IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 February 2024, accepted 19 March 2024, date of publication 25 March 2024, date of current version 1 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381536

RESEARCH ARTICLE

Effects of Class Imbalance Countermeasures on Interpretability

DAVID CEMERNEK[1], SHAFAQ SIDDIQI[2], AND ROMAN KERN[2]

[1]Know-Center GmbH, 8010 Graz, Austria
[2]Faculty of Computer Science and Biomedical Engineering, Institute of Interactive Systems and Data Science, Graz University of Technology (TU Graz), 8010 Graz, Austria

Corresponding author: Shafaq Siddiqi (shafaq.siddiqi@tugraz.at)

Know Center is funded within the Austrian COMET Program—Competence Centers for Excellent Technologies—under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Austrian Federal Ministry for Digital and Economic Affairs (BMDW) and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

ABSTRACT The widespread use of artificial intelligence (AI) in more and more real-world applications is accompanied by challenges that are not obvious at first glance. In machine learning, class imbalance, characterized by an imbalance in the frequency of classes, is one key challenge that poses essential problems for many common machine learning algorithms. This challenge led to the development of various countermeasures to tackle class imbalance. Although these countermeasures improve the prediction performance of models, they often jeopardize interpretability for both AI users and AI experts. Especially in sensitive domains where class imbalance is regularly present, for example, medicine, meteorology, or fraud detection, interpretability is of utmost importance. In this paper, we evaluate the effect of class imbalance countermeasures on interpretability with methods of explainable AI (XAI). Our work contributes to a more in-depth understanding of these countermeasures and connects the research fields of class imbalance learning and XAI. Our experimental results suggest that only feature selection and cost-sensitive approaches are the only class imbalance countermeasures that preserve interpretability for both AI users and AI experts. In contrast, resampling and most classification algorithms for imbalance learning are not suitable in settings where knowledge should be derived and where interpretability is a key requirement.

INDEX TERMS Classification, class imbalance, explainable AI, interpretability.

I. INTRODUCTION

While AI is improving processes in many areas, it is accompanied by challenges that need to be embraced. With AI, or more precisely, machine learning (ML), the quality in the modern manufacturing industry is constantly improving [1]. This improvement leads to fewer and fewer defects, which consequently reduces the number of data instances describing such defects. This lack of training data affects the ability of models to detect these defects, reducing the quality of defect detection of these models. Many real-world tasks, such as detecting fraudulent credit card transactions, predicting rare weather conditions, or using ML to treat patients with rare diseases, have similar situations where relevant data instances are rare [2].

In ML, situations where rare instances represent the minority and normal instances represent the majority are referred to as class imbalance. Over the past few decades, the research field of class imbalance learning has focused on the detection and modeling of minority instances, or "rare events". Many successful approaches have been developed [1], often referred to as class imbalance countermeasures. These countermeasures incorporate specific techniques that focus on rare events to increase the overall accuracy, for example, re-sampling with the Synthetic Minority Over-Sampling Technique (SMOTE) [3]. Most of these countermeasures are typically applied purely based on the existence of class imbalance in the training data. Class imbalance is only a concern if a model is incapable of distinguishing the majority from the minority class [4]. Furthermore, certain countermeasures require distinct pre- or post-processing, adjusting of their inherent (hyper-)parameters, and in some instances, multiple countermeasures are required [5]. Countermeasures increase the complexity of the model selection process, compromising its traceability. The interpretability

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

of the countermeasures themselves being a secondary factor, the overall interpretability of the resulting model is jeopardized [6].

This lack of interpretability contradicts the typical information needs of humans, for example, a customer wants to understand why his credit card payment was declined; a meteorologist wishes to understand why extreme weather phenomena are predicted; doctors and patients desire to understand why a specific treatment is recommended; and finally, a manufacturer intends to understand why a model predicts that a specific product does not fulfill the quality criteria. Similar to AI and ML in general, acceptance of class imbalance countermeasures depends on the trust of humans, which is only established if concerns about fairness, privacy, reliability, and robustness are overcome [7].

