | HLR vs SAFE | HLR vs INAFEN |
|---|---|---|
| Accuracy | 1.97*** | 0.92 |
| $F_1$ | 1.85*** | 0.72 |
| AUC | 1.76*** | 0.93 |
| H-measure | 1.97*** | 0.88 |
| Log Loss | -1.93*** | -2.54*** |
| Computational performance | -4.67*** | -6.04*** |

$H_0$: The performance levels of both the models are equal
$H_1$: HLR offers improved performance over SAFE/INAFEN

### D. INTERPRETABILITY ASSESSMENTS

We perform model interpretability evaluation against an ensemble model (XGB) and an interpretable model, INAFEN [16]. The normalized interpretability score of the three models are shown in Figure 5. The normalized score is computed by treating the baseline LR model as the best interpretable model (with a perfect score of 1). The results clearly indicate that the proposed HLR model offer higher interpretability over state-of-the-art ensemble and interpretable models. We also conducted Wilcoxon rank sum test to check the statistical significance of improvements offered by the interpretable models (HLR and INAFEN). The hypothesis tested was $H_0$: The interpretability levels of HLR and INAFEN are equal and $H_1$: HLR offers greater interpretability over INAFEN [16]. The observed p-value was 0.0246 signifying that the interpretability improvements are statistically significant at 5%.

## VI. CASE STUDIES

Three case studies in healthcare and finance domains are presented to demonstrate the interpretability aspects of the proposed models. The first case study is based on a dataset shared by National Institute of Diabetes and Digestive and Kidney Diseases [45].[6] The objective is to predict the onset of diabetes based on diagnostic measures. The second case study is based on a direct marketing campaign data collected by a

[6]https://data.world/uci/pima-indians-diabetes

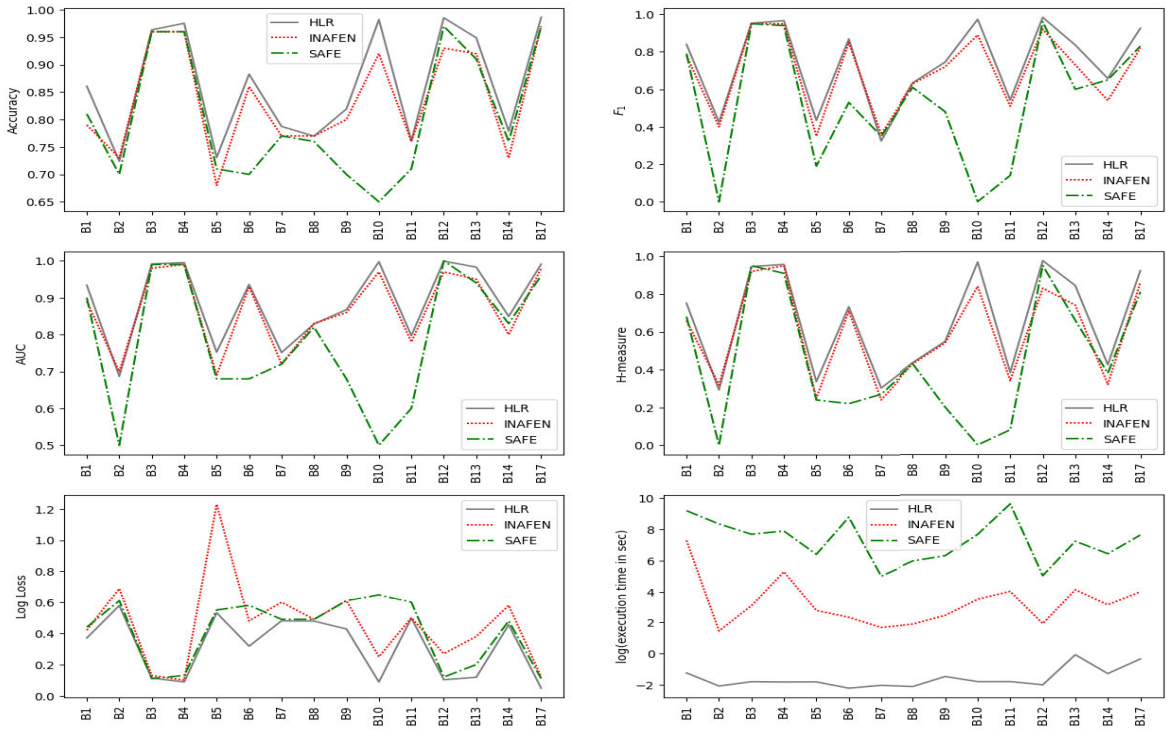


**FIGURE 3.** Comparative analysis of SAFE [9], INAFEN [16], and HUG-LR (HLR) interpretable classifier models on diverse prediction quality and computational performance measures.

