This is the published version of a paper published in *Computer graphics forum (Print)*.

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

HardVis: Visual Analytics to Handle Instance Hardness Using Undersampling and Oversampling Techniques

A. Chatzimparmpas,[1] [iD] F. V. Paulovich[2] [iD] and A. Kerren[1,3] [iD]

[1]Department of Computer Science and Media Technology, Linnaeus University, Växjö, Sweden
{angelos.chatzimparmpas, andreas.kerren}@lnu.se
[2]Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
f.paulovich@tue.nl
[3]Department of Science and Technology, Linköping University, Norrköping, Sweden
andreas.kerren@liu.se

Abstract

Despite the tremendous advances in machine learning (ML), training with imbalanced data still poses challenges in many real-world applications. Among a series of diverse techniques to solve this problem, sampling algorithms are regarded as an efficient solution. However, the problem is more fundamental, with many works emphasizing the importance of instance hardness. This issue refers to the significance of managing unsafe or potentially noisy instances that are more likely to be misclassified and serve as the root cause of poor classification performance. This paper introduces HardVis, a visual analytics system designed to handle instance hardness mainly in imbalanced classification scenarios. Our proposed system assists users in visually comparing different distributions of data types, selecting types of instances based on local characteristics that will later be affected by the active sampling method, and validating which suggestions from undersampling or oversampling techniques are beneficial for the ML model. Additionally, rather than uniformly undersampling/oversampling a specific class, we allow users to find and sample easy and difficult to classify training instances from all classes. Users can explore subsets of data from different perspectives to decide all those parameters, while HardVis keeps track of their steps and evaluates the model's predictive performance in a test set separately. The end result is a well-balanced data set that boosts the predictive power of the ML model. The efficacy and effectiveness of HardVis are demonstrated with a hypothetical usage scenario and a use case. Finally, we also look at how useful our system is based on feedback we received from ML experts.

Keywords: instance hardness, imbalanced data, sampling techniques, machine learning, visual analytics, visualization

CCS Concepts: • Human-centred computing → Visualization; Visual analytics; • Machine learning → Supervised learning

1. Introduction

In machine learning (ML), *easy to classify instances* are those for which ML models have a high probability of predicting the correct class label, whereas the opposite is true for the *difficult to classify instances* [YLW*21]. The assessment of **instance hardness** can reveal useful information about the boundaries of ML capabilities [PHOMU15]. Instance hardness is a common problem that inspired the creation of well-known boosting algorithms [YLF*21], such as AdaBoost [FSA99]. It can also highlight when and where human intervention is required to resolve data-related issues. The ultimate goal of such a procedure is to identify misclassified instances and interpret why this has happened [CdMP14], as well

as improve predictive performance [SMGC14]. This scenario is where visual analytics (VA) approaches are considered as a possible solid solution [WDC*22] with many recent works focusing on problematic subsets of data for the interpretation and performance boost of ML models [CVW22, ZOS*23]. However, the classification problem becomes significantly more complex when the data set contains both *class overlap* and *class imbalance*. There are many problems [RKN06, WLC*13, HKB18, CCS06, KHM98] in which the minority class—composed of mostly unsafe instances such as borderline examples, rare cases and outliers—is of great interest [NS16]. A medical diagnosis task of detecting ill patients within a healthy majority is an example that illustrates the great importance of imbalanced data problems. Learning from such

unbalanced data sets can be difficult because most models will theoretically attain high accuracy by merely predicting the majority class [Ste16].

There are two fundamental methodologies to deal with these kinds of imbalance problems: *data-level* and *algorithm-level approaches* [Kra16]. The first method utilizes pre-processing strategies in order to balance the training set. The second method aims at determining what causes a certain ML model to fail in imbalanced circumstances and addressing those flaws to create new robust ML models [CZV13, CT17]. *Ensemble approaches* have also grown in popularity, as they allow for a fusion of model combinations and the usage of one of the methodologies discussed above [KGW17, WGC14]. In this paper, we solely focus on the data-level approaches because they are not tied to a specific ML algorithm; and they remain as an underrepresented category without the support of VA solutions [CMJK20, CMJ*20, YCY*21]. These approaches perform data sampling with either undersampling or oversampling techniques. The former removes instances from the training set, while the latter generates synthetic/artificial instances from the already existing data to balance the class distribution.

For *undersampling*, two advanced techniques for concurrently eliminating and maintaining instances are: one-sided selection (OSS) [KM97] and neighbourhood cleaning rule (NCR) [Lau01]. The goal here is to remove ambiguous points on the class boundary and, at the same time, keep any non-redundant examples far from the decision boundary. On the other hand, a frequently used *oversampling* algorithm is the synthetic minority oversampling technique (SMOTE) [CBHK02], but it comes with several drawbacks. One of those is the uniform approach to oversampling which considers all minority instances equally important. To deal with this flaw, adaptive synthetic (ADASYN) [HBGL08] was invented that dynamically determines which cases may represent a greater challenge for an ML model, thus oversampling instances around class borders. A non-trivial issue with these algorithms is that they require the exploration of specific parameters from the user side. The common ground in all these techniques is the k-value that should be set for the k-nearest neighbours (KNN) algorithm [Alt92, FH89]. Depending on this critical value, more or fewer instances will be removed or used for artificial addition, which might cause harm to the predictive performance of the ML model under training. For example, in an imbalanced healthcare data scenario, a data analyst who blindly trusts one of the previous heuristic-based approaches for undersampling and chooses a high k-value will eventually remove many healthy patients (belonging to the majority class), leading to a balanced training set but with a significant loss of critical data for generalizing when the system is put into production. Tuning those parameters is not straightforward to automate since there are multiple ways on how to combine undersampling and oversampling; thus, this makes room for human-centric solutions such as interactive visualizations that facilitate human exploration and domain knowledge injection into this complex problem. Furthermore, the local characteristics of each instance are at least equally important as the global extracted patterns, which are usually investigated with automated methods [RVV*15]. Consequently, a remaining open question is: **(RQ1)** for a given data set, how can visualization assist users in deciding the optimal parameters for the undersampling and oversampling techniques?

Another challenge related to the previous one is to identify common *local characteristics* of the instances in order to classify them into data types, as in the work of Napierala and Stefanowski [NS16] who acknowledge four types of data: *safe, borderline, rare* and *outliers* (SBRO in short). As described before, depending on the selected k-value, the distribution of instances in those types is subject to change [SK17]. Outliers can account for a sizable fraction of a class, especially in minority groups; as a result, in some data sets, they may even pre-dominate [NS16]. It is dangerous to treat outliers as noise and utilize noise-handling approaches such as re-labelling or eliminating them from the learning set without extensively analysing them [XYX*19, BNR20]. Separating noise from outliers is a necessary but non-trivial task [SAPV16]. If we consider the previously established example, a data analyst will receive various distributions of SBRO instances (*i.e.* separations of patients) depending on the k-value selected for splitting the data with KNN into these four data types, where some combinations will lead to more outliers that could be potentially treated as noisy data compared to others. Moreover, rare cases exist in several data sets [Rav11]. This indicates that class difference is not the only source of difficulties when dealing with unbalanced data, but local characteristics of each class are also essential [NS16]. This problem is partially addressed with upgraded versions of SMOTE and hybrid algorithms. For example, Borderline-SMOTE [HWM05] focuses on oversampling cases that are near to class boundaries. Safe-Level-SMOTE [BSL09] allocates weights to instances based on how 'safe' they are from the majority class influence, and it uses these weights to guide the introduction of artificial examples. Additionally, selective pre-processing of imbalanced data (SPIDER) [NSW10] focuses on highlighting problematic cases, particularly those that overlap with the majority class. Nevertheless, it would be better to dynamically adjust this ratio based on the exploration of local data features and the varying density of examples. In such dynamic approaches, evaluating several types of data could be useful [NS16]. Thus, a question that arises is: **(RQ2)** which algorithmic suggestions should users accept based on the visual analysis of particular SBRO areas or even whole regions?

In this paper, we present a VA system, called HARDVIS, that incorporates undersampling and oversampling techniques for the management of both instance hardness and class imbalance independent of the ML algorithm in use. It adopts validation metrics suitable for imbalanced multi-class classification problems and includes several iterative phases that enable users to apply undersampling and oversampling in various strategic schemes. Our contributions are summarized as follows:

- a coherent visual analytic workflow that takes into account instance hardness, while leveraging undersampling and oversampling techniques;
- a working prototype of the suggested workflow in the form of our VA system, HARDVIS, which comprises a novel combination of multiple coordinated views to support the entire process of selectively undersampling and oversampling parts of the data set;
- a proof-of-concept showcasing the proposed system's applicability with a hypothetical usage scenario, and a use case that illustrates the utility of our decision to deploy sampling approaches and involves humans in-between automated methods; and

- the discussion of the methodology and findings of interview sessions with five ML experts, presenting positive results.

The remainder of this paper is organized as follows. In Section 2, we review automated methods for the detection of different data types, visually assisted identification of outliers and rare examples, and visualization approaches for data-centric ML error analysis. Afterwards, in Section 3, we outline the analytical tasks and design goals for using VA to manage instance hardness in imbalanced data sets, and we emphasize the need for both automatic approaches and human intuition. Section 4 presents the system's functionalities and simultaneously describes a first simple use case with multiple cycles of undersampling and oversampling applied to specific instances in order to enhance predictive performance. Following that, in Section 5, we illustrate the applicability and utility of HARDVIS with two real-world data sets concentrating on detecting breast cancer and recognizing vehicles from their silhouettes. Thereafter in Section 6, we examine the input received from the expert interviews, including limitations identified by the experts. Subsequently, in Section 7, we reflect further on the visual design and the limitations of our work that lead to future plans for HARDVIS. Finally, Section 8 concludes our paper.

2. Related Work

This section summarizes previous research on automatic approaches for the identification of different types of instances, visualization methods for outlier/anomaly and rare category detection, and data-centric ML solutions from the visualization community. To underscore the uniqueness of our approach, we explain the difference between such solutions contrasted to HARDVIS. To the best of our knowledge, there is no literature explaining the use of VA for the complete undersampling and oversampling procedure, along with the partial application in specific types based on the visual exploration of data and distributions.

2.1. Automatically distinguishing types of instances

In the ML community, several methods for automatically categorizing data instances into different types exist, with a particular focus on the outlier/anomaly detection research in the past decades [CBK09, HA04]. Nevertheless, most algorithms cannot identify rare cases that are typically isolated groups, including a set of comparable data examples that deviate from the majority—rather than single isolated instances which are outliers. The majority of anomaly detection techniques can be divided into five categories: (1) classification-based [HHWB02, WMCW03, MC03], (2) density-based [BS03, BKNS00], (3) clustering-based [MLC07, VW09, SPBW12], (4) statistical-based [YTWM04, KK17] and (5) ensemble approaches [VK09, SLSH15, VC17, ZDH*17]. The last category is a hybrid one, which aims to combine the benefits of the various techniques from the other categories. The problem with all the approaches, except for the density-based approaches, is the misalignment with sophisticated undersampling (*e.g.* NCR) and oversampling algorithms (*e.g.* ADASYN) that are using KNN to propose instances for removal or addition, respectively. Two empirical studies [SK17, NS16] that were conducted with density-based sampling algorithms deploy KNN to distin-

guish the type of each instance along with multi-dimensional scaling (MDS) [Kru64], which is a global linear dimensionality reduction algorithm. We follow the same methodology to characterize instances based on local characteristics, but HARDVIS uses an interactive UMAP projection [MHM18] since it preserves better the local structure [EMK*21]. Although those studies suggest that applying sampling techniques in specific types of instances (*e.g.* by using only outliers) can boost predictive performance, controlling which subsets of particular instance types are considered when undersampling and oversampling is an unexplored step. This research opportunity inspired us to design HARDVIS.

Density-based algorithms [HHHM11, HLL08] also work well with the detection of rare categories by discovering substantial changes in data densities using a KNN search in the high-dimensional space. But how to choose the best k-value for a given data set? While it is possible to estimate the best k-value automatically by using the local outlier factor [BKNS00], the balance of the distribution of safe and unsafe instances could be off when focusing merely on rare cases and outliers. Huang *et al.* [HCG*14] proposed a method for automatically selecting k-values. However, their algorithm starts with a seed depending on the target category, which is often difficult to set. iFRED and vFRED [LCH*14] are two approaches for identifying rare categories based on wavelet transformation without the necessity of any pre-defined seed. Nevertheless, these methods are robust in low-dimensional data only but fail to discover the remaining types of data introduced in Section 1, which are important for HARDVIS. Regarding decision boundaries and borderline examples, Melnik [Mel02] analyses their structure using connectivity graphs [MS94]. And finally, Ramamurthy *et al.* [RVM19] utilize persistent homology inference to describe the ambiguity (or even lack) of decision boundaries. All described methods, while being valuable, do not focus on the problem of undersampling or oversampling at all, as it happens with our system.