Interpretability has the potential to eliminate these concerns to better understand, and therefore trust these models. This motivates us to investigate the interpretability of the most common class imbalance countermeasures and their representatives. We therefore analyze these countermeasures and their effect on model interpretability. Currently, the evaluations, reviews, and surveys in class imbalance learning focus either on specific domains only or only on one specific type of countermeasure, for example, re-sampling. Recent publications often apply heavy-tailed evaluations using complex pipelines, comparing thousands of different combinations of various (hyper-)parameters and pipeline steps with a focus on the pure accuracy of involved approaches.

On the contrary, the focus of our evaluation relies on an interactive and visual analysis of current class imbalance countermeasures. Therefore, we use a self-developed, bespoke web application that incorporates methods of explainable AI (XAI) to compare their effects on interpretability. One can choose from 27 data sets incorporating class imbalance. Once a data set is chosen, the user of the web app can choose and combine several class imbalance countermeasures and different classification models. After the model is trained with the given countermeasures, we provide a comparison of the performance of the ML models with and without class imbalance countermeasures. In the earlier stages of our investigation, we discovered that interpretability has different implications from the perspectives of AI users and AI experts. We will elaborate the key differences of these perspectives in our background section, and will also provide separate conclusions for these two different perspectives in our results & discussion section. Overall, our findings suggest that only cost-sensitive methods and feature selection fulfill the requirement of interpretability from both perspectives. To provide full traceability and to aid the reproducibility of our results, the datasets, and our web application are publicly available.

This work has the following contributions to the current state-of-the-art:

- An introduction to class imbalance, its characteristics, challenges, and state-of-the-art countermeasures.
- To the best of our knowledge, the first systematic union of class imbalance countermeasures with XAI.
- A clear definition of interpretability concerning the perspectives of both AI users and AI experts.
- An overview of the most common class imbalance countermeasures and their interpretability.
- A publicly available web application, including its source code, and datasets.

II. CLASS IMBALANCE COUNTERMEASURES AND INTERPRETABILITY

A. CLASS IMBALANCE

In classification, a model is trained on a given training set to learn a mapping between its input and output data. The training set consists of instances, where each instance is represented by its feature values (input or independent variable) and a concept or class label (output or dependent variable). After training, the model should be able to predict the correct class value by only being presented with the instance's feature values. Suppose the number of instances for one class, namely the majority class, outnumbers the number of instances for the other, namely the minority class. In that case, it is referred to as skewed, biased, imbalanced data or *class imbalance*. If instances have only one label with two possible values it is referred to as binary classification, which is the focus of this work. If there are three or more distinct labels, it is multi-class classification.

Since this task has already been studied for quite some time, it has already received various names. Beginning with small or rare class learning problems [1], learning from imbalanced data [8], learning from class-imbalanced data [2], class imbalance learning, and class imbalance problem [9]. In the real world, numerous examples of imbalanced data exist, for example, fraudulent credit card transaction data where the representation of legitimate transaction instances is expected to be greater than the number of fraudulent transactions. In this case, the data is generated with an uneven distribution and a naive classification model will classify the unseen fraudulent instances as non-fraudulent, resulting in high accuracy but low performance [10]. One important aspect of imbalanced data is the ratio between the number of majority instance and the number of minority instances, referred to as class imbalance ratio (CIR). The CIR typically ranges from 1:4 to 1:100, but can also range from 1:1000 up to 1:5000 or even worse, in real-world applications such as fraud detection or chemo-informatics [8]. The global CIR is often used as the sole and most "striking" criterion for describing the existence of class imbalance in a data set.

Nevertheless, the pure existence of class imbalance does not indicate whether a model can separate the majority from the minority class [4]. Next to the CIR, [1] identified and analyzed the *small sample size*, the *class separability*, and *within-class concepts* as aspects that influence the capability of a model to identify the minority class. According to

TABLE 1. Some fictitious predictions of an arbitrary classifier assembled in a confusion matrix.

		Predicted	
		+	-
Actual	+	950 (TP)	0 (FN)
	-	45 (FP)	5 (TN)

the authors, the small sample size makes it challenging to uncover regularities inherent in the minority class. Furthermore, patterns among each class are overlapping in some feature spaces, hindering the model from inducing rules to discriminate the minority from the majority class. "Within-class concepts" is the term used to describe classes that are composed of implicit subclasses, which additionally increase the learning complexity and even worsen the effect of class imbalance. So far, we have discussed what class imbalance is, how it emerges, and what are the characteristics of the minority and majority instances. In the following section, we focus on the challenges of common classifiers when dealing with class imbalance.