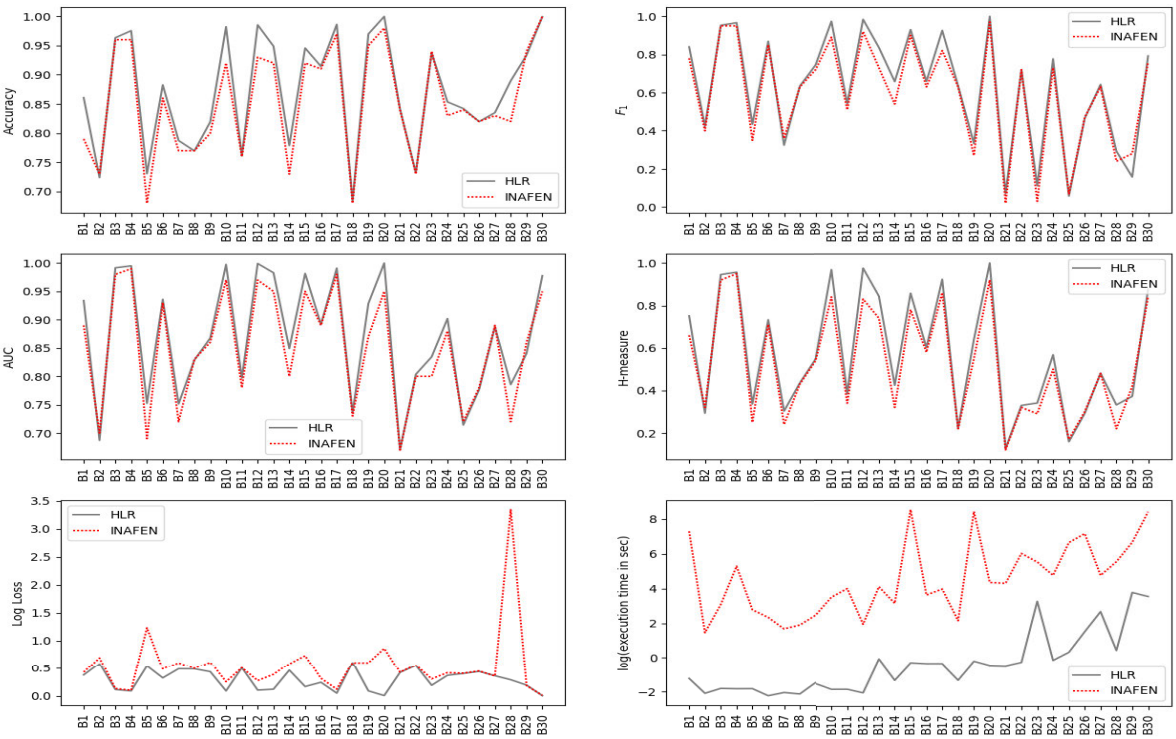


**FIGURE 4.** Comparative analysis of INAFEN [16] and HUG-LR (HLR) interpretable classifier models on diverse prediction quality and computational performance measures.

Portugese banking institution [45].[7] The objective of the bank was to predict the success of their tele-marketing campaign for term deposit subscriptions. The third case study pertains to FICO explainable machine learning challenge[8] where the

[7]https://archive.ics.uci.edu/dataset/222/bank+marketing

[8]https://community.fico.com/s/explainable-machine-learning-challenge

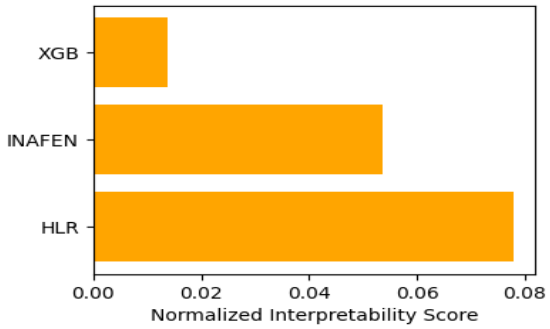


**FIGURE 5.** Normalized interpretability score of XGB, INAFEN [16] and HUG-LR (HLR) models.

objective is to predict whether an applicant will repay their Home Equity Line of Credit (HELOC) account within two years.

### A. CASE 1: PREDICT ONSET OF DIABETES MELLITUS

This case involves the use of female patient data who are at least 21 years of age and are of Pima Indian heritage. The data consists of 8 diagnostic measure variables and 1 outcome variable. The diagnostic measures captured were: number of pregnancies, glucose levels, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. The outcome variable is whether the patient had diabetes or not. The data has 768 patient observations and 268 of them are known to have diabetes mellitus condition. The objective is to train a model based on this data and determine the factors that influence the onset of diabetes mellitus disease.

We applied the proposed HUG models on this dataset (B8) with $\mathscr{B} = 7$, $\mathscr{L} = 1$, and $\mathscr{G} = 5$e-3. Our model generated on an average 24 patterns across all the 10 cross-validation folds. The performance results of our model was comparable to that of advanced ensemble models (refer to Table 12).