2.2. Visualization for outlier and rare category detection

Numerous VA approaches are combined with detection algorithms as described in Section 2.1. Usually, they are designed for supporting outlier and rare categories identification and classification, which could be considered relevant to our work. Oui [ZCW*19] is a tool that assists users in comprehending, interpreting and selecting outliers identified by multiple algorithms. #FluxFlow [ZCW*14] is another VA system that utilizes complex analytical methods to find, summarize and understand aberrant information spreading patterns. TargetVue [CSL*16] detects users with abnormal behaviours using the local outlier factor and intuitive behaviour glyph designs. An extension of such glyphs, named as Z-Glyph [CLGD18], was developed to aid human judgement in multivariate data outlier analysis. RCLens [LGG*18] is an active learning system that uses visualization approaches to support the discovery of rare instances. EnsembleLens [XXM*19] is a hybrid visual system that utilizes a modified Gaussian mixture model [AY19] to identify problematic patterns in human behaviours. RISSAD [DB21] is an interactive approach that not only assists users in detecting abnormalities but also automatically defines them using descriptive rules. Even border detection has recently gotten some attention thanks to a VA method [MM21] which uses the power of explainability from linear

projections to help analysts study non-linear separation structures. However, the final goal of HardVis differs since we try to merge the gap between instance hardness and sampling techniques for evaluating their suggestions. None of the above VA systems incorporate sampling mechanisms, as defined in Section 1.

VERONICA [RAS*21] is a domain-specific VA system that uses undersampling and SMOTE for specific classes of data and groups of features. On the other hand, HardVis is inherently designed to be generalizable to any numerical data set stored in a tabular form. It also accounts for instance hardness while enabling the micromanagement of the sampling techniques. To improve the efficiency of model construction, Li *et al.* [LFM*18] presented a VA approach that allows infusing dynamic user feedback in various forms, with interactive addition of new samples being one of them. Despite that, the goal is different from HardVis since the focus is on learning with a limited amount of data or incrementally learning as in Paiva *et al.* [PSPM15]. During the main use case of RuleMatrix [MQB19], Ming *et al.* manually selected a problematic subset of instances and applied oversampling, which resulted in improved model accuracy. In contrast, HardVis enables knowledgeable users to systematically explore the distribution of data in different types and the suggestions of the undersampling and oversampling to enhance predictive performance.

2.3. Data-centric machine learning

Most of the model-centric ML work so far has focused on how model developers incrementally improve an existing or newly invented ML algorithm's predictive performance while making no changes to the collected data [Ham22]. On the other hand, practitioners of data-centric ML maintain the ML model stable while iteratively upgrading the quality of the data at hand [Ham22]. Advocates for data-centric ML have recently increased in volume. A few reasons for this shift are the benefits of involving domain experts in the data analysis process and the necessity for very configurable solutions that focus on subsets (or *slices*) of data [Ham22]. Closely related to this paradigm, ModelTracker [ACD*15] and Squares [RAL*17] are two interactive visualization approaches that improve a more standard confusion matrix to detect issues with particular instances and enable users to tune the input by monitoring the output of the model. The former proposes a visualization that incorporates information from a variety of typical descriptive statistics while providing instance-level performance and allowing for direct error analysis and troubleshooting. The latter computes performance measurements and assists users in concentrating their efforts on instance-level issues. Therefore, both works follow the general framework of visual parameter space analysis (vPSA) [SHB*14]. Although HardVis is also an applied example of the vPSA framework, it is explicitly designed for the first stages of an ML model-building pipeline, addressing a clear need for applying sampling techniques in specific types of instances only.

Active learning is also part of data-centric ML solutions. It can be defined as the active usage of a learning algorithm to iteratively suggest to a user to classify unknown instances in order to increase the ML model's performance quickly [Set12]. In the visualization community, many VA techniques have been developed explicitly for active learning [BZL*18, BHS*21, BHZ*18, GBSW21]. More specif-

ically, these works have focused on how VA can help users during the labelling process for semi-supervised learning problems. The challenges are somewhat similar to ours since understanding how hard (or important) it is for an instance to be labelled before the rest is a relatable problem. However, our end goal is to prioritize which instances should be undersampled and oversampled first (and how exactly) in supervised learning classification problems containing labels for all data instances.

3. Analytical Tasks and Design Goals

This section outlines the basic analytical tasks (**T1–T5**) that a user should be able to complete when undersampling or oversampling while using a VA system for support and direction. Following that, we present the design goals (**G1–G5**) that guided the development of HardVis.

3.1. Analytical tasks for undersampling and oversampling

From the in-depth examination of the related work highlighted in Section 2 and our own recent experiences implementing VA tools for ML [CMK20, CMKK21a, CMKK21b, CMKK22, CMK23], we came up with five analytical tasks.

T1: Identify the various types of instances. As the decrease in predictive performance is connected to data distribution-related factors, such as the presence of many rare subgroups obscuring the classification [WH00, Jap01], the consequences from the overlap between the classes [PBM04, GSM07] or the existence of several misclassified examples [NSW10], a primary goal is to spot such groups of points—as precisely as possible—with the use of VA systems.

T2: Support the exploration of undersampling versus oversampling alternatives applied globally and locally. When applying such techniques, the data instances used as input for undersampling and oversampling algorithms could differ depending on the stable anchors a user sets. An example of a stable anchor is how the partitioning of data into four types occurs, leading up to 16 different SBRO combinations used as input for the sampling algorithms. Also, the distribution of SBRO (as defined in **T1**) is another factor to be considered as a stable anchor under investigation. On the one hand, global undersampling or oversampling will allow all instances to be candidates for removal or under consideration when creating synthetic data, respectively. On the other hand, locally applied algorithms will dynamically enable users to consider local characteristics of data points and exclude a few suggestions from the pool of recommendations. Modifying this ratio dynamically could be beneficial for the ML model, thus the user's interaction guided by visual feedback is necessary.

T3: Explore automated methods' suggestions. The identification of conditions for the efficient use of a particular method is an open research problem [NS16]. A user should be competent in judging the influence of a suggestion on the whole data set. For example, what if, by removing too many rare cases, the model overfits the training data but generalizes poorly in a test set? A user should be empowered by VA systems that facilitate exploratory analysis of unsafe instances.

T4: Confirm suggestions by making justifiable decisions. A user should have the ability to partially confirm the proposal of the automated methods based on the analysis he/she has performed earlier in the preceding task. How will the data distribution change due to the acceptance of such a suggestion? VA systems should envision these future steps and enhance users' decision-making.

T5: Monitor and evaluate the results of the sampling process. At any stage of the sampling process (**T2–T4**), a user should be able to observe performance fluctuations with the use of appropriate validation metrics for imbalanced data sets (*e.g. balanced accuracy* and *f1-score*). A user might also wish to look back at the history of activities to see if any crucial actions corresponded to better results. Thus, VA systems must be capable of providing ways to monitor performance.

3.2. Design goals for HARDVIS

We identified five design goals for our system to meet in order to fulfil the more general aforementioned analytical tasks for undersampling and oversampling. We implemented them in Section 4.

G1: Visual examination of several data types' distributions and projections to choose a generic 'number of neighbours' parameter. Our goal is to assist in the search for distinctive distributions of data types that might consider different populations of SBRO instances (**T1**). By systematically modifying the *number of neighbours* parameter of UMAP, we aim to assure that users will pick a better value based on the visual exploration of data types in the generated projection. Furthermore, this value propagates in the undersampling and oversampling techniques that require a *k-value*, which works similarly to the above parameter.

G2: Application of undersampling and oversampling in specific data types only, with different parameter settings. There are several different undersampling and oversampling techniques, but they are usually only applied to the entire training set (*i.e.* global sampling). However, with our proposed system, we enable users to choose a technique, tune the parameters depending on the visual exploration and even deploy them in particular subsets of the training data (*i.e.* local sampling as established in **T2**).

G3: Exploratory data analysis of unsafe suggestions. Next, the system should provide sufficient visual guidance to users to focus on the exploration of the values in each feature for unsafe suggestions (**T3**). The analysis of borderline, rare and outlier data types should be feasible in a generic and detailed manner.

G4: Comparison of trade-offs while removing or adding training instances throughout the decision-making process. After the extraction of evidence as defined in **G3**, users should see how the distribution of instances will change due to the undersampling and/or oversampling phases. Next, the system should give a prediction for a data point and juxtapose it to all other points. With this, users should be able to estimate the impact of algorithmic recommendations during exclusion or inclusion of instances (**T4**).

G5: Keep track of critical steps and evaluate predictive performance in general and for specific test instances. Users' inter-

actions should be tracked in order to preserve a history of modifications in the training set, and the performance should be monitored with validation metrics (**T5**). Finally, using an unseen test set, the system should continuously stress the difference in the model's predictive performance.

4. HardVis: System Overview and First Application

Following the analytical tasks and the resulting design goals, we have developed HARDVIS, an interactive web-based VA system that allows users to identify areas where instance hardness occurs and to micromanage sampling algorithms. Section 7.2 contains further implementation details.

The system consists of eight interactive visualization panels (Figure 1): (a) data types projections ($\rightarrow$ **G1**) incl. data sets and sampling techniques ($\rightarrow$ **G2**), (b) data overview, (c) data types distribution, (d) data details, (e) data space, (f) predicted probabilities ($\rightarrow$ **G3** and **G4**), (g) sampling execution tracker and (h) test set confusion ($\rightarrow$ **G5**). We propose the following **workflow** for the integrated use of these panels (*cf.* Figure 2): (i) explore various projections with alternative distributions of data types, leading to the division of training data into SBRO (*cf.* Figure 3(b)); (ii) in the undersampling or oversampling phase, tune the active algorithm's parameters to affect specific types of data (Figure 1(a)); (iii) during the confirmation phase, identify which suggestions will impact negatively or positively the predictive performance and approve or reject any suggestion (*cf.* Figures 1(b)–(f)) and (iv) store every manually operated sampling execution, identify confused test instances and compare the predictive performance in each step of the process according to two validation metrics designed explicitly for imbalanced classification problems (Figures 1(g) and (h)). These steps are iterative, and they might occur in any sequence. The created knowledge obtained from the undersampled/oversampled data set is the end result. This knowledge can be useful to users that have to explain and are accountable for their actions, *e.g.* people working in critical domains such as medicine.

HARDVIS employs a state-of-the-art ensemble learning approach named as XGBoost [CG16], and its workflow is model-agnostic. To make our approach even more future-proof, we train this ML algorithm with the *Bayesian Optimization* package [Nog14]. HARDVIS utilizes OSS, NCR, SMOTE and ADASYN, which are state-of-the-art sampling algorithms that are tweaked to receive specific SBRO instances as an input. Despite that, these algorithms are easily replaceable. The reader is referred to Refs. [HG09, HM13] for a more detailed analysis of different strategies that cope with class imbalance. For this section and the use cases in Section 5, we split the data sets into 75% training and 25% testing sets with the stratified strategy (*i.e.* keeping the same balance in all classes for both sets) and validate our results with 5-fold cross-validation. Also, we scan the hyperparameter space for 25 iterations, choosing the model with the best accuracy. The hyperparameters we used are the same as in another VA system developed by us [CMKK22].