B. CHALLENGES POSED BY CLASS IMBALANCE

A desirable objective for a model is its ability to generalize, which is the capability of correctly predicting the class values of "previously unknown" instances. These unknown instances follow the underlying generative nature, but fall outside the observed instances during training. To achieve this generalization, the model first has to fit the training data accordingly. If the model fails to fit the training data, it is under-fitting, and consequently will not be able to generalize. If the model fits the training data too well, it is over-fitting, meaning it captures too many details statistically independent of the target class of the training data, and will most likely, again, fail to generalize [11]. To avoid over-fitting, techniques that search for the most common regularities are introduced to ensure generalization. Such techniques generate maximum-generality bias, favoring the discovery of more general rules. This inductive bias of maximum-generality poses a serious obstacle for the classification of imbalanced data, since rules that predict the minority class are often scarce, undiscovered, or overlooked, resulting in a weak generalization performance of such models [1]. According to [12] the problem with standard learning algorithms and imbalanced data is that the rules describing the minority class are often fewer and weaker than those of the majority class.

For the evaluation of a model in the presence of class imbalance, an evaluation metric that accurately represents class imbalance is required [6], [9], [13], [14], [15]. Common evaluation metrics, such as accuracy, precision, recall, and the F1- score strongly vary when swapping classes. An example, consisting of a confusion matrix of fictitious predictions of an arbitrary classifier, and common evaluation metrics from a recent work by [16], can be found in Table 1 and 2. To illustrate the need for proper evaluation metrics, we added the result for the Matthew's Correlation Coefficient (MCC),

TABLE 2. Common evaluation metrics and MCC for the fictitious predictions of Table 1 rounded to three decimal places.

Metric	Result
Accuracy	0.955
Precision	0.955
Recall	1.000
F1	0.977
MCC	0.309

TABLE 3. Common evaluation metrics and MCC when the labels are swapped in Table 1 rounded to three decimal places.

Metric	Result
Accuracy	0.955
Precision	1.000
Recall	0.100
F1	0.181
MCC	0.309

an evaluation metric invariant of class swapping [13]. In this fictitious case, we have a total of 1,000 instances, 950 for the majority class (labeled as positive) and 50 for the minority class (labeled as negative). Although the classification fails to predict the overwhelming number of the minority class, there are 45 false positives out of a total of 50. All evaluation metrics, except MCC, fail to represent the poor performance of the classification for the minority class. To depict the class-swapping-invariant characteristic of MCC, we re-computed all the measures in Table 2 by swapping the class labels of majority and minority class in Table 3. Reference [15] concluded that if the focus is on pure successes (dismissing the errors), bookmaker informedness and geometric mean are the best evaluation metrics. If classification errors must also be considered, then the MCC is the preferred option.

In addition to MCC, the Polygon Area Metric (PAM) [17] is a state-of-the-art evaluation metric. PAM avoids the need to create large tables in the presence of multiple evaluation metrics. PAM provides one single scalar value as an estimate of a classifier's ability by considering all the aspects of a classifier's performance, i.e., specificity, sensitivity, AUC, Kappa, F-measure, and classifier accuracy. Despite the promising performance of PAM, its applications are limited when data is imbalanced [18] and results are difficult to interpret as one of its shortcomings mentioned by the authors is that "unlike confusion matrix, it does not provide information about exact values of TP, TN, FP, and FN." Reference [17], which are useful in our use case to understand the behavior of a model. As we are primarily interested in a single scalar, which is invariant of class swapping, as an estimation of classification errors and confusion matrix, we will solely utilize the MCC score as the evaluation metric in our experiments. Therefore, we shall briefly introduce it. The MCC can take values in the interval $[-1 \dots 1]$, whereas a value of -1 represents complete disagreement, 0 means that the prediction is uncorrelated with the ground truth, and 1 means complete agreement. From the equation of the MCC depicted in Equation 1 one can see that in case any of the expressions in the denominator are zero,

then the MCC is not defined [19].