Figure 6 gives a scatter plot (1D HUG profiles) of top-30 instances with highest prediction probabilities. The plot indicates whether a particular instance has a specific pattern or not. One can observe clear differences in observed patterns for patients with and without diabetes.

The class level patterns (with predicted probabilities above 75%) are shown in Figure 7. One can clearly observe distinct differences in the age, bmi, bp, and other diagnostic measure values of patients. While there are some observed overlap in patterns, the higher values of age, bmi, and pregnancies are observed in the case of patients with diabetes. The profiles can be generated at different predicted probability values to study the distinct and overlapping patterns across classes to design suitable patient intervention strategies.

An illustrative individual patient level prediction and the observed explanatory pattern is shown in Figure 8. The differences in patterns discovered by the model is in line with the expected diagnostic profiles of diabetic and non-diabetic patients. The generated profiles can also be easily extended to generate counterfactual explanations to help a patient achieve desired outcomes.

### B. CASE 2: TERM DEPOSIT SUBSCRIPTION IN A PORTUGESE BANK

A Portugese banking institution has conducted a direct marketing campaign for term deposit subscriptions. Based on the campaign, the bank had collected information on 15 different variables. These variables can be broadly categorized as: demographics, credit performance, past and current campaign performance, and socio-economic variables. The data (B28) has 41,188 client observations and 4,640 of them belonged to those who have successfully subscribed to term deposits. The objective of the bank was to use an analytical model to predict factors that influence term deposit subscriptions.

Our model was applied with $\mathscr{B} = 11, \mathscr{L} = 2$, and $\mathscr{G} = 3$e-3. Figure 9 gives the scatter plot of top-30 instances with highest predicted probabilities. It is evident that a fewer set of patterns are required to explain non-subscription outcomes. The higher 3-months euribor rate is associated with non-subscription of term deposits. Similarly, a lower 3-months euribor rate contributes to higher term deposit subscription outcomes. This finding corroborates earlier findings in the literature [48]. Other patterns discovered by the model also aligns well with the past observations on feature importance analysis on the bank marketing data.

Figure 10 provides two dimensional HUG profiles. The figure primarily depicts the discovered patterns of length two using a relationship plot. The 'x' (closed 'o') symbol indicates instances belonging to successful (unsuccessful) subscription outcomes. All the patterns that pertain to a specific instance are marked in the same color. A jitter is applied to display all the instances that share a common pattern.

The class profiles and individual instance predictions can also be generated as illustrated in the earlier diabetes prediction case study. These plots can help a decision maker easily explain prediction outcomes. It also allows a decision maker to identify necessary intervention strategies to achieve improved term deposit subscription outcomes.

### C. CASE 3: HOME EQUITY LENDING DECISIONS

The introduction of data protection acts such as General Data Protection Regulation (GDPR) and Equal Credit Opportunity Act (ECOA) and their emphasis on 'right to explanation' has made it imperative for financial institutions to invest in explainable and more accurate credit scoring models. The FICO explainable machine learning challenge is primarily an attempt to leverage community skills and explore innovative explainable models in the credit scoring domain. This case uses the Home Equity Line of Credit (HELOC) dataset that was shared as part of the FICO challenge to demonstrate the utility of the proposed models.