In the following subsections, we explain the system by using a running example with the *iris flower* data set [FIS36] obtained from the UCI ML repository [DG17]. The data set represents a balanced multi-class classification problem and consists of four numerical
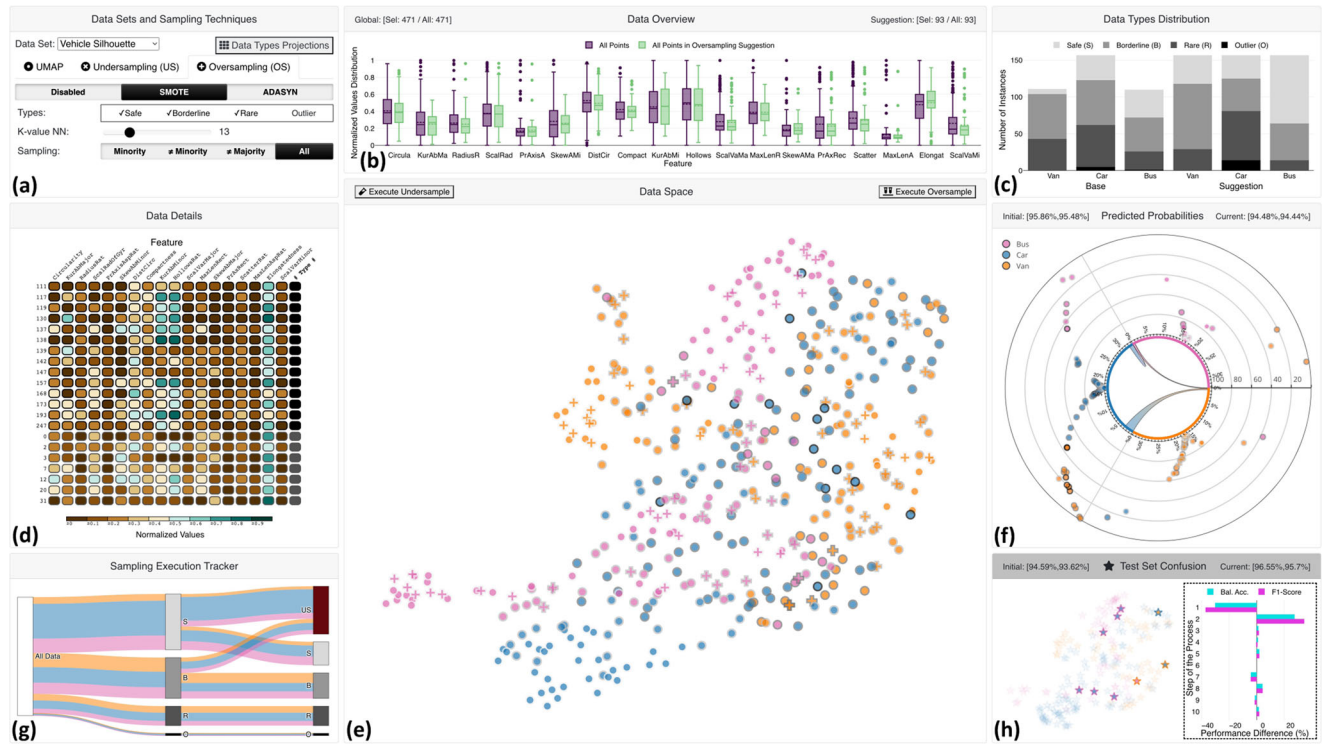
Figure 1: *Undersampling and oversampling certain data types with* HardVis*: (a) the panel with many tunable parameters for UMAP, undersampling and oversampling; (b) box plots for comparing the values of all points against the algorithm's suggestion in each feature; (c) a stacked bar chart showing the base* versus *the new distribution if the suggestion is approved; (d) a table heatmap view for comparing the instances' values across all features; (e) a UMAP projection emphasizing the additions/deletions of points, along with the data type for every instance; (f) an inverse polar chart with chords that depicts the predicted probabilities, as well as the training confusion; (g) a Sankey diagram for tracking any undersampling or oversampling confirmed actions and (h) a visual embedding based on (e) to highlight the confusing test instances, and a horizontal bar chart to illustrate the performance difference for each step.*

features and 150 instances. The three classes are: *setosa*, *versicolor* and *virginica*.

4.1. Data types

HardVis follows the Napierala and Stefanowski [NS16] methodology in order to label all training instances in one of the following types: safe (S) examples, borderline (B) samples, rare (R) cases and outliers (O). To calculate the difference between instances in the high-dimensional space, we use KNN [Alt92, FH89] with the *default* value of k being 5 and the *Euclidean* distance metric. For determining the type of a sample with $k = 5$, we would have, *e.g.* five or four nearest instances being from the same class, then the sample gets labelled as S; three or two instances from the same class, then it belongs to B; only one instance from the same class, it is R; and zero (*i.e.* the five nearest instances are from the other class), it becomes O. However, the analogies will change with $k > 5$.

As shown in Figure 3(a), stacked bar chart, the distributions of instances change accordingly as the *number of neighbours* in the UMAP [MHM18] shifts since we utilize the same value for the KNN algorithm. Thus, the goal of the two-dimensional projection is to reflect visually the same separation of training instances into

the SBRO types. The *minimum distance* is another parameter of UMAP that (in our case) is being automatically computed from the maximum achievable Shepard diagram correlation (SDC) [CMK20] score (see Figure 3(a), line chart). This metric serves as a first indicator of optimal distance preservation between the low- and the high-dimensional space. Nevertheless, it cannot be trusted blindly, and human exploration is necessary to conclude which parameters are optimal for the given data set.

The main challenge of KNN is the user-selected k-value, thus it is a highly parametric-dependent approach. To resolve this problem, we enable the user to explore different data types' projections generated by the systematic change of k-value from 5 to 13 (*cf.* Figure 3(b)). This range is chosen intentionally because, in low k-values, a slight modification is more impactful to the projection [NS16]. However, these values are adjustable within the code.

4.2. Undersampling

Figure 3(c) presents the tab for Undersampling (US), which along with the standard method's parameters comprises a Types menu with options to exclude any SBRO group. The k-value is automatically tuned due to the selection of the number of neighbours
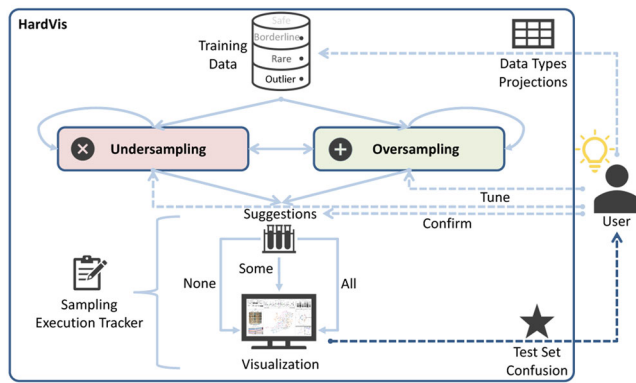
Figure 2: *The* HardVis *workflow starts by classifying the training data into four types according to the user's visual inspection of nine alternative projections. The data are sent for either undersampling or oversampling, which can make suggestions continuously. The user's confirmation is requested after the exploratory data analysis through the visualizations.*

parameter, as explained in Section 4.1. OSS [KM97] uses Tomek links [Tom76] which are ambiguous points on the class boundary that are typically identified and removed. Moreover, it employs the condensed nearest neighbour rule [Har68] to remove redundant examples far from the decision boundary. In contrast, NCR [Lau01] is an undersampling technique that combines the condensed nearest neighbour rule to exclude redundant examples and the edited nearest neighbours rule [Wil72] to remove noisy or ambiguous points. Its main difference from OSS is that fewer redundant examples are deleted, and more attention is placed on 'cleaning' those retained instances. Each algorithm expects input for a unique parameter. In particular, NCR has *Threshold* which is used for deciding whether to consider a class or not during the cleaning after applying edited nearest neighbours. *Seeds* is the number of samples to extract in order to build a set S for OSS. All these techniques can be employed in the Majority, ≠ Minority, ≠ Majority, or All classes according to the user's choice. In multi-class classification problems, the Majority will be merely the class that contains the most instances, ≠ Minority will be all classes except the one with the least instances, and so on. In balanced data sets, only the All option is relevant.

The UMAP projection in Figure 3(d) allows users to observe the type of each instance concurrently and if it was suggested for removal with an '×' symbol or addition with a '+' mark by the active undersampling or oversampling algorithm, respectively. The parameters for the UMAP are set as discussed in the preceding subsection. Hovering over a point will present details on demand such as the ID of the point, the predicted probability and the values for each feature.

The distribution of data types is known due to a stacked and grouped bar chart with the instances distributed in SBRO and per class, simultaneously (*cf.* Figure 3(e)). The base distribution is also comparable with the suggestion from the sampling algorithm that will modify the initial distribution.

Figure 3(f) is a box plot that facilitates the comparison of all points per feature versus the selected points via lasso functionality in the projection. When a sampling algorithm is active, the same group of instances with merely the sampling suggestions is also visualized. In case of no selection, a simpler version of all points against all points in either undersampling or oversampling suggestion exists (see Figure 1(b)). Users' actions determine the mode automatically. The features are sorted from left to right, from the least important to the most important at each execution step of undersampling/oversampling (due to XGBoost retraining process). The proposals for removal are denoted in light red colour, and light green is used for the suggested additions.

The table heatmap view in Figure 3(g) is a more detailed view of the aggregated results present in the box plots. It normalizes the values from 0 to 1, evident in dark brown to dark teal colours, and it shows for each feature the current value in each instance. The features are sorted as in the box plots. Moreover, the *# Type #* is perceivable through this visual representation, with outliers, then rare cases, next borderline examples, and finally safe instances being at the top of the list. The selection of a specific feature in this view applies the diverging colourmap to the projection for comparing all instances for this particular feature (see Figure 6(d), zoomed in view). More detailed discussions on the visual design behind some of the views can be found in Section 7.1.

The inverse polar chart in Figure 3(i) is deliberately designed to provide more space to instances that are in the borders between two classes or completely misclassified cases. The predicted probability with the ground truth class is used for the 100 to 0 axis, and the angle/orientation is computed as the difference in predicted probability of belonging to the remaining two classes. The greater this difference is, the farther a point deviates from the centre of its circular segment corresponding to the correct class label. In our example, the versicolor has a few instances mainly confused with the virginica and vice versa. This is why all setosa instances are near 100% predicted probability in the purple circular segment. The size of each piece is calculated from the number of training instances that belong to a particular class, with extra space being provided to larger classes (*i.e.* consisting of more points). The same symbols as in the projection are also retained here. This approach can easily work for two or three classes but becomes challenging to interpret with more classes; such limitations are discussed in Sections 6 and 7.2. The centrepiece of this visualization is a chord diagram that summarizes the confusion matrix for the training data, as in Alsallakh *et al.* [AHH*14]. Thus, in Figure 3(i), the confusion between versicolor and virginica is immediately distinguishable by the chords linking the different circular segments. The number of confused instances from one to the other classes is encoded as chord width.

4.3. Oversampling

Two mainstream oversampling techniques are implemented in HARDVIS. SMOTE [CBHK02] finds the KNN in the minority class for each of the samples in the class. Next, it draws a line between the neighbours and generates random points on the lines. ADASYN [HBGL08] is the same as SMOTE, just with a minor
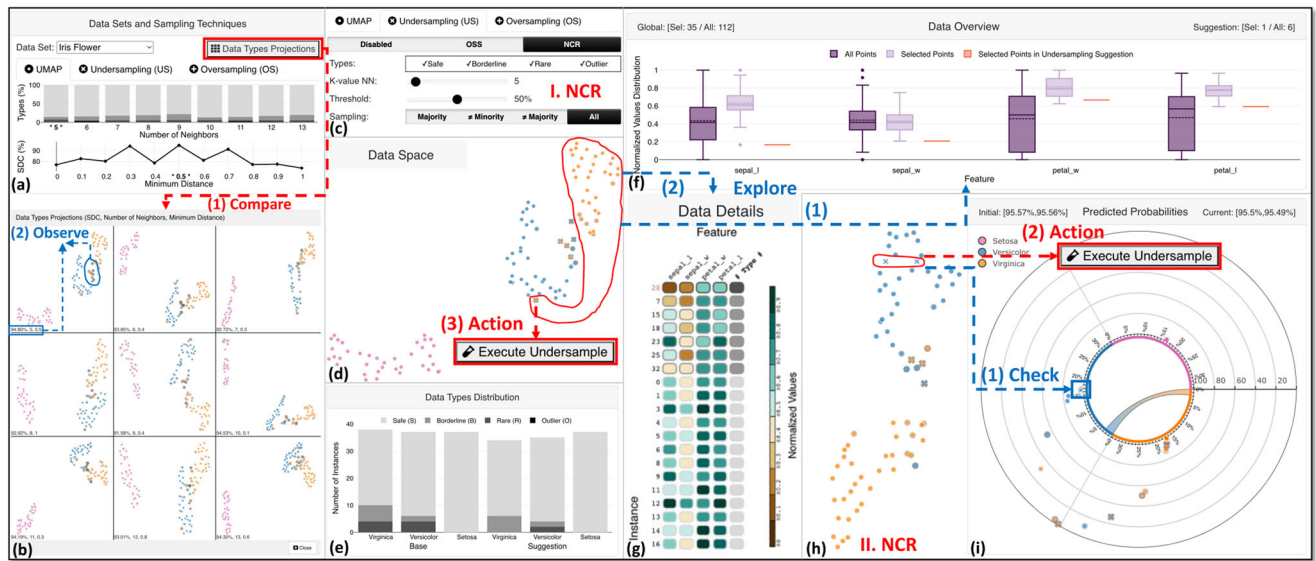
Figure 3: *At first, a comparison of different data types projections and then two consecutive undersampling phases with the NCR algorithm are shown in this arrangement of screenshots. The* default *value for the number of neighbours is 5 (see (a)), which is used as input for computing the type of each instance with KNN. The projections are generated by systematically tweaking the above parameter, as illustrated in (b); the best choice is theoretically the highest value for the Shepard diagram correlation (SDC) metric. In (c), we have activated the algorithm, and we check the impact of this automated technique on the projection in (d). (e) presents the difference in distributions of all data types per class label from when the algorithm was inactive as opposed to its activation. In (f), we explore a specific rare case under removal consideration. This instance is contrasted against the remaining points of this same class (i.e. virginica in orange colour); the selection was made using a lasso interaction, as demonstrated in (d). While the values for all features are lower for this sample than the rest,* sepal_l *appears the furthest away. Additional details can be found in (g) that highlights these differences in values of particular features and confirms our findings from the data overview. Consequently, we choose to delete this instance because it might cause further confusion to the model, as depicted in (d). The second time we deploy NCR (cf. (h)), two safe instances are in our focus since they are easily classified due to the high predicted probability visible from the inverse polar chart in (i). Therefore, we decide to remove these two points.*