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \quad (1)$$

In general, caution should be exercised not only concerning an appropriate evaluation metric, but also in the validation of a given model regarding class imbalance. Reference [9] investigated the effect of absolute rarity on model validation and showed that the validation performance of a model is affected by randomness in the model, which leads to an incorrect evaluation of the generalization of the model. When evaluating models by cross-validation, the partitioning of the already absolute rare minority instances into k separate validation sets is noticeably worse [20]. Absolute rarity also favors the induction of *small disjuncts* in the learned classifier. Small disjuncts are regions that cover only a few training instances, and it has been observed in many empirical studies, that small disjuncts have a much higher error rate than large disjuncts [21]. Finally, we briefly describe the severe effect of rather common issues in machine learning, namely data shift, noise, and high dimensionality of data in combination with class imbalance. *Data shift* is a change of the distribution between the training and testing set, which occurs frequently and leads to small performance degradation. For class imbalance, severe performance losses may happen due to this data shift [22]. The additional, negative impact of *noise* in the class imbalance domain is related to the absolute rarity of data. Since the minority class has fewer instances "to work with", noise has a greater impact on the minority class, than on instances of the majority class [21]. Some domains are represented by *high dimensional* data, and the main challenge concerning class imbalance is to select features containing the key information for the corresponding problem. In general, feature selection is considered an appropriate countermeasure for class imbalance [22], which is the topic of the next Section.

C. CLASS IMBALANCE COUNTERMEASURES

In the past 20 years, a growing number of countermeasures for handling class imbalance evolved. For example, in 2019 at least 85 different variants of the most common oversampling method SMOTE [3] existed. With the growing number of countermeasures, also a growing number of taxonomies emerged ([1], [2], [5], [8], [12], [14], [22], [23]. In this work, we will refrain from introducing an additional taxonomy and refer to renowned publications in the area of class imbalance, and elaborate on their key aspects. Reference [1] presented one of the first taxonomies, in which the authors introduced two basic approaches: data-level and algorithm-level approaches.

Data-level approaches focus on solutions on data-level, which include different forms of resampling, generating synthetic data, or a combination of the aforementioned approaches.

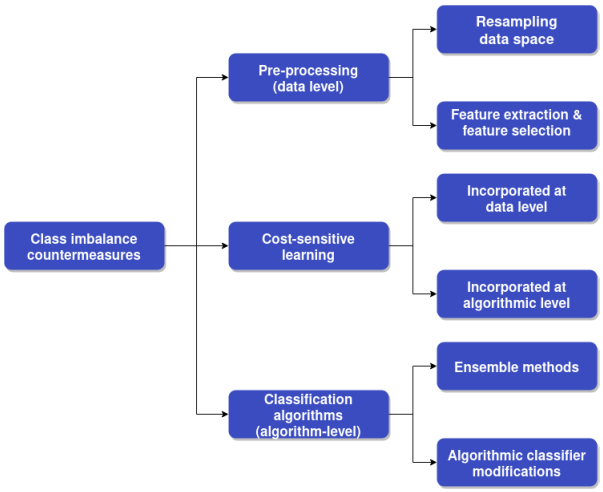


FIGURE 1. Basic approaches of class imbalance countermeasures and their representatives.

Algorithm-level approaches are mainly dealing with the adaptation of existing algorithms. This can be accomplished by introducing an appropriate inductive bias, for example, for decision trees by adjusting probabilistic estimates at tree leaves, or by inventing novel pruning techniques. Modifying the learning paradigm by treating the problem as a one-class learning problem, such as with one-class support vector machines (OCSVM) [1] is another option.

Although very similar to [1] and [2] represents a more "up-to-date" version of the basic approaches and their representatives, discussed in more detail in the rest of this subsection. We provide an overview of these approaches in Figure 1.

Resampling data space or just resampling is used to re-balance the minority and the majority classes. This is performed by either removing instances of the majority class, referred to as *under-sampling*, or by adding instances of the minority class, referred to as *over-sampling*. The main advantage of both approaches is the independence of the classifier being used [23]. There are both "informed" and random variants for both techniques[1]. The primary disadvantage of under-sampling is the potential disposal of useful information by removing instances of the majority class. The drawbacks of over-sampling depend on the used variant, for example, since random over-sampling creates exact copies of minority instances, the chances for over-fitting are increased, and for focused over-sampling variants more specific decision regions are created [14].