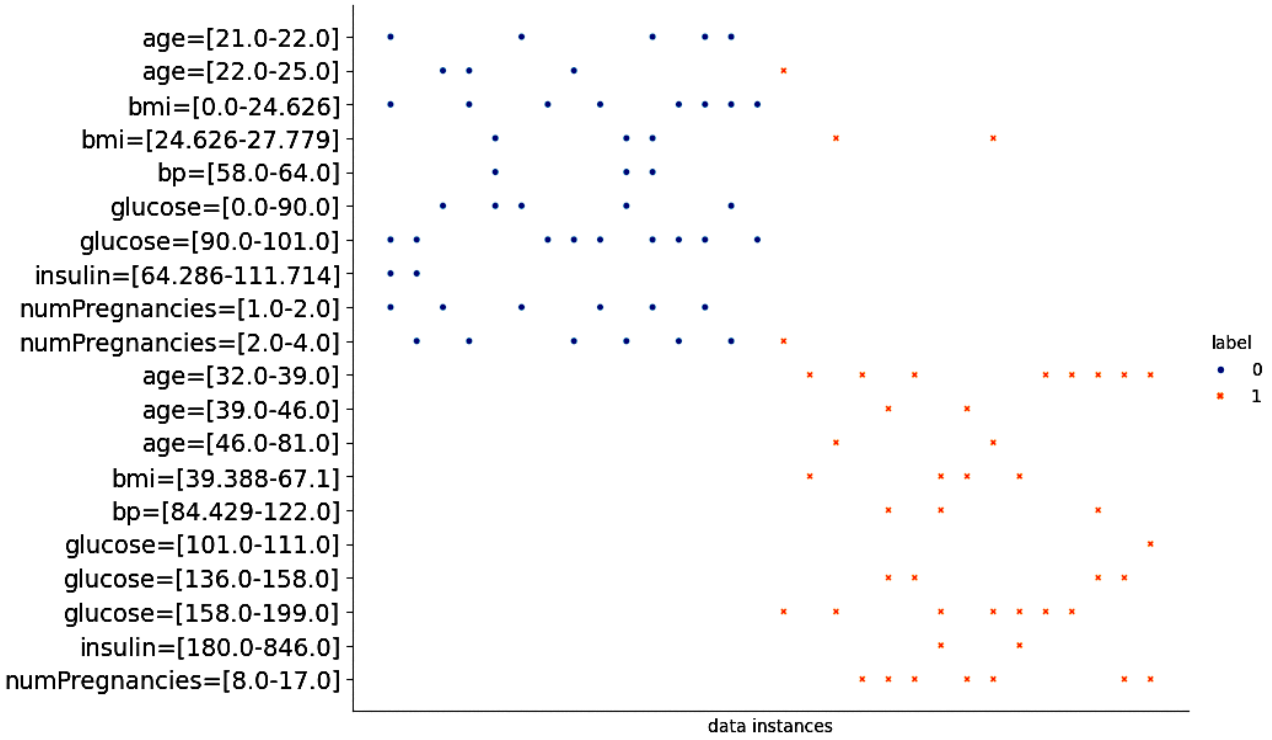


**FIGURE 6.** Pima DM Case: Overall pattern analysis (30 instances).

**CLASS 0: NO DIABETES**

| | |
|---|---|
| age: | 21.0-22.0, 22.0-25.0, 39.0-46.0 |
| bmi: | 0.0-24.626, 24.626-27.779, 39.388-67.1 |
| bp: | 58.0-64.0, 84.429-122.0 |
| glucose: | 0.0-90.0, 90.0-101.0, 101.0-111.0, 136.0-158.0 |
| insulin: | 64.286-111.714, 180.0-846.0 |
| numPregnancies: | 1.0-2.0, 2.0-4.0 |

**CLASS 1: DIABETES**

| | |
|---|---|
| age: | 32.0-39.0, 39.0-46.0, 46.0-81.0 |
| bmi: | 24.626-27.779, 39.388-67.1 |
| bp: | 84.429-122.0 |
| glucose: | 136.0-158.0, 158.0-199.0 |
| insulin: | 180.0-846.0 |
| numPregnancies: | 8.0-17.0 |

**FIGURE 7.** Pima DM Case: Class profiles.

**EXAMPLE INSTANCES**

| CLASS 0 | CLASS 1 |
|---|---|
| Predicted probability: 0.82 | Predicted probability: 0.83 |
| Explanatory patterns | Explanatory patterns |
| age: 22.0-25.0 | glucose: 158.0-199.0 |
| numPregnancies: 1.0-2.0 | bmi=39.388-67.1 |
| glucose: 101.0-111.0 | insulin: 180.0-846.0 |
| bp: 58.0-64.0 | bp=84.429-122.0 |

**FIGURE 8.** Pima DM Case: Individual instance explanations.

The data available for this study has information about 10,459 (5,000) real home owners (good) credit applications described on 23 different variables. The variables can be broadly categorized as estimated risk, length of applicant's credit history, delinquency, inquiries, and balances.

We filtered records with no bureau records (identified with a special flag of -9). We replaced other missing value cases (marked with a special flag of -7 and -8) with median column values. The final dataset used in our experiments had 9,861 client observations with 4,733 of them labeled as the ones with good credit. Our HUG model was then applied on the processed dataset with $\mathscr{B} = 8, \mathscr{L} = 1$, and $\mathscr{G} =$ 1e-3. Our model discovered 75 patterns (on an average across all 10 cross-validation folds). The performance of the proposed model against the ensemble models (RF and XGB) can be referred to in Table 12.

Figure 11 gives the class profiles of good and bad credit clients whose predicted probabilities are more than 94%. It is evident that good credit customers had higher estimated external risk, higher average number of months in file,