improvement. After creating those samples, it adds a small random deviation to the points, thus making it more realistic with the additional variance. Similar to the undersampling techniques before, SMOTE and ADASYN have all options except for the division of types provided via a separate menu of our system. The All option is equivalent to ≠ Majority, but we implemented them differently when a type of instance is deactivated. The former considers removing all points of the specific deactivated type/s irrelevant to the class that will be oversampled, leading to more excluded points for the active algorithm. The latter excludes from the pool of points only those from the deactivated type/s but from the class that will be oversampled. The Minority and ≠ Minority are implemented based on the second schema described here. The same exploration and analysis options mentioned in the previous subsection also apply for oversampling.

### 4.4. Sampling execution tracker and test set confusion

Each manual undersampling or oversampling confirmation is registered in the Sankey diagram (see Figure 4(d)). The initial setting is to record the distribution of all training data to the SBRO types. Then, as an undersample or oversample execution takes place, the instances move from their type to the US (in dark red) or OS (in dark green) bin of the Sankey diagram.

The test set is also plotted using the visual embedding of training data in each step (*cf.* Figure 4(e), left). All test instances are transparent when predicted correctly by the ML model and opaque in cases of confusion. For example, in Figure 4(e), left, the star with blue colour is from the versicolor class, but it was predicted as virginica due to the orange outline. Furthermore, the initial and current balanced accuracy (bright turquoise) and f1-scores (deep magenta) are visible in the text at each side of the heading of the Test Set Confusion panel. The difference in performance based on those metrics is tracked for every step of the process with a horizontal bar chart (Figure 4(e), right).

4.5. First application

In our first application, we observe that the maximum SDC value is 94.80% (high correlation, Figure 3(b)), resulting in a most probably trustworthy projection. Another reassurance stems from the visual inspection of points in the middle of two classes that appear clearly confused, with most rare and borderline instances being located there.

The undersampling phase is perhaps most crucial since removing unsafe instances without justifying one's action could cause a severe issue to the ML model. We choose to activate the *de facto*
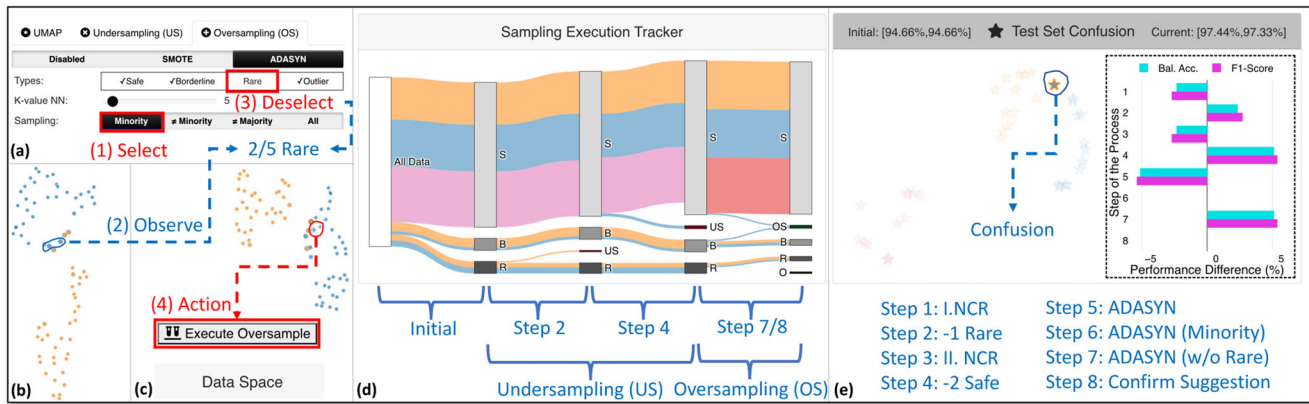
Figure 4: *An oversampling phase that aims to balance the data set again. According to (a), we use ADASYN for the minority class (versicolor in blue) that contains fewer instances. Also, we exclude from the input of the algorithm two rare cases near the borders of two classes, as illustrated in (b). The system proposes two artificially created points for addition that we approve, see (c). The Sankey diagram in (d) summarizes the core execution phases with undersampling and oversampling steps. Only one test instance is confused according to (e), while the manual decisions (steps 2, 4 and 7/8) improved the balanced accuracy and f1-score scores compared to the automated methods (steps 1, 3 and 5).*

NCR algorithm without any tweaks to check the suggestions (Figures 3(c) and (d)). The distribution of instances changes according to this global suggestion for removal of orange and blue points, as seen in Figure 3(e). Despite that, we want to explore further a suggestion for removal that is a distant point from the core virginica cluster. We use the lasso to select those points and proceed with the investigation. The box plot in Figure 3(f) enables us to conclude that this is an extreme case relatively different from the remaining selected points of its own class since the values for all features are very low. The table heatmap view in Figure 3(g) reaffirms our hypothesis because the instance with ID 28 has the lowest *sepal_l* value (<0.1 due to dark brown colour). We exclude this instance, but we keep the rare cases around the borders of the two classes that can easily flip class labels. Another phase of undersampling is also capable with HardVis since the new data become the ground zero for the next application of the automatic algorithm; NCR is again our choice. This time, five instances are proposed for deletion (*cf*. Figure 3(h)). However, by checking the inverse polar chart in Figure 3(i), we see that two of them are easily predictable and potentially redundant for the ML model. Therefore, we decide to exclude those two safe samples solely.

Using the Oversampling (OS) tab, we try to balance the classes that contain fewer training samples. In Figure 4(a), we activate ADASYN for the minority class, which requires two more examples to restore balance in the training set. This setting, in combination with the observation of two rare cases that are in the borders of the versicolor class (Figure 4(b)), leads to the deselection of the Rare type. Consequently, these two rare cases are excluded from the pool of available for oversampling training instances. Without the appropriate choice of *k*-value, resulting in an expressive and effective distribution of data types, it would have been challenging to detect and handle such cases (especially if no class labels were provided). The oversampling generated two instances that we accept in step 8, as depicted in Figure 4(c).

In Figure 4(d), the deletion of one rare instance during the first NCR phase, the removal of two safe instances during the second NCR phase and the oversampling phase utilizing ADASYN for generating a safe and a borderline instance is visible. Also, the confusion of a test instance is highlighted in Figure 4(e) with the decisions of the automatic algorithm hurting the performance and the manual decisions in steps 2, 4 and 7/8 improving the predictive power of the ML model.

5. Use Cases

In this section, we present a hypothetical usage scenario and a use case about how HardVis can evaluate suggestions based on local data characteristics to build trust in ML and to improve the balanced accuracy and f1-score scores for both training and testing sets.

5.1. Usage scenario: local assessment of undersampling

Zoe is a data analyst in a hospital, working primarily with healthcare data. She receives a manually labelled data set with nine features related to *breast cancer* [DG17]. This data set is rather imbalanced, with 458 *benign* and 241 *malignant* cases. From her experience, she knows that instance hardness and class imbalance can be troublesome for the ML model. Thus, she wants to experiment with well-known algorithms for undersampling and oversampling the data. However, especially with medical records, the use of merely automated methods is questionable because they cannot be trusted blindly. The doctors need explanations, and the minority class in this binary classification problem is of more importance than the majority consisting of healthy patients. In reality, patients who are healthy but predicted as ill will undergo extensive follow-up diagnostic tests before treatments such as surgery and chemotherapy are advised; however, the opposite is not true. To accomplish this main objective and to control the sampling techniques, Zoe deploys HardVis.
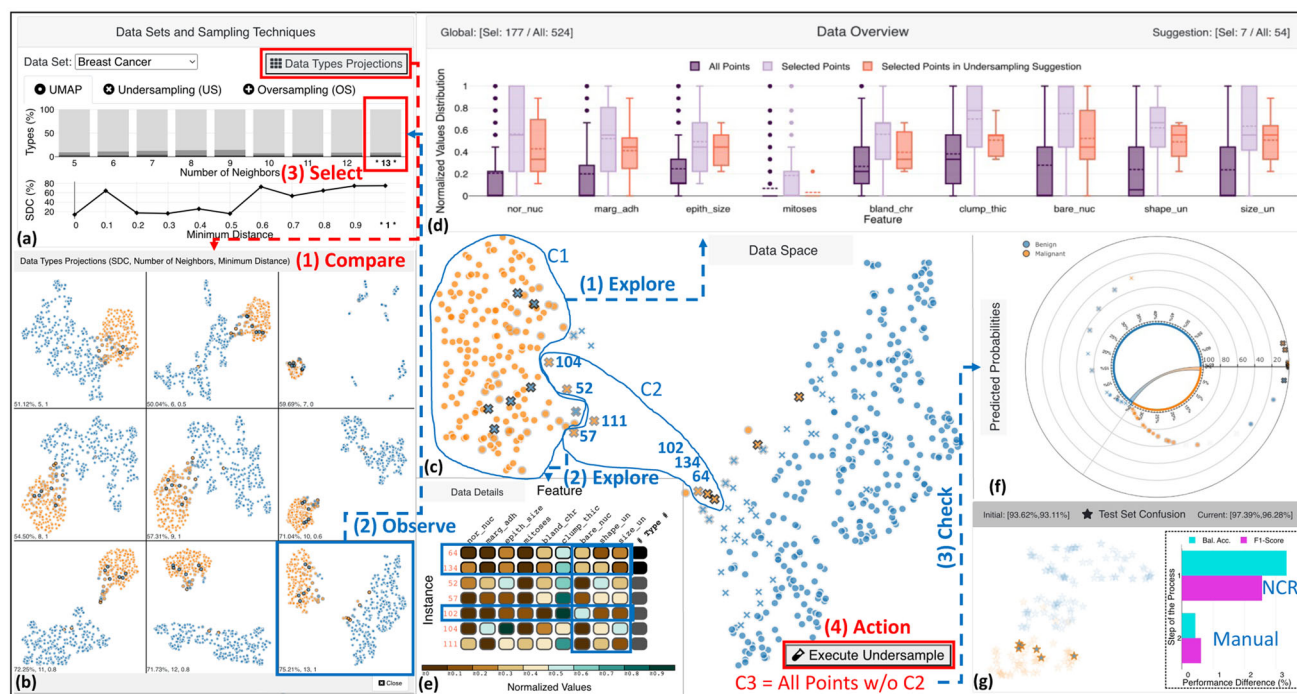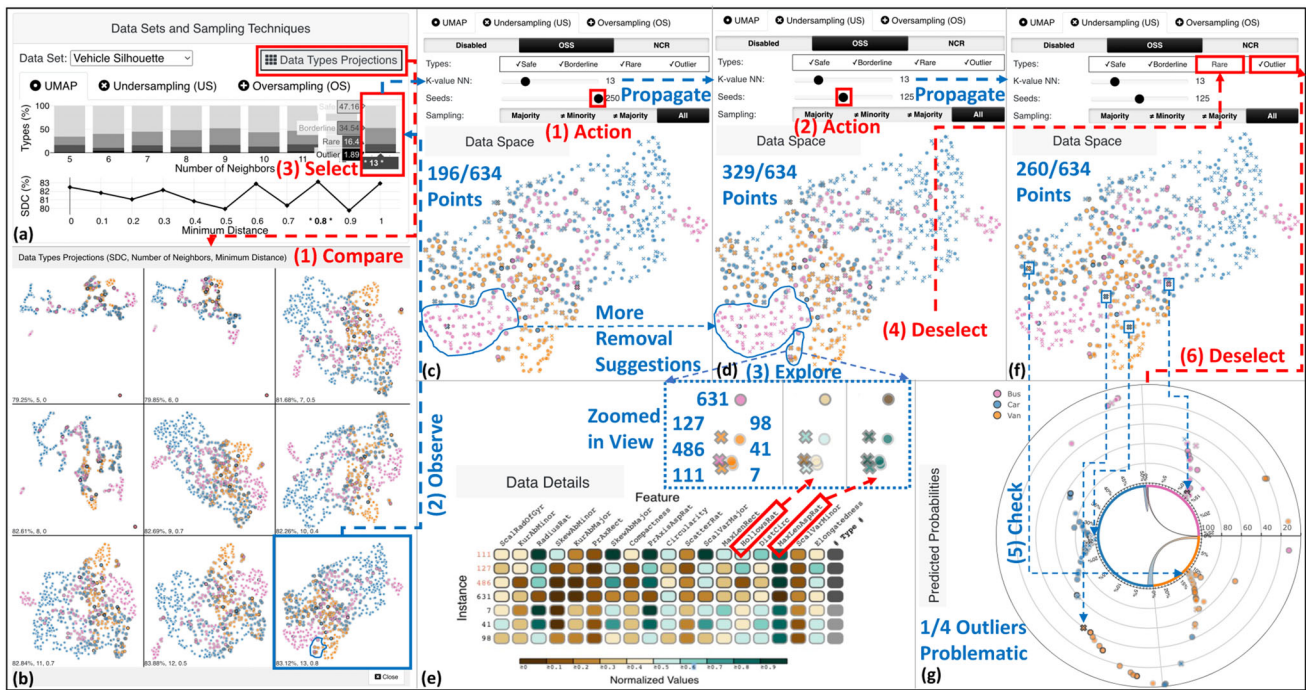
Figure 5: *The investigation of diverse structures of data types and alternative suggestions in an undersampling scenario. View (a) shows the selection of the number of neighbours value of 13, which has 75.21% Shepard diagram correlation (SDC) score, as illustrated in (b). The UMAP visible in (c) has one rare sample and six outliers belonging to the benign class that holds relatively normal values compared to the malignant cluster (C), as shown in (d). Therefore, the suggestions for removal in C1 are valid since even humans cannot understand why these points are benign cases. On the other hand, C2 contains five rare examples and two outliers that serve as a bridge between the two classes,* cf. *(c). Interestingly, the three most important features differentiate the right group of points (IDs: 64, 102 and 134) from the left,* i.e. size_un*,* shape_un *and* bare_nuc *in (e). This diversity is crucial when predicting difficult to classify instances, hence the analyst chooses to keep this cluster despite the NCR algorithm's suggestion for removal. C3 is the final selection, with most outliers being removed because the model badly predicted them, as seen in (f). This leads to (g), which presents an improved performance with six confused test instances that are cancer-free but predicted as the opposite. The malignant class is secure due to the rare cases being intact.*