Feature extraction is the overall concept for both feature construction and feature selection. While feature construction creates the representation of the data to model, for example transforms the data, such as standardization, normalization, or discretization[2]. *Feature selection* is the process of selecting

[1] "Random" is actually meant literally here: The instances to remove or to be replicated are selected randomly.

[2] We stick to the term "feature extraction" as introduced by [2], instead of using the term "feature construction".

the most relevant features from data and is used to improve the prediction performance of models by reducing the dimensionality of data. This reduction leads to simpler and less complex models, fostering a more profound understanding of the underlying process [24]. For class imbalance, specifically designed feature selection methods, for example, class decomposition-based feature selection, which is applied on smaller pseudo-subclasses of the majority class, exist [5].

Cost-sensitive learning handles the integration of costs into models or introducing different weights of instances in the data space. Since this integration could happen both on data-level and algorithm-level, cost-sensitive learning is often assigned to both levels of approaches [1]. We will integrate techniques for both approaches, namely the integration of different class weights, and cost-sensitive post-processing in the form of changing or calibrating the prediction probabilities in a cost-sensitive manner [22]. Next to calibrating, there also exists a technique referred to as the threshold method, or only thresholding.

Thresholding is a cost-sensitive meta-learning method, that can be integrated into any classifier that produces probability estimates on training and test data. In essence, thresholding selects the best probability from training instances as a threshold, and uses this threshold to adapt the probability estimates of test instances according to class imbalance, to minimize total misclassification cost [25]. In real-world applications, the cost of misclassifying the minority and the majority class typically vary. For example, misclassifying a healthy person as sick will "only" lead to further medical investigations, whereas the costs of misclassifying a sick person as healthy will lead to excessive health cost and potential fatality [14]. Cost-sensitive learning is often combined with ensemble methods, for example, the models AdaC1-AdaC3 and AdaCost, which introduce costs into the weight updating strategy of AdaBoost [12], one of the first ensemble methods.

Ensemble methods are based on a simple principle: "The Wisdom of the Crowd". The wisdom of the crowd is the collective opinion of a group of individuals, rather than that of a single expert[3]. For ensemble methods, this idea is applied as follows: the predictions of multiple predictors are aggregated, and ideally, each predictor represents a different hypothesis. This aggregated prediction results in a better prediction than the best individual predictor. This group of predictors is referred to as an ensemble [26]. Ensembles typically improve generalization performance in many scenarios, and according to [27] the reasons can be summarized as follows:

- Overfitting avoidance - Averaging multiple hypothesis reduces the risk of choosing an incorrect hypothesis.
- Computational advantage - Single learners that conduct local searches may get stuck in local optima, but several learners decrease the risk of obtaining a local minimum.
- Representation - An optimal hypothesis may be outside the space of any single model, but combining different models, the search space may be extended, which achieves a better fit to the data space.

Due to its nature, AdaBoost was one of the first ensemble methods to be used for tackling the class imbalance problem. AdaBoost is an iterative boosting algorithm, where each training instance is weighted based on the error of the previous model, resulting in putting the most weight on hard-to-classify instances. This approach is equivalent to resampling the data space by combining under- and over-sampling [1]. Nowadays, ensemble methods belong to the most popular approaches for handling class imbalance [8]. For example, the authors of [2] presented that from 527 reviewed papers, 218 papers proposed novel ensemble methods or applied existing ensemble models to solve practical class imbalance problems.

Algorithmic classifier modifications adapt the learning ability of existing classification algorithms, to improve their prediction performance for imbalanced data. These adaptions are manifold, for example by enhancing the discriminatory power of classifiers, such as SVM or nearest neighbors using kernel transformations, or fuzzy-based methods integrated into decision trees [2]. Recently, a new breed of classification algorithms handling class imbalance evolved, referred to as *hybrid approaches*- Hybrid approaches employ more than one machine learning algorithm, often through hybridization with other learning algorithms, to alleviate the problem of sampling, feature subset selection, cost matrix optimization, and fine-tuning of classical learning algorithms [5]. One group of representatives for hybrid methods for class imbalance are *active learning* approaches [22]. Traditionally, only used to handle unlabeled training data, active learning has also been used to solve problems related to class imbalance. The main principle is to select only the most informative instances to train a model. These active learning approaches are typically integrated into *kernel-based methods*, mostly involving SVMs [12]. Due to the skew associated with class imbalance, the hyperplane of SVMs is usually located closer to the minority class.