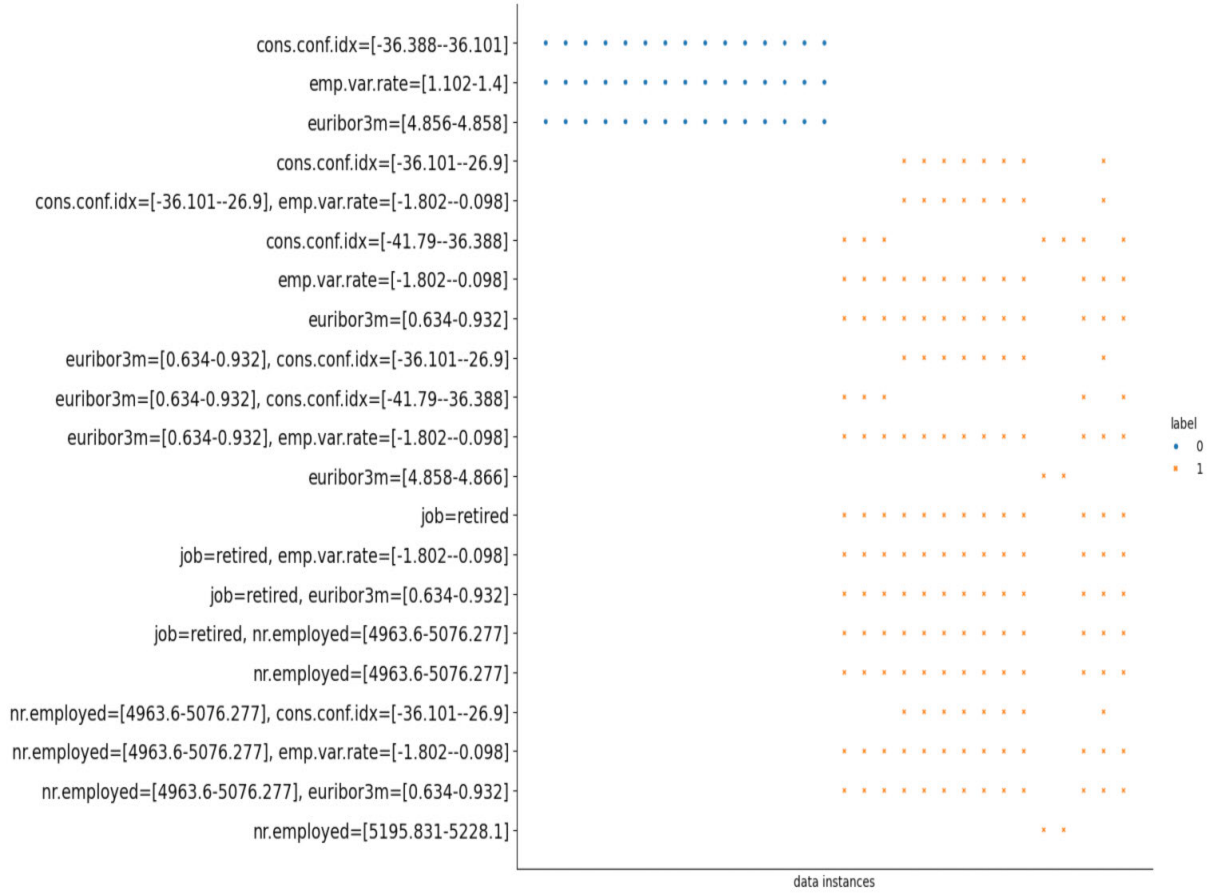


**FIGURE 9.** Bank Marketing Case: Overall pattern analysis.

incurred lower revolving burdens, and completed higher number of satisfactory trades. These findings are in line with prior interpretability analysis that explored complex models [10].

### D. DISCUSSIONS AND IMPLICATIONS

The foregoing discussions clearly reveal the merits of the proposed HUG based models for interpretable machine learning. The model relies on three key parameters ($\mathscr{B}$, $\mathscr{L}$, $\mathscr{G}$). The value of the pattern length ($\mathscr{L}$) was found to be quite small ($\leq 2$ for 90% of the datasets studied). It is to be noted that exhaustive enumeration of patterns of size 2 and 3 can be also very expensive. This paper explored a specific class of patterns (*HUG*) to limit the search space and identify key patterns that aid the downstream supervised learning task. Rigorous experimental evaluation and illustrative cases demonstrate the value of the proposed approach in terms of prediction quality performance, computational performance, and interpretability.

#### 1) THEORETICAL CONTRIBUTIONS

The paper makes important contributions to the field of utility pattern mining as well as machine learning. High utility pattern mining is one of the active areas of research. Numerous algorithms have been proposed in the literature to mine several variants of utility patterns. However, there are limited research studies on demonstrations of the practical use of utility mining for specific business contexts. This paper introduced a new class of utility patterns (named *HUG*), presented a seven stage HUG-IML process workflow, and demonstrated how it could effectively be used in supervised learning problem contexts. Our experiments primarily focused on binary and multi-class classification tasks. But, the core ideas can be easily extended to regression and multi-label learning problems.

The HUG-IML model proposed in this paper allows one to systematically explore simpler models in the Rashomon set [11]. The simpler models can be explored by tuning the three key parameters in the HUG model. For instance, one could reduce the size of $\mathscr{L}$ to explore if an alternate model descriptions can be generated without significant reduction in overall model performance. A more systematic investigation of the search for simpler models will require development of exact or meta-heuristic search algorithms.