Choosing an accurate projection. Zoe begins with the selection of a *number of neighbours* parameter by activating a window containing data types projections (*cf*. Figure 5(a)). HARDVIS enables her to compare a grid of diverse projections, as presented in Figure 5(b). The one with the highest SDC score (*i.e.* 75.21%) is a noteworthy candidate because the two classes are clearly separated. Rare cases and outliers are also easily visible, forming a bridge between benign and malignant instances. She clicks on the bar with the number 13 in Figure 5(a), and this projection becomes the main for further exploration. At this initial phase, six benign test instances were incorrectly classified, while the remaining four out of the 10 misclassified patients were actually malignant cases.

Examining unsafe instances proposed for removal. Afterwards, she activates the NCR algorithm with the default settings (*k*-value is synchronized to 13 due to the previously selected projection) from the Undersampling (US) tab. Cluster 1 (C1) in Figure 5(c) is interesting because seven benign cases (mostly marked as outliers) are in between the malignant class. She chooses to compare the selected points in C1 against these suggestions of undersampling, as depicted in the box plots (Figure 5(d)). In summary, the values are lower for these points but still in between normal

margins. Therefore, it would have been almost impossible for the doctors to conclude that these are healthy patients with benign cancer. A thorough check should be performed in these cases, *e.g.* to determine if the labels are erroneous. She first notifies the data collection team and doctors about this important finding and then removes C1 suggestions. On the contrary, C2 includes five rare cases and two outliers with *size_un*, *shape_un* and *bare_nuc* features separating the points closer to the benign class from the rest, as illustrated in the table heatmap view (Figure 5(e)). The right group of points has mostly lower values for the *size_un* and *shape_un* features, while the *bare_nuc* is higher compared to the points on the left. Zoe understands that such diversity is important when dividing borderline patients located at the conjunction of the two huge clusters. Therefore, she uses lasso selection to grab all points except for C2, which will be her manual undersample strategy. The inverse polar chart in Figure 5(f) highlights the training instances that will be deleted, which are mostly completely misclassified instances or safe examples. The samples between the two classes already explored remain intact, which is essential since they all belong to the more important minority class. In Figure 5(g), Zoe observes that only six test instances were incorrectly classified as having malignant cancer while they were healthy. When inspecting the balanced accuracy

Figure 6: *The examination of diverse structures of data types and alternative suggestions while undersampling with OSS. View (a) shows the selection of the number of neighbours value of 13, which is also used as an input to KNN for sampling similarly in the high-dimensional space. This decision was made after a careful review of the nine projections in (b), leading to a distribution of mostly safe instances, then borderline examples, next rare cases, and finally a few outliers. This is the second-best projection in terms of Shepard diagram correlation (SDC) score, but it preserves exceptionally well the clusters of buses in purple versus vans in orange colour at the bottom-left region. In (c), we experiment with the maximum seed value for the OSS algorithm, but it seems that several safe instances that could have been proposed for removal are actually not. Thus, we reduce the seed value in half to check if the new suggestion fits our viewpoint, as depicted in (d). The goal we set has been accomplished, however, a cluster of rare cases mixed in two classes is about to be deleted. In (e), we investigate further this group of points at the bottom-left corner; these instances differ mainly because of either* MaxLensAspRat *or* HollowsRat. *Such rare cases are critical information for the model to split these two classes; hence, we exclude the rare cases from the automatic algorithm, see (f). From the remaining 260 under consideration instances, there are four outliers that caught our interest. Only one out of the four outliers appears problematic based on (g). As a result, we exclude the outliers from the analysis and Execute Undersample to the remaining 256 points.*

and F1-score scores, the overall predictive performance for the test set seems slightly improved contrasted to the automatic algorithm. Nevertheless, the major gain is that the doctors might trust this modified data set more because the model correctly predicts all patients with malignant cancer (since there is no highlighted yellow star for the test set in Figure 5(g), left). Based on prior findings [NS16], Zoe stops her exploration at this phase because oversampling is ineffective for data sets with mostly safe instances.

5.2. Use case: explorative sampling for better classification

This use case is about a multi-class classification problem. There are 18 features collected for the *vehicle silhouettes* data set [Sie87]. With the main task of classifying 199 *vans*, 218 *buses* and 429 *cars*, the class distribution is somewhat unbalanced.

Comparing projections and distributions of data types. Similar to the procedure described in Section 4, we start by exploring which projection represents the data types in the best possible way. Three projections reaching high enough SDC with *minimum dis-*

tance parameter equals to zero are extremely condensed, making it hard to observe anything (see Figure 6(b)). Among the remaining, two of them have SDC score of more than 83%. Although they are two similar projections, the last one clearly shows the difference between bus and van classes in purple and orange, respectively (see the circled area at the bottom). We choose to continue with this projection; thus, we go back to Figure 6(a) and select a *number of neighbours* parameter equal to 13. When we hover over the stacked bar chart in Figure 6(a), we observe that safe and borderline cases account for 47.16% and 34.54% of the training set, respectively. This is significantly different in the distributions of data created with lower values for the number of neighbours (*e.g.* 5). In summary, the visual analysis guides us in picking all the aforementioned parameters.

Tuning the undersampling based on exploratory data analysis. After selecting the projection (which results in a specific distribution of data types), we decide to apply the OSS undersampling algorithm. Nevertheless, the default settings cause the van class to disappear completely, thus the predictive performance gets extremely

penalized (see step 1 in Figure 1(h), right). We pick the highest available *seeds* parameter to consider more points except for the minority class. The algorithm suggests 196 instances to be removed (step 2), as illustrated in Figure 6(c). It seems from the projection that our previous setting does not capture several buses while being safe to remove examples. Therefore, we decrease the parameter to 125, half of the prior selection. The effect is that 329 are currently suggested for removal (step 3), as depicted in Figure 6(d). This action accomplishes our initial goal, but 7 regional points are about to be undersampled. As we should be very careful when deleting rare cases, we further explore this group of points in the table heatmap view (Figure 6(e)). It is observable that the instance with ID 486 is separated from the others mainly due to the *HollowsRat* feature, while instance 631 is different because of a low value in the *MaxLensAspRat* feature. We decide not to exclude rare cases with such high variance because they may be part of our test set (step 4). The new suggestion after excluding the rare points is visible in Figure 6(f). Another critical category of data types is the outliers. From all outliers in the last projection, four are proposed for deletion by the oversampling algorithm. Only 1 out of the 4 points appears marginally problematic with prominent confusion between car and van classes, as depicted in the inverse polar chart (see Figure 6(g)). Since the majority of points is safely predicted correctly, we decide to keep the outliers in the training set. After this step, 256 out of the 634 points are getting removed (steps 5 and 6).

Deciding to oversample all types except outliers. To understand if a new round of undersampling would be beneficial, we activate the OSS algorithm again with the same settings (step 7). However, the outcome is to decrease the relatively safe population that much, so that the result is becoming worse. Therefore, we disable the algorithm and stop the undersampling phase (step 8). Moving on to the oversampling phase, we aim at utilizing SMOTE to generate artificial points for increasing the number of instances in the underrepresented classes. The oversampling of all data types reduces both balanced accuracy and F1-score (step 9 in Figure 1(h)). From Figure 6(f), we can understand that several problematic outliers are not considered for removal at all by the OSS algorithm during the previous phase. In particular, four outliers are predicted as vans while they belong to the car class according to the ground truth, as shown in Figure 6(g), at the bottom-left region. The oversampling algorithm should not eternalize this confusion. Consequently, we choose to exclude all six outliers from the pool of instances in order to primarily generate safe and borderline instances for the van and bus classes (*cf.* Figure 1(a)). The resulting distribution of points achieves our goal (see Figure 1(c)) and leads to an improvement in the overall predictive power (step 10).

Tracking the process and evaluating the results. To verify our sampling execution actions, we continuously monitor the process through the Sankey diagram, as shown in Figure 1(g). From this representation, we acknowledge that the population of safe instances decreased drastically when the undersampling was executed. The manual undersampling and oversampling processes (described previously) led to the best predictive result we managed to accomplish, with nine confused test instances (seven of them belonging to the car class, as presented in Figure 1(h), left). From the horizontal bar chart in Figure 1(h), right, the performance difference in each step suggests that using directly the automated sampling algorithms led to worse results (cf. steps 1 and 9). With the help of HARDVIS, we managed to improve, even more, both balanced accuracy and F1-score by approximately +2%. To sum up, our VA system guided us in systematically setting the parameters of the sampling algorithms and applying them in subsets of the data throughout the various rounds of undersampling and oversampling. As pointed out by the experts in Section 6, this would have been (almost) impossible without direct human intervention.

6. Evaluation

We performed online, semi-structured interviews with five independent experts to gain qualitative feedback on our system's usefulness, using the procedure described in prior works [MXLM20, XXM*19]. The first ML expert (**E1**) is a full professor with a PhD in computer science. He has 15 years of experience with ML, and he is head of the natural language processing (NLP) group at his university. The second ML expert (**E2**) is a full professor in ML and data science addressing mainly challenges in humanities. He has worked with ML for the past 30 years, and he holds a PhD in applied mathematics. The third ML expert (**E3**) is an assistant professor working with ML and deep learning, with 7 years of experience in ML. His PhD is in media technology. The fourth ML expert (**E4**) is a postdoc also focusing on ML and deep learning, and she has 8 years of experience in ML. Finally, the fifth ML expert (**E5**) is a post-doc with 20 years of experience in ML. The latter two experts have PhDs in computer science. **E1** was the only one who reported a colourblindness issue (deuteranomaly), but he affirmed having no problem perceiving correctly the specific colour combinations we used in HARDVIS. Each interview lasted about 1 h and 15 min, and the interviews were structured as follows: (1) introduction of the primary objectives of HARDVIS, including the analytical tasks and design goals of Section 3; (2) presentation of the functionality of every visualization and interaction with the system using the *iris flower* data set (as in Section 4) and (3) explanation of the steps taken to arrive at the results in Section 5. We asked the participants to freely comment on anything. Their responses are summarized below.