New kernel-based approaches adapt the kernel function, to try to "bias" the hyperplane so that it is "moved" further away from instances of the minority class [22]. SVMs are also the most used algorithms to implement the one-class learning paradigm [1]. Instead of learning the boundaries to distinguish the majority from the minority class, one-class learning attempts to learn boundaries that surround only the class of interest, for example, next to SVMs, autoencoders are used more recently [22]. Due to this increased use of more complex models, e.g., SVMs or neural networks, the consideration of explainability and, above all, interpretability in class imbalance countermeasures is becoming more and more important.

D. EXPLAINABLE ARTIFICIAL INTELLIGENCE AND INTERPRETABILITY

Recently, researchers in the field of AI and especially in ML have focused purely on improving prediction performance,

[3]https://en.wikipedia.org/wiki/Wisdom_of_the_crowd

leading to complex models with exceptional predictive abilities. These complex models have the drawback that both AI users and AI experts are unable to comprehend their predictions, their inner structure, and the process that generated them. This development led to a new research field aimed at making AI systems and their results more understandable to humans, namely explainable AI (XAI) [28]. Currently, the majority of research in XAI is focused on explicating black box models through the utilization of post-hoc analysis and techniques that scrutinize previously trained models and their predictions [29]. Examples of such techniques are Local Interpretable Model-agnostic Explanation (LIME), SHap-ley Additive exPlanations (SHAPE), surrogate models, partial dependency plots, sensitivity analysis, and feature importance [28]. From the perspective of the philosophy of science, explanations have a heuristic function, should guide for further inquiry and, since their delivery is not always an end point, must be considered a continuous process [30]. All of these functions are typically not provided for the aforementioned techniques. Two fundamental inconsistencies with the concept of explaining black box models were identified by recent work from XAI. First, since explanation techniques do not have perfect fidelity regarding the original model, these techniques are not faithful. Otherwise, they would not be needed, or they could be used without the models they try to explain. Second, explanation techniques level out some information to prevent too much information concerning the to-be-explained model, and hence there is some information loss [31].

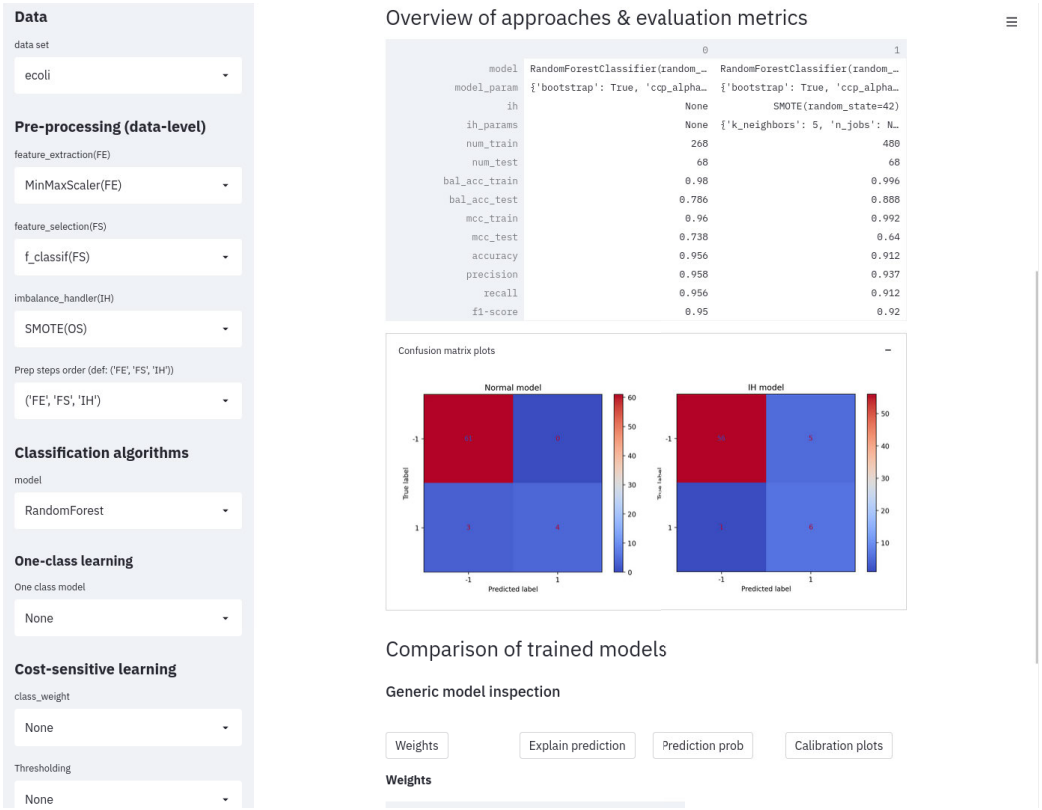
Black box models require post-hoc explanations that are not faithful or leave out information. Instead, white box or interpretable models, such as linear regression, rule-based learning, decision trees and Bayesian models "naturally include" or have "built-in" mechanism to understand how their predictions are made. Especially for well-structured data with understandable features, the difference in prediction performance between complex black box models and white box models with inherent interpretability is often not significant [31]. Although Bayesian models allow for probabilistic inference by modeling the conditional probabilities between the variables, if not decomposable, they could fall into the category of complex black-box models due to their acyclic nature [32]. The central concept is **interpretability**, a term that describes, how well a human can understand decisions in a given context, or more generic, "the degree to which an observer can understand the cause of a decision" [33]. The key to grasping the concept of "interpretability" is "understanding", which allows us to distinguish interpretability from explainability.