One can also view the proposed idea as a flipped neural model. In a standard neural model, the neural architecture is initialized (or fixed) and weights are learnt through optimization. On the other hand, the proposed model uses a

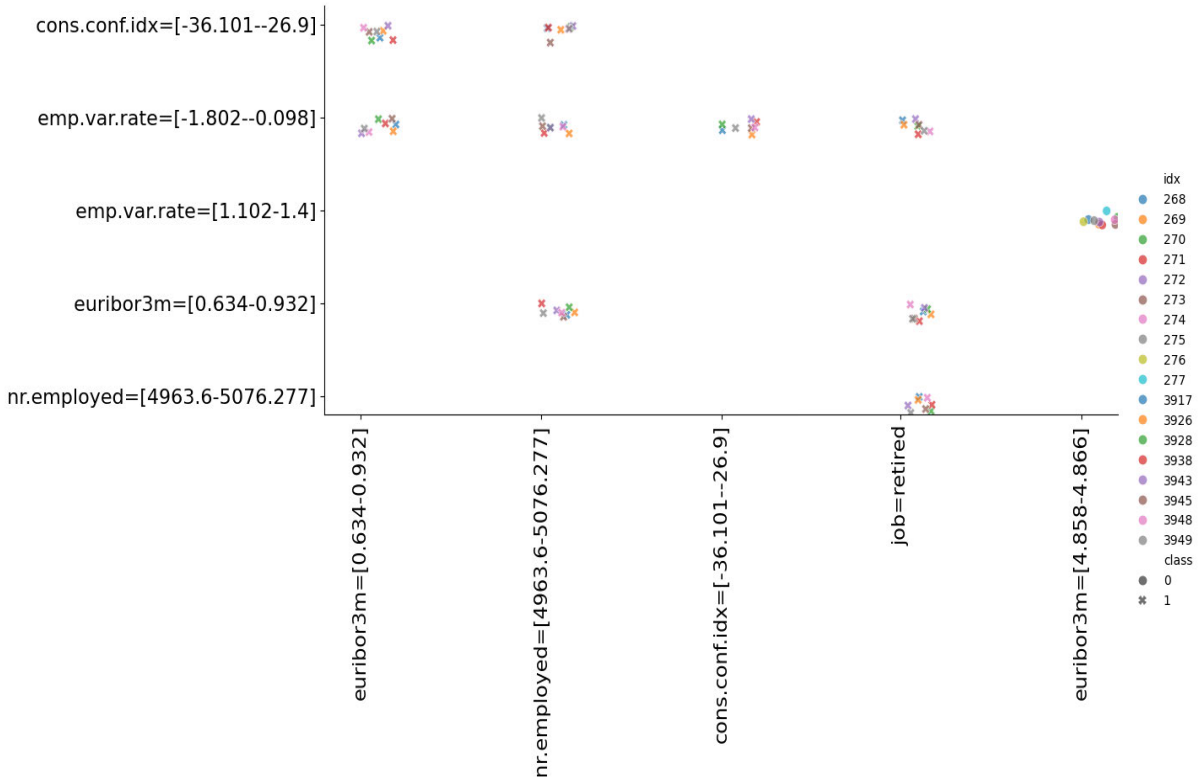


**FIGURE 10.** Bank Marketing Case: 2D HUG Profile.

**CLASS 0: BAD CREDIT**

AverageMInFile: 4.0-43.114, 43.114-57.106, 85.09-96.856
ExternalRiskEstimate: 33.0-60.023, 60.023-63.988, 63.988-68.014, 68.014-71.979
MSinceMostRecentInqexcl7days: 0.0-1.001
NetFractionRevolvingBurden: 41.064-55.912, 55.912-74.008, 74.008-232.0
NumInqLast6Mexcl7days: 0.0-1.003, 3.003-66.0
NumSatisfactoryTrades: 0.0-9.006, 9.006-13.035
NumTotalTrades: 0.0-8.996

**CLASS 1: GOOD CREDIT**

AverageMInFile: 96.856-114.982, 114.982-322.0
ExternalRiskEstimate: 84.972-94.0
MSinceMostRecentInqexcl7days: 1.001-4.992, 4.992-24.0
NetFractionRevolvingBurden: 0.0-2.993, 2.993-9.002
NumInqLast6Mexcl7days: 0.0-1.003
NumSatisfactoryTrades: 27.966-33.97, 33.97-79.0
NumTotalTrades: 37.024-104.0

**FIGURE 11.** Home equity lending case: Class profiles.

additive weighting scheme (i.e. utilities acts as fixed weights) and learns the best architecture (higher-order interactions) through pattern mining. The latter approach explored in this paper makes the model easily interpretable as demonstrated through this research study.

The existing utility mining algorithms in the literature are evaluated using synthetically generated utility values on both real and artificial datasets. The external (internal) utility values are commonly generated using log-normal (uniform) distributions [29]. Our novel utility construction mechanism based on supervised labels allows generation of large-scale benchmark datasets, assess the efficacy of utility mining algorithms, and advance the field further.