Workflow. **All experts** agreed that HARDVIS' workflow is well-designed and reasonable from their perspective. They characterized the workflow as straightforward and aligned with respective fully automated sampling processes. **E1** and **E2** repeatedly commented positively upon our systematic and fine-grained approach that they have never seen before in all those years of developing new and deploying already existent ML models. 'The offered granularity of undersampling and oversampling is exceptional, *i.e.* the fact that several phases can be applied in a row and for different subsets of the data space is something that I believe is almost impossible to accomplish without such a tool', said **E1**. **E2** underlined the clear benefit of controlling the automatic algorithms' suggestions since blindly following them could overfit the training set (and hurt generalization). He then stated that letting users be completely free to remove or generate artificial instances manually could probably harm the predictive performance similarly. Thus, **E2** found that our tool combines the best of both worlds.

Visualization and interaction. The promising findings we were able to obtain with the help of our VA system in the usage scenario of Section 5 amazed **E3** and **E4**. While using the same value for

the number of neighbours parameter and the *k*-value for the distribution of data types, **E3** appreciated that the *k*-value could still be adapted freely, as illustrated in Figure 3(c). The most intuitive visualization according to **E2**, **E4** and **E5** was the box plots view (Figure 3(f)) which was found exceptionally well-linked with the UMAP projection (Figure 3(d)). Especially with this view, **these experts** were able to understand the decisions we made in Sections 4 and 5. The inverse polar chart (*cf.* Figure 3(i)) was the most confusing view at first. However, after a careful explanation from our side, **all experts** understood its meaning and claimed this visualization was the most novel visual representation of our tool. Since the same encoding as with the UMAP projection makes this view intuitive, they were able to inspect the instances immediately with low predicted probability (and with which specific class) from the eyes of the model. An interesting suggestion by **E2** was to visualize the KNN-graph for a particular instance when users hover over a specific point/instance. Although HardVis already enables users to make justifiable actions by exploring all training instances from both global and local perspectives, this recommendation could be seen as an extra validation step for the projection. He also mentioned that for a more unsupervised-focused approach, the main colour of the projected points could show the data types, and the outline of points could be used for the ground truth labels (if there are any). Despite that, **E3** and **E4** thought that with the current colour scale, the focus is on unsafe cases, which could decrease the model's accuracy if they are removed before or without reasoning about them at all.

Limitations identified by the experts. **E1** and **E2** were concerned about the *scalability* of the system. The former concentrated on the problem of visualizing hundreds of features, while the latter on the exploration of more than three classes. **E1** acknowledged that the box plots and the table heatmap view are interactive with zooming and panning functionalities, which could partially address this issue. Also, the feature importance could be useful for deciding which features are not informative for a provided data set to exclude them beforehand. Regarding the second issue, the main bottlenecks are the inverse polar chart and the extensive use of colours. The proposed visualization could be further improved to scale with more than three classes by using advanced RadViz-based approaches [POSC*15, TBVLH*14]. Furthermore, **E2** noted that multi-class classification problems could be resolved as being binary due to the one-*versus*-rest strategy. **E5** proposed to deploy HardVis in a cloud server supporting parallel processing to improve further the *efficiency* of the system. **E1** and **E3** mentioned that heavily modifying our VA system is inevitable in case we would like to extend it to *other types of data*, *e.g.* image or NLP data sets that consist of non-interpretable features such as pixels and word vectors. However, they completely agreed that this was not our original intention. **E1** stated that non-expert users or even domain experts could find it difficult to operate HardVis and be advised by all visualizations concurrently, despite the views being logically positioned in a single window. Therefore, as an improvement of *generalizability to other target groups*, he proposed to separate the views in different tabs depending on the certain domain problem at hand and the users' prior experience to reduce the cognitive load. However, for ML experts, this deep level of granularity and the guidance received from the tool are necessary for making decisions. Finally, **E3** described that as with any other VA tool and ML model in general, the *quality of the data set* would probably affect negatively the ca-

pability of the tool to explore a complex and low-quality data set to the point that it could be challenging to improve the predictive performance. A pre-processing phase that handles missing values and wrangles the data could alleviate this problem. We plan to work on methods to surpass such limitations.

Overall assessment. The provided feedback was encouraging and in favour of HardVis compared to employing automatic approaches. **All experts** were confident about the benefits of using our VA system.

7. Discussion

In this section, we discuss the visual design and overall limitations of our approach as well as the current implementation.

7.1. Visual design

Here, we elaborate further on the key design concepts of our VA system that were presented in Section 4.

Familiarity with the prevalent types of data visualization. The visual representations used are intentionally simple but form a powerful system when combined. Specifically, the benefits originate from the identification of areas where sampling strategies should be applied with guidance across the entire process. Similar to the user profile selected for the ML experts that participated in our interview sessions, we deem that the users of our tool would have worked with box plots, bar charts, tabular representations and visual embeddings in the past. Therefore, there may be a gradual learning curve relevant to the familiarity with the visualizations. Two exceptions could be the Sankey diagram and the inverse polar chart. The former is for keeping track of their actions (usually studied under the term *provenance* in visualization [XOW*20]). A simpler alternative we considered is a log list of user's actions being registered in each step, as well as empowerements of this representation with highlighted text. However, it would capture too much space for a view that can be deemed as optional, especially since the Sankey diagram is not crucial during the exploration and analysis phases (*i.e.* before either undersampling or oversampling take place). The latter representation needs to be learned but can be a game-changer for finding instances of confusion with a particular class and observing the distribution of SBRO types from the perspective of the ML model, as already mentioned in Section 6. As a straightforward alternative, we tried out a multi-class confusion matrix. However, it only provides aggregated information and fails to use the same visual encodings as the main view (see below).

Commonality in the visual encoding and colour scales. Throughout the whole HardVis system, the visual encodings propagate from one view to the others. For example, the common grayscale denotes the four distinct types of instances in all views. Tightly connected views—such as the UMAP projection and the inverse polar chart—share identical encodings, *i.e.* label class mapped to filled-in colour, data type as outline colour and US/OS represented with symbols. The inverse polar chart is compact and uses the available space effectively due to its inherent design; it spares more area for the misclassified instances. For the table heatmap view, the

diverging colour scale emphasizes the extreme values and allows users to notice more differences on the left- and right-hand sides of the middle point, with five colours having the same origin. For example, this middle point is crucial for the *breast cancer* data set, because instances with values closer to 1 for all features should be classified as malignant, while samples with values around 0 should be benign cancer. Finally in this view, hovering over a specific cell interaction partly resolves the ambiguity problem introduced due to distributing the normalized values into 10 distinct bins.

7.2. Limitations

In the following, we acknowledge limitations we have discovered for our system (beyond those mentioned in Section 6), which imply prospective future developments.

Scalability for a large number of instances and features. In general, the number of instances and features that can be visually expressed with our approach has no intrinsic limit. Collaris and van Wijk [CVW22] found that usually the top 10–20 features were impactful for the tabular data sets they experimented with. For hundreds of features, it would be cognitively demanding for a human to analyse the influence of all these features at different levels of granularity. The methodology that might be used is first to limit the space under inspection using an additional pre-processing phase in the pipeline before employing HARDVIS for a deep analysis of features, as already stated by an ML expert in Section 6. Collaris and van Wijk [CVW22] also limited the number of instances to 5000 in order to prevent overplotting issues in their projection-based view. Arguably, similar constraints should apply to our tool, especially for the UMAP projection and the inverse polar chart view. However in our case, zooming and panning functionalities implemented for both views can partly solve this problem along with overlap removal strategies that could be helpful [HMJE*19, YXX*21]. Regarding the table heatmap view, it is mostly useful for comparing a group of instances after a lasso selection has been performed. Additionally, we have the box plots that offer an overview first and scale better to many more instances.

Other kinds of data sets. Despite the vast range of application domains covered with all our use cases, HARDVIS has merely been evaluated with structured tabular data consisting of numerical values [SZA22]. We want to enable other data types in the future. Nevertheless, the features of each data set under investigation should be meaningful, because we focus on human expertise and knowledge to resolve problematic situations where essential instances for the generalizability of unseen data are being considered for deletion and to avoid the generation of artificial samples that negatively impact the predictive performance of the model. Overall, since our prototype tool is a proof-of-concept, the system's workflow and theoretical co[ntributions are generalizable in this respect.

Target group. The primarily targeted users that would gain the most from adopting our approach are ML experts. We suppose that they understand the fundamentals of their data sets and know how to interpret common visual representations, but they require additional assistance with the sampling procedure. As evident from Section 6, the five ML experts who participated in our 1-h and 15-min interview sessions were able to grasp the main concepts and operate

Table 1: *Time taken to complete each activity of the sampling process for all use cases. The completion time is expressed in* minute:second *format. Please note that for the* iris flower *data set, the undersampling time refers to two consecutive rounds.*

Data set	Data types	Sampling process	
		Undersampling	Oversampling
Iris flower	0:45	2:57	1:06
Breast cancer	1:53	6:52	-
Vehicle silhouettes	3:29	8:58	5:12

HARDVIS. Another potential here is to create a more basic version of our tool, geared explicitly for ML developers and even inexperienced ML users with a low level of visualization literacy.

Completion time for each activity. The frontend of HARDVIS has been developed in JavaScript and uses Vue.js [vue14], D3.js [D311] and Plotly.js [plo10], while the backend has been written in Python and uses Flask [Fla10] and Scikit-learn [PVG*11]. More technical details are made available on GitHub [Har22]. All experiments were performed on a MacBook Pro 2019 with a 2.6 GHz (6-Core) Intel Core i7 CPU, an AMD Radeon Pro 5300M 4 GB GPU, 16 GB of DDR4 RAM at 2667 Mhz and running macOS Monterey. By taking into account the specifications of the computer, we recorded the total wall-clock time dedicated to completing the sampling process for each data set (see Table 1, rows). For the time reported, we aggregate both the computational analysis and the execution of the user's actions, as described in Sections 4.5 and 5. Table 1 columns map the time for each activity of the sampling process (*i.e.* distribution of data types, undersampling phase and oversampling phase). In particular, as the number of instances and features to be examined grows, so does the time necessary to compare alternative options and finalize the user-defined actions. Unsurprisingly, the undersampling phase took the longest in all situations, followed by the oversampling phase, and lastly the distribution of data types. Depending on the quantity and importance of the extracted patterns, these values might become rather different. In general, the rendering time after a major user's action is restricted to a couple of seconds for all the data sets we tried. To sum up, the efficiency of HARDVIS could be increased in various ways, as explained before.

8. Conclusion

In this paper, we developed HARDVIS, a VA system that uses hardly configurable undersampling and oversampling techniques to handle instance hardness. As part of an intensively iterative process, multiple coordinated views assist users in defining an ideal distribution of data types, undersampling particular safe for removal samples and oversampling others. Additionally, it facilitates the exploration of algorithmic suggestions using a variety of visual clues to confirm non-harmful removal or addition proposals. Finally, our VA approach is ideal for dealing with the instance hardness and class imbalance challenges because it makes the entire process adjustable and more transparent. The effectiveness of HARDVIS was investigated using real-world data sets, which revealed an increase of trustworthiness and in performance due to removed and synthetically generated instances. The workflow and visualizations

of our system received positive feedback from experts suggesting that such in-depth sampling would be impossible without our tool. They also assisted us in identifying the existing limitations of HARDVIS, which we are considering as future research directions.

Acknowledgement

References

[ACD*15] AMERSHI S., CHICKERING M., DRUCKER S. M., LEE B., SIMARD P., SUH J.: ModelTracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 337–346. https://doi.org/10.1145/2702123.2702509

[AHH*14] ALSALLAKH B., HANBURY A., HAUSER H., MIKSCH S., RAUBER A.: Visual methods for analyzing probabilistic classification data. *IEEE* Transactions on Visualization and Computer Graphics 20, 12 (2014), 1703–1712. https://doi.org/10.1109/TVCG.2014.2346660

[Alt92] ALTMAN N. S.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician 46, 3* (1992), 175–185.