In the past, there was little consensus on what "interpretable" or "interpretability" in ML means [7]. More recently, research refers to it as the ability to direct transparent modeling mechanisms, and interpretability is used for comprehending how a certain prediction of a model works as a whole [34]. Reference [35] gather and unified terms dealing with the problem of interpretability and state that interpretability itself, is associated with three sub-problems, namely, accuracy, efficiency, and understandability. "Accuracy" focuses on the fact that one can always create a trivial, easily understandable model that has no connection to the data and therefore no real prediction performance. "Efficiency" is concerned with the time that is necessary for an AI user or an AI expert to grasp or understand a model, which is related to the understandability of that model. Without this efficiency constraint, one could use an infinite amount of time to understand a model, which is not constructive. In consequence, even white box models could happen to be not interpretable by both AI users and AI experts, for example, imagine a linear model like logistic regression which uses several dozen or even hundreds of features. Humans can only understand the predictions of interpretable models, if the number of features used in the model are small, since each feature has a level of contribution towards the final prediction [36].

From the different roles of AI users and AI experts, there are no major differences concerning the first two sub-problems accuracy and efficiency, since both can be measured objectively. However, understandability depends on the different knowledge and requirements of these two roles, requiring different definitions. For an AI user, understandability is given, when he or she can grasp the process of a model's predictions, for example, by inspecting the formula for a regression model, or by visualizing a decision tree. This ''why'' a prediction is made is not sufficient for an AI expert. For an AI expert, understandability is only given, when he or she can fully understand and reproduce the process or algorithm that created the model (how?). We even would go so far as to claim that this process or algorithm must be reproducible manually, for example, with a pencil and paper in efficient time. This claim for full understandability should be given for the overall process of the creation of models, i.e., interpretability should be given for all involved pre-processing and model selection steps that contribute to the creation of the final model.

III. RELATED WORK

The work [20] focuses on the application of oversampling techniques on imbalanced data, for which the authors focused on three main categories. The first category, "learning from imbalanced data", focused on works that performed extensive experiments to evaluate various techniques. The second category, namely "comparing approaches in a specific context", performed a comparison of specific algorithms in a specific context. The third category, "solving a classification problem", had the main objective of solving a particular classification problem, where the imbalance problem was not the focus of the paper. Within our work, we focus on the application of countermeasures for class imbalance and their impact on interpretability, categorizing our work into the aforementioned "learning from imbalanced data" category. Within this category, we investigated all evaluations, reviews, and surveys concerning class imbalance, and found two