### 2) MANAGERIAL IMPLICATIONS

There is a growing need in organizations to build machine learning models that can generate explanations. The practitioners who use complex models for high performance often use post-hoc models for explanation. However, prior research has raised concerns on use of such models and called for development of new models that are intrinsically interpretable [6]. This paper presented an approach to address this need and achieve high interpretability without compromising overall model performance. It also presented case studies on diabetes prediction, term deposit subscriptions, and lending decisions. The practitioners can use the HUG models and the interpretability profiles to generate rich explanations. They also have the flexibility to explore alternate model explanations and manage their performance and interpretability needs by configuring the HUG model parameters.

### 3) LIMITATIONS OF THE STUDY

Our HUG-based classifier model uses three parameters namely, the number of bins $\mathscr{B}$, maximum pattern length $\mathscr{L}$, and information gain threshold value $\mathscr{G}$. While our model uses certain heuristics for automatic selection of default parameter values, it may not always produce the best results. The parameter tuning using grid search is an expensive process and is one of the limitations of the current study. The data transformation process used in this work converts the tabular data to an unordered transaction list format before HUG pattern mining. This obviates the need for missing value imputations. But, our current work do not consider these factors. Our work also doesn't consider rapidly changing environments and streaming data that require continuous model monitoring and management. Another limitation of the work is the nature of tasks (binary and multi-class classifications) supported by our model.

## VII. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

We presented an intrinsically interpretable machine learning model for classification problems. The model explored novel *HUG* patterns to identify specific class of higher order patterns that aid supervised learning tasks. A seven stage HUG-IML process workflow was also presented. The utility of the proposed model was demonstrated through rigorous experimental evaluation on forty benchmark binary and multi-class classification datasets. Three case studies in healthcare and finance domains were described to illustrate the interpretability aspects of the proposed HUG models.

As part of the future work, we plan to extend the HUG model to handle other supervised learning tasks e.g. regression, ranking, and recommender system problems. We also plan to investigate the use HUG-based models in environments where there are very high missing and noisy values without any imputation mechanisms or treatments. The proposed approach adapted the THUI algorithm [25] that relies on simple utility list structure for pattern mining. More recent algorithms in the literature use advanced utility list structures (e.g. compact lists) to significantly speed up the mining process. Future work could explore use of such advanced methods to achieve further scale in performance for the supervised learning problems. Another interesting avenue of research is to examine the applications of sequential high utility patterns [49] or its variants for problems that involve use of non-tabular data sources such as images and text.

## REFERENCES

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias," in *Ethics of Data and Analytics*. New York, NY, USA: Auerbach, 2022, pp. 254–264.

[2] R. Wexler. (2017). *When a Computer Program Keeps You in Jail, the New York Times*. [Online]. Available: https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.

[4] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Med.*, vol. 15, no. 11, Nov. 2018, Art. no. e1002683.

[5] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, Dec. 2019.

[6] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statist. Surveys*, vol. 16, no. none, pp. 1–85, Jan. 2022.

[7] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki, "The dangers of post-hoc interpretability: Unjustified counterfactual explanations," 2019, *arXiv:1907.09294*.

[8] H. Lakkaraju and O. Bastani, "'How do I fool you?' Manipulating user trust via misleading black box explanations," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2020, pp. 79–85.

[9] A. Gosiewska, A. Kozak, and P. Biecek, "Simpler is better: Lifting interpretability-performance trade-off via automated feature engineering," *Decis. Support Syst.*, vol. 150, Nov. 2021, Art. no. 113556.

[10] C. Chen, K. Lin, C. Rudin, Y. Shaposhnik, S. Wang, and T. Wang, "A holistic approach to interpretability in financial lending: Models, visualizations, and summary-explanations," *Decis. Support Syst.*, vol. 152, Jan. 2022, Art. no. 113647.

[11] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Stat. Sci.*, vol. 16, no. 3, pp. 199–231, Aug. 2001.

[12] L. Semenova, C. Rudin, and R. Parr, "On the existence of simpler machine learning models," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Jun. 2022, pp. 1827–1858.

[13] U. Khurana, D. Turaga, H. Samulowitz, and S. Parthasrathy, "Cognito: Automated feature engineering for supervised learning," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 1304–1307.

[14] U. Khurana, H. Samulowitz, and D. Turaga, "Feature engineering for predictive modeling using reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–20.

[15] E. Dumitrescu, S. Hué, C. Hurlin, and S. Tokpavi, "Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects," *Eur. J. Oper. Res.*, vol. 297, no. 3, pp. 1178–1192, Mar. 2022.

[16] M. Liu, C. Guo, and L. Xu, "An interpretable automated feature engineering framework for improving logistic regression," *Appl. Soft Comput.*, vol. 153, Mar. 2024, Art. no. 111269.

[17] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, Jun. 2000.

[18] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 1993, pp. 207–216.

[19] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. 4th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1998, pp. 80–86.

[20] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2001, pp. 369–376.