[AY19] ARAKAWA R., YAKURA H.: REsCUE: A framework for real-time feedback on behavioral cues using multimodal anomaly detection. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), ACM, pp. 1–13. https://doi.org/10.1145/3290605.3300802

[BHS*21] BERNARD J., HUTTER M., SEDLMAIR M., ZEPPELZAUER M., MUNZNER T.: A taxonomy of property measures to unify active learning and human-centered approaches to data labeling. *ACM Transactions on Interactive Intelligent Systems 11, 3–4* (Sep. 2021). https://doi.org/10.1145/3439333

[BHZ*18] BERNARD J., HUTTER M., ZEPPELZAUER M., FELLNER D., SEDLMAIR M.: Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics 24, 1* (2018), 298–308. https://doi.org/10.1109/TVCG.2017.2744818

[BKNS00] BREUNIG M. M., KRIEGEL H.-P., NG R. T., SANDER J.: LOF: Identifying density-based local outliers. *ACM SIGMOD Record 29, 2* (May 2000), 93–104. https://doi.org/10.1145/335191.335388

[BNR20] BÄUERLE A., NEUMANN H., ROPINSKI T.: Classifier-guided visual correction of noisy labels for image classification tasks. *Computer Graphics Forum 39, 3* (2020), 195–205. https://doi.org/10.1111/cgf.13973

[BS03] BAY S. D., SCHWABACHER M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), ACM, pp. 29–38. https://doi.org/10.1145/956750.956758

[BSL09] BUNKHUMPORNPAT C., SINAPIROMSARAN K., LURSINSAP C.: Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-sampling TEchnique for handling the class imbalanced problem. In *Proceedings of the Advances in Knowledge Discovery and Data Mining* (2009), Springer, Berlin Heidelberg, pp. 475–482.

[BZL*18] BERNARD J., ZEPPELZAUER M., LEHMANN M., MÜLLER M., SEDLMAIR M.: Towards user-centered active learning algorithms. *Computer Graphics Forum 37, 3* (2018), 121–132. https://doi.org/10.1111/cgf.13406

[CBHK02] CHAWLA N. V., BOWYER K. W., HALL L. O., KEGELMEYER W. P.: SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16, 1* (June 2002), 321–357.

[CBK09] CHANDOLA V., BANERJEE A., KUMAR V.: Anomaly detection: A survey. *ACM Computing Surveys 41, 3* (July 2009). https://doi.org/10.1145/1541880.1541882

[CCS06] CIESLAK D., CHAWLA N., STRIEGEL A.: Combating imbalance in network intrusion datasets. In *Proceedings of the IEEE International Conference on Granular Computing* (2006), 732–737. https://doi.org/10.1109/GRC.2006.1635905

[CdMP14] CASTOR DE MELO C. E., PRUDENCIO R. B. C.: Cost-sensitive measures of algorithm similarity for meta-learning. In *Proceedings of the 2014 Brazilian Conference on Intelligent Systems* (2014), 7–12. https://doi.org/10.1109/BRACIS.2014.13

[CG16] CHEN T., GUESTRIN C.: XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[CLGD18] CAO N., LIN, Y.-R., GOTZ D., DU F.: Z-Glyph: Visualizing outliers in multivariate data. *Information Visualization 17, 1* (2018), 22–40. https://doi.org/10.1177/1473871616686635

[CMJ*20] CHATZIMPARMPAS A., MARTINS R. M., JUSUFI I., KUCHER K., ROSSI F., KERREN A.: The state of the art in enhancing trust in machine learning models with the use of visualizations. *Computer Graphics Forum 39, 3* (June 2020), 713–756. https://doi.org/10.1111/cgf.14034

[CMJK20] CHATZIMPARMPAS A., MARTINS R. M., JUSUFI I., KERREN A.: A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization 19, 3* (July 2020), 207–233. https://doi.org/10.1177/1473871620904671

[CMK20] CHATZIMPARMPAS A., MARTINS R. M., KERREN A.: t-viSNE: Interactive assessment and interpretation of t-SNE projections. *IEEE Transactions on Visualization and Computer Graphics 26, 8* (Aug. 2020), 2696–2714. https://doi.org/10.1109/TVCG.2020.2986996

[CMK23] CHATZIMPARMPAS A., MARTINS R. M., KERREN A.: Vis-Ruler: Visual analytics for extracting decision rules from bagged and boosted decision trees. *Information Visualization* (2023). To appear.

[CMKK21a] CHATZIMPARMPAS A., MARTINS R. M., KUCHER K., KERREN A.: StackGenVis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (Feb. 2021), 1547–1557. https://doi.org/10.1109/TVCG.2020.3030352

[CMKK21b] CHATZIMPARMPAS A., MARTINS R. M., KUCHER K., KERREN A.: VisEvol: Visual analytics to support hyperparameter search through evolutionary optimization. *Computer Graphics Forum* 40, 3 (June 2021), 201–214. https://doi.org/10.1111/cgf.14300

[CMKK22] CHATZIMPARMPAS A., MARTINS R. M., KUCHER K., KERREN A.: FeatureEnVi: Visual analytics for feature engineering using stepwise selection and semi-automatic extraction approaches. *IEEE Transactions on Visualization and Computer Graphics* 28, 4 (2022), 1773–1791. https://doi.org/10.1109/TVCG.2022.3141040

[CSL*16] CAO N., SHI C., LIN S., LU J., LIN, Y.-R., LIN, C.-Y.,: TargetVue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 280–289. https://doi.org/10.1109/TVCG.2015.2467196

[CT17] CZARNECKI W. M., TABOR J.: Extreme entropy machines: Robust information theoretic classification. *Pattern Analysis and Applications* 20, 2 (2017), 383–400.

[CVW22] COLLARIS D., VAN WIJK J.: StrategyAtlas: Strategy analysis for machine learning interpretability. *IEEE Transactions on Visualization and Computer Graphics* (2022). https://doi.org/10.1109/TVCG.2022.3146806

[CZV13] CANO A., ZAFRA A., VENTURA S.: Weighted data gravitation classification for standard and imbalanced data. *IEEE Transactions on Cybernetics* 43, 6 (2013), 1672–1687. https://doi.org/10.1109/TSMCB.2012.2227470

[D311] D3—Data-driven documents (2011). https://d3js.org/. Accessed December 20, 2022.

[DB21] DENG J., BROWN, E. T.: RISSAD: Rule-based interactive semi-supervised anomaly detection. In *Proceedings of the EuroVis—Short Papers* (2021), The Eurographics Association. https://doi.org/10.2312/evs.20211050

[DG17] DUA D., GRAFF C.: UCI machine learning repository (2017). https://archive.ics.uci.edu/ml. Accessed December 20, 2022.

[EMK*21] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA, N. S. T., TELEA, A. C.: Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization*

and Computer Graphics 27, 3 (2021), 2153–2173. https://doi.org/10.1109/TVCG.2019.2944182

[FH89] FIX E., HODGES J. L.: Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique* 57, 3 (1989), 238–247.

[FIS36] FISHER, R. A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 2 (1936), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[Fla10] Flask—A micro web framework written in Python (2010). https://flask.palletsprojects.com/. Accessed December 20, 2022.

[FSA99] FREUND Y., SCHAPIRE R., ABE N.: A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence* 14, 5 (Sep. 1999), 771–780.

[GBSW21] GROSSMANN N., BERNARD J., SEDLMAIR M., WALDNER M.: Does the layout really matter? A study on visual model accuracy estimation. In *Proceedings of the IEEE Visualization Conference (VIS)* (2021), 61–65. https://doi.org/10.1109/VIS49827.2021.9623326

[GSM07] GARCÍA V., SÁNCHEZ J., MOLLINEDA R.: An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Proceedings of the Progress in Pattern Recognition, Image Analysis and Applications* (2007), Springer-Verlag, pp. 397–406.

[HA04] HODGE V., AUSTIN J.: A survey of outlier detection methodologies. *Artificial Intelligence Review* 22, 2 (2004), 85–126.

[Ham22] HAMID, O. H.: From model-centric to data-centric AI: A paradigm shift or rather a complementary approach? In *Proceedings of the 8th International Conference on Information Technology Trends (ITT)* (2022), 196–199. https://doi.org/10.1109/ITT56123.2022.9863935

[Har68] HART P.: The condensed nearest neighbor rule (corresp.). *IEEE Transactions on Information Theory* 14, 3 (1968), 515–516. https://doi.org/10.1109/TIT.1968.1054155

[Har22] HardVis code (2022). https://github.com/angeloschatzimparmpas/HardVis. Accessed December 20, 2022.

[HBGL08] HE H., BAI Y., GARCIA E. A., LI S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (2008), 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[HCG*14] HUANG H., CHIEW K., GAO Y., HE Q., LI Q.: Rare category exploration. *Expert Systems with Applications* 41, 9 (2014), 4197–4210. https://doi.org/10.1016/j.eswa.2013.12.039

[HG09] HE H., GARCIA, E. A.: Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[HHHM11] HUANG H., HE Q., HE J., MA L.: RADAR: Rare category detection via computation of boundary degree. In *Proceedings of the Advances in Knowledge Discovery and Data Mining* (2011), Springer, Berlin Heidelberg, pp. 258–269.

[HHWB02] HAWKINS S., HE H., WILLIAMS G., BAXTER R.: Outlier detection using replicator neural networks. In *Proceedings of the Data Warehousing and Knowledge Discovery* (2002), Springer, Berlin Heidelberg, pp. 170–180.

[HKB18] HERLAND M., KHOSHGOFTAAR T. M., BAUDER R. A.: Big data fraud detection using multiple medicare data sources. *Journal of Big Data 5*, 1 (2018), 1–21.

[HLL08] HE J., LIU Y., LAWRENCE R.: Graph-based rare category detection. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (2008), 833–838. https://doi.org/10.1109/ICDM.2008.122

[HM13] HE H., MA Y.: *Imbalanced Learning: Foundations, Algorithms, and Applications*. John Wiley Sons, Hoboken (2013).

[HMJE*19] HILASACA G. M., MARCÍLIO- JR W. E., ELER D. M., MARTINS R. M., PAULOVICH, F. V.: Overlap removal of dimensionality reduction scatterplot layouts. *ArXiv e-prints 1903.06262* (2019).https://arxiv.org/abs/1903.06262

[HWM05] HAN H., WANG W.-Y., MAO B.-H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the Advances in Intelligent Computing* (2005), Springer, Berlin, Heidelberg, pp. 878–887.

[Jap01] JAPKOWICZ N.: Concept-learning in the presence of between-class and within-class imbalances. In *Proceedings of the Advances in Artificial Intelligence* (2001), Springer, Berlin, Heidelberg, pp. 67–77.

[KGW17] KSIENIEWICZ P., GRANA M., WOŹNIAK M.: Paired feature multilayer ensemble—concept and evaluation of a classifier. *Journal of Intelligent & Fuzzy Systems 32*, 2 (2017), 1427–1436.

[KHM98] KUBAT M., HOLTE R. C., MATWIN S.: Machine learning for the detection of oil spills in satellite radar images. *Machine Learning 30*, 2 (1998), 195–215.

[KK17] KWAK S. K., KIM J. H.: Statistical data preparation: Management of missing values and outliers. *Korean Journal of Anesthesiology 70*, 4 (2017), 407–411.

[KM97] KUBAT M., MATWIN S.: Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the International Conference on Machine Learning (ICML)* (1997), Morgan Kaufmann, pp. 179–186.

[Kra16] KRAWCZYK B.: Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence 5*, 4 (2016), 221–232.

[Kru64] KRUSKAL, J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (Mar. 1964), 1–27. https://doi.org/10.1007/BF02289565

[Lau01] LAURIKKALA J.: Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the AIME* (2001), Springer-Verlag, pp. 63–66.