[21] X. Yin and J. Han, "CPAR: Classification based on predictive association rules," in *Proc. SIAM Int. Conf. Data Mining*, May 2003, pp. 331–335.

[22] F. Berzal, J.-C. Cubero, D. Sánchez, and J. M. Serrano, "ART: A hybrid classification model," *Mach. Learn.*, vol. 54, no. 1, pp. 67–92, Jan. 2004.

[23] J. Li and O. R. Zaiane, "Exploiting statistically significant dependent rules for associative classification," *Intell. Data Anal.*, vol. 21, no. 5, pp. 1155–1172, Oct. 2017.

[24] C.-W. Wu, P. Fournier-Viger, J.-Y. Gu, and V. S. Tseng, "Mining closed+ high utility itemsets without candidate generation," in *Proc. Conf. Technol. Appl. Artif. Intell. (TAAI)*, Nov. 2015, pp. 187–194.

[25] S. Krishnamoorthy, "Mining top-k high utility itemsets with effective threshold raising strategies," *Expert Syst. Appl.*, vol. 117, pp. 148–165, Mar. 2019.

[26] N. T. Tung, T. D. D. Nguyen, L. T. T. Nguyen, and B. Vo, "An efficient method for mining high-utility itemsets from unstable negative profit databases," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121489.

[27] F. Wang and C. Rudin, "Falling rule lists," in *Artificial Intelligence and Statistics*. New York, NY, USA: PMLR, 2015, pp. 1013–1022.

[28] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Amer. Stat. Assoc.*, vol. 107, no. 500, pp. 1590–1598, Dec. 2012.

[29] P. Fournier-Viger, C. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning," in *Proc. Int. Symp. Methodologies Intell. Syst.*, 2014, pp. 83–92.
[30] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "EFIM: A fast and memory efficient algorithm for high-utility itemset mining," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 595–625, May 2017.
[31] S. Krishnamoorthy, "HMiner: Efficiently mining high utility itemsets," *Expert Syst. Appl.*, vol. 90, pp. 168–183, Dec. 2017.
[32] G.-C. Lan, T.-P. Hong, and V. S. Tseng, "Discovery of high utility itemsets from on-shelf time periods of products," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5851–5857, May 2011.
[33] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Sci. Pattern Recognit.*, vol. 1, no. 1, pp. 54–77, 2017.
[34] J. C. Lin, S. Ren, P. Fournier-Viger, and T.-P. Hong, "EHAUPM: Efficient high average-utility pattern mining with tighter upper bounds," *IEEE Access*, vol. 5, pp. 12927–12940, 2017.
[35] V. V. Vu, M. T. H. Lam, T. T. M. Duong, L. T. Manh, T. T. T. Nguyen, L. V. Nguyen, U. Yun, V. Snasel, and B. Vo, "FTKHUIM: A fast and efficient method for mining top-K high-utility itemsets," *IEEE Access*, vol. 11, pp. 104789–104805, 2023.
[36] M. Han, N. Zhang, L. Wang, X. Li, and H. Cheng, "Mining closed high utility patterns with negative utility in dynamic databases," *Int. J. Speech Technol.*, vol. 53, no. 10, pp. 11750–11767, May 2023.
[37] J. C.-W. Lin, J. Zhang, and P. Fournier-Viger, "High-utility sequential pattern mining with multiple minimum utility thresholds," in *Proc. 1st Int. Joint Conf.*, 2017, pp. 215–229.
[38] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.
[39] G. I. Allen, L. Gan, and L. Zheng, "Interpretable machine learning for discovery: Statistical challenges and opportunities," *Annu. Rev. Statist. Appl.*, vol. 11, no. 1, pp. 97–121, Apr. 2024.
[40] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.
[41] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–42, Sep. 2019.
[42] S. García, J. Luengo, J. A. Sáez, V. López, and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 734–750, Apr. 2013.
[43] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve," *Mach. Learn.*, vol. 77, no. 1, pp. 103–123, Oct. 2009.
[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[45] M Litchman. (2013). *Machine Learning Repository. 2013*. [Online]. Available: http://archivee.ics.uci.edu/ml
[46] M. Kull, T. S. Filho, and P. Flach, "Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *Artificial Intelligence and Statistics*. New York, NY, USA: PMLR, 2017, pp. 623–631.
[47] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, Sep. 2023.
[48] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decis. Support Syst.*, vol. 62, pp. 22–31, Jun. 2014.
[49] J. Yin, Z. Zheng, and L. Cao, "USpan: An efficient algorithm for mining high utility sequential patterns," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 660–668.

**SRIKUMAR KRISHNAMOORTHY** (Member, IEEE) received the Ph.D. degree in IT and systems management from Indian Institute of Management Lucknow, India. He is currently a faculty with the Information Systems Area, Indian Institute of Management Ahmedabad, India. His key research interests include data mining, machine learning, social media analytics, and personalization in e-commerce.