[LCH*14] LIU Z., CHIEW K., HE Q., HUANG H., HUANG B.: Prior-free rare category detection: More effective and efficient solutions. *Expert Systems with Applications 41*, 17 (Dec. 2014), 7691–7706. https://doi.org/10.1016/j.eswa.2014.06.026

[LFM*18] LI H., FANG S., MUKHOPADHYAY S., SAYKIN A. J., SHEN L.: Interactive machine learning by visualization: A small data solution. In *Proceedings of the IEEE BigData* (2018), 3513–3521. https://doi.org/10.1109/BigData.2018.8621952

[LGG*18] LIN H., GAO S., GOTZ D., DU F., HE J., CAO N.: RCLens: Interactive rare category exploration and identification. *IEEE Transactions on Visualization and Computer Graphics 24*, 7 (2018), 2223–2237. https://doi.org/10.1109/TVCG.2017.2711030

[MC03] MAHONEY M., CHAN P.: Learning rules for anomaly detection of hostile network traffic. In *Proceedings of the IEEE ICDM* (2003), 601–604. https://doi.org/10.1109/ICDM.2003.1250987

[Mel02] MELNIK O.: Decision region connectivity analysis: A method for analyzing high-dimensional classifiers. *Machine Learning 48*, 1–3 (Sep. 2002), 321–351. https://doi.org/10.1023/A:1013968124284

[MHM18] MCINNES L., HEALY J., MELVILLE J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *ArXiv e-prints 1802.03426* (Feb. 2018). https://arxiv.org/abs/1802.03426

[MLC07] MÜNZ G., LI S., CARLE G.: Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet* (2007), vol. 7, pp. 9.

[MM21] MA Y., MACIEJEWSKI R.: Visual analysis of class separations with locally linear segments. *IEEE Transactions on Visualization and Computer Graphics 27*, 1 (2021), 241–253. https://doi.org/10.1109/TVCG.2020.3011155

[MQB19] MING Y., QU H., BERTINI E.: RuleMatrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 342–352. https://doi.org/10.1109/TVCG.2018.2864812

[MXLM20] MA Y., XIE T., LI J., MACIEJEWSKI R.: Explaining vulnerabilities to adversarial machine learning through visual analytics. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (Jan. 2020), 10751085. https://doi.org/10:1109/TVCG:2019:2934631

[MS94] MARTINETZ T., SCHULTEN K.: Topology representing networks. *Neural Networks 7*, 3 (1994), 507–522. https://doi.org/10.1016/0893-6080(94)90109-0

[Nog14] NOGUEIRA F.: Bayesian Optimization: Open source constrained global optimization tool for Python (2014). https://github.com/fmfn/BayesianOptimization. Accessed December 20, 2022.

[NS16] NAPIERALA K., STEFANOWSKI J.: Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems 46*, 3 (2016), 563–597.

[NSW10] NAPIERAŁA K., STEFANOWSKI J., WILK S.: Learning from imbalanced data in presence of noisy and borderline examples. In *Proceedings of the Rough Sets and Current Trends in Computing* (2010), Springer, Berlin, Heidelberg, pp. 158–167.

[PBM04] PRATI R. C., BATISTA G. E. A. P. A., MONARD M. C.: Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Proceedings of the MICAI 2004: Advances in Artificial Intelligence* (2004), Springer, Berlin, Heidelberg, pp. 312–321.

[PHOMU15] PRUDÊNCIO R. B., HERNÁNDEZ-ORALLO J., MARTINEZ-USÓ A.: Analysis of instance hardness in machine learning using item response theory. In *Proceedings of the International Workshop on Learning over Multiple Contexts* (2015).

[plo10] Plotly—JavaScript open source graphing library (2010). https://plot.ly. Accessed December 20, 2022.

[POSC*15] PIAZENTIN ONO J. H., SIKANSI F., CORRÊA D. C., PAULOVICH F. V., PAIVA A., NONATO, L. G.: Concentric RadViz: Visual exploration of multi-task classification. In *Proceedings of the 28th SIBGRAPI Conference on Graphics, Patterns and Images* (2015), 165–172. https://doi.org/10.1109/SIBGRAPI.2015.38

[PSPM15] PAIVA, J. G. S., SCHWARTZ W. R., PEDRINI H., MINGHIM R.: An approach to supporting incremental visual data classification. *IEEE Transactions on Visualization and Computer Graphics 21*, 1 (2015), 4–17. https://doi.org/10.1109/TVCG.2014.2331979

[PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-Learn: Machine learning in Python. *Journal of Machine Learning Research 12* (Nov. 2011), 2825–2830. https://doi.org/10.5555/1953048.2078195

[RAL*17] REN D., AMERSHI S., LEE B., SUH J., WILLIAMS, J. D.: Squares: Supporting interactive performance analysis for multi-class classifiers. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan. 2017), 61–70. https://doi.org/10.1109/TVCG.2016.2598828

[RAS*21] ROSTAMZADEH N., ABDULLAH S. S., SEDIG K., GARG A. X., MCARTHUR E.: VERONICA: Visual analytics for identifying feature groups in disease classification. *Information 12*, 9 (2021). https://doi.org/10.3390/info12090344

[Rav11] RAVINDRAN S.: Learning with imprecise classes, rare instances, and complex relationships. In *Proceedings of the AAAI/SIGART Doctoral Consortium* (2011).

[RKN06] RAO R. B., KRISHNAN S., NICULESCU, R. S.: Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter 8*, 1 (June 2006), 3–10. https://doi.org/10.1145/1147234.1147236

[RVM19] RAMAMURTHY K. N., VARSHNEY K., MODY K.: Topological data analysis of decision boundaries with application to model selection. In *Proceedings of the International Conference on Machine Learning (ICML)* (June 2019), vol. *97*, pp. 5351–5360. PMLR

[RVV*15] RAMENTOL E., VLUYMANS S., VERBIEST N., CABALLERO Y., BELLO R., CORNELIS C., HERRERA F.: IFROWANN: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. *IEEE Transactions on Fuzzy Systems 23*, 5 (2015), 1622–1637. https://doi.org/10.1109/TFUZZ.2014.2371472

[SAPV16] SALGADO C. M., AZEVEDO C., Proença H., VIEIRA, S. M.: *Noise versus Outliers*. Cham, Switzerland: Springer, 2016, pp. 163–183.https://doi.org/10.1007/978-3-319-43742-2_14

[Set12] SETTLES B.: Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 6*, 1 (2012), 1–114.

[SHB*14] SEDLMAIR M., HEINZL C., BRUCKNER S., PIRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2161–2170. https://doi.org/10.1109/TVCG.2014.2346321

[Sie87] SIEBERT, J. P.: Vehicle Recognition Using Rule Based Methods. Research Memorandum TIRM-87-018, Turing Institute, Mar. 1987.

[SK17] SKRYJOMSKI P., KRAWCZYK B.: Influence of minority class instance types on smote imbalanced data oversampling. In *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications* (Sep. 2017), vol. *74*, PMLR, pp. 7–21.

[SLSH15] SÁ ez J. A., LUENGO J., STEFANOWSKI J., HERRERA F.: SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences 291* (2015), 184–203. https://doi.org/10.1016/j.ins.2014.08.051

[SMGC14] SMITH M. R., MARTINEZ T., GIRAUD-CARRIER C.: An instance level analysis of data complexity. *Machine Learning 95*, 2 (2014), 225–256.

[SPBW12] SYARIF I., PRUGEL-BENNETT A., WILLS G.: Unsupervised clustering approach for network anomaly detection. In *Proceedings of the Networked Digital Technologies* (2012), Springer, Berlin, Heidelberg, pp. 135–145.

[Ste16] STEFANOWSKI J.: Dealing with data difficulty factors while learning from imbalanced data. In *Challenges in Computational Statistics and Data Mining*. Springer, Cham (2016), 333–363.

[SZA22] SHWARTZ-ZIV R., ARMON A.: Tabular data: Deep learning is not all you need. *Information Fusion 81* (2022), 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

[TBVLH*14] THANH BINH H. T., VAN LONG T., HOAI N. X., ANH N. D., TRUONG, P. M.: Reordering dimensions for radial visualization of multidimensional data—a genetic algorithms approach. In *Proceedings of the IEEE Congress on Evolutionary Computation* (2014), 951–958. https://doi.org/10.1109/CEC.2014.6900619

[Tom76] TOMEK I.: An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics SMC-6*, 6 (1976), 448–452. https://doi.org/10.1109/TSMC.1976.4309523

[VC17] VANERIO J., CASAS P.: Ensemble-learning approaches for network security and anomaly detection. In *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks* (2017), ACM, pp. 1–6. https://doi.org/10.1145/3098593.3098594

[VK09] VAN HULSE J., KHOSHGOFTAAR T.: Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering 68*, 12 (2009), 1513–1542. https://doi.org/10.1016/j.datak.2009.08.005

[vue14] Vue.js—The progressive JavaScript framework (2014). https://vuejs.org/. Accessed December 20, 2022.

[VW09] VATTURI P., WONG, W.-K.,: Category detection using hierarchical mean shift. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), ACM, pp. 847–856. https://doi.org/10.1145/1557019.1557112

[WDC*22] WU A., DENG D., CHENG F., WU Y., LIU S., QU H.: In defence of visual analytics systems: Replies to critics. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–11. https://doi.org/10.1109/TVCG.2022.3209360

[WGC14] WOŹNIAK M., GRAÑA M., CORCHADO E.: A survey of multiple classifier systems as hybrid systems. *Information Fusion 16* (2014), 3–17. https://doi.org/10.1016/j.inffus.2013.04.006

[WH00] WEISS G. M., HIRSH H.: A quantitative study of small disjuncts. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (2000), AAAI Press. pp. 665–670.

[Wil72] WILSON, D. L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics SMC-2*, 3 (1972), 408–421. https://doi.org/10.1109/TSMC.1972.4309137

[WLC*13] WEI W., LI J., CAO L., OU Y., CHEN J.: Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web 16*, 4 (2013), 449–475.

[WMCW03] WONG W.-K., MOORE A. W., COOPER G. F., WAGNER M. M.: Bayesian network anomaly pattern detection for disease outbreaks. In *Proceedings of the International Conference on Machine Learning (ICML)* (2003), AAAI Press. pp. 808–815.

[XOW*20] XU K., OTTLEY A., WALCHSHOFER C., STREIT M., CHANG R., WENSKOVITCH J.: Survey on the analysis of user interactions and visualization provenance. *Computer Graphics Forum 39*, 3 (2020), 757–783. https://doi.org/10.1111/cgf.14035

[XXM*19] XU K., XIA M., MU X., WANG Y., CAO N.: Ensemble-Lens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (Jan. 2019), 109–119. https://doi.org/10.1109/TVCG.2018.2864825

[XYX*19] XIANG S., YE X., XIA J., WU J., CHEN Y., LIU S.: Interactive correction of mislabeled training data. In *Proceedings of the 2019 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2019), 57–68. https://doi.org/10.1109/VAST47406.2019.8986943

[YCY*21] YUAN J., CHEN C., YANG W., LIU M., XIA J., LIU S.: A survey of visual analytics techniques for machine learning. *Computational Visual Media 7*, 1 (2021), 3–36. https://doi.org/10.1007/s41095-020-0191-7

[YLF*21] YU S., LI X., FENG Y., ZHANG X., CHEN S.: An instance-oriented performance measure for classification. *Information Sciences 580* (2021), 598–619. https://doi.org/10.1016/j.ins.2021.08.094

[YLW*21] YU S., LI X., WANG H., ZHANG X., CHEN S.: BIDI: A classification algorithm with instance difficulty invariance. *Expert Systems with Applications 165* (2021), 113920. https://doi.org/10.1016/j.eswa.2020.113920

[YTWM04] YAMANISHI K., TAKEUCHI J.-I., WILLIAMS G., MILNE P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery 8*, 3 (2004), 275–300.

[YXX*21] YUAN J., XIANG S., XIA J., YU L., LIU S.: Evaluation of sampling methods for scatterplots. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1720–1730. https://doi.org/10.1109/TVCG.2020.3030432

[ZCW*14] ZHAO J., CAO N., WEN Z., SONG Y., LIN, Y.-R., COLLINS C.: #FluxFlow: Visual analysis of Transactions on Visualization and Computer Graphics information spreading on social media. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1773–1782. https://doi.org/10.1109/TVCG.2014.2346922

[ZCW*19] ZHAO X., CUI W., WU Y., ZHANG H., QU H., ZHANG D.: Oui! Outlier interpretation on multi-dimensional data via visual analytics. *Computer Graphics Forum 38*, 3 (June 2019), 213–224. https://doi.org/10.1111/cgf.13683

[ZDH*17] Zhang X., Dou W., He Q., Zhou R., Leckie C., Kotagiri R., Salcic Z.: LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis. In *Proceedings of the IEEE 33rd International Conference on Data Engineering* (2017), 983–994. https://doi.org/10.1109/ICDE.2017.145

[ZOS*23] Zhang X., Ono J. P., Song H., Gou L., Ma K.-L., Ren L.: SliceTeller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics* (2023), *29*, 1, 842–852. https://doi.org/10.1109/TVCG.2022.3209465

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Data S1