	0	1
model	RandomForestClassifier(random_…	RandomForestClassifier(random_…
model_param	{'bootstrap': True, 'ccp_alpha…	{'bootstrap': True, 'ccp_alpha…
ih	None	SMOTE(random_state=42)
ih_params	None	{'k_neighbors': 5, 'n_jobs': N…
num_train	268	480
num_test	68	68
bal_acc_train	0.98	0.996
bal_acc_test	0.786	0.888
mcc_train	0.96	0.992
mcc_test	0.738	0.64
accuracy	0.956	0.912
precision	0.958	0.937
recall	0.956	0.912
f1-score	0.95	0.92

FIGURE 2. Screenshot of the web application for interactive evaluation of class imbalance countermeasures and their effects on interpretability.

publications are related to class imbalance countermeasures with some representative of XAI.

In [8] the authors identify six vital areas of research, namely classification, regression, clustering, data streams, big data analytics and applications which are discussed and finally suggestions concerning lines of future research are presented. The authors pointed out in their Section "Imbalanced big data" that interpretable classifiers that can handle massive and skewed data are of future interest. The authors stated that more focus should be put on a more profound understanding of the structure and the nature of the instances of the minority class, to incorporate background knowledge into the training procedure of new classifiers. The main focus of this survey is to discuss open issues and challenges to further develop the field of imbalanced learning.

Reference [22] presents a survey of existing approaches for handling class imbalance, for which the authors discuss the main challenges, define the problem of class imbalance, propose their taxonomy and summarize the conclusion of existing studies. According to the authors, both pre-processing (data level) and special-purpose learning methods (algorithmic level) should be biased towards the goals of the user, to receive models that are more interpretable and comprehensible for the user. Focusing on prediction post-processing methods, for example, threshold method, and cost-sensitive post-processing the authors claim that the model's interpretability may be jeopardized, by optimizing loss functions that do not follow user preference bias. In the conclusion section, the authors state, that there is still a need to understand why, when and how class imbalance countermeasures work and that not much work exists involving comparisons among the main different types of approaches.

To identify additional works, we also included one relevant work from the second category in which approaches in specific contexts are compared. Reference [6] investigated the impact of "class rebalancing techniques" (re-sampling) on the performance measures and the interpretation of defect prediction models. The main goal was to find out which re-balancing techniques, classifiers, and evaluation metrics are beneficial for the prediction performance of defect models. Concerning interpretability, the authors came to the conclusion, that regardless of which class rebalancing technique was used, the learned concepts (classes) shifted, which was presented in the form of changes of feature importance. Although rebalancing techniques can be beneficial for prediction performance, depending on the model and used technique, the authors recommend that class rebalancing techniques should be avoided when the models are used to derive knowledge and understanding from defect models.

In the current state-of-the-art literature, integrated exploration of class imbalance countermeasures within the domain of XAI are under-represented. Our study aims to fill this void by introducing a cohesive evaluation addressing interpretability, accommodating both AI users and AI experts' perspectives.

TABLE 4. Overview of class imbalance countermeasures in terms of interpretability for both AI users and AI experts.

Countermeasure	Interpretability		Description
	AI user	AI expert	
Resampling data space	No	No	Due to random sampling, understandability not given.
Feature extraction	No	Yes	Understandability of feature values, without re-transformation or visual aids not given.
Feature selection	Yes	Yes	If method or model used is interpretable.
Integrating of class weights	Yes	Yes	If model used is interpretable, only obstacle, "real" weights or costs often not known.
Calibration	Yes	Yes	If used model is interpretable and uses prediction estimates.
Ensemble methods	No	No	Understandability only given, if interpretable base model and low number of models, which could affect accuracy.
One-class learning	No	Yes	Understandability for SVM (AI user: complex transformations) and isolation forest (AI expert: sampling) not given.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

We conducted our experiments visually and interactively to understand the effects of class imbalance countermeasures on interpretability. We developed a simple, yet powerful Python web application (see Figure 2). The main principle of these visual and interactive experiments is the comparison of two models: one model (baseline) that does not apply the selected class imbalance countermeasure, and one model for which the selected countermeasure is applied. With this approach, we assure the ad hoc comparison of the effect on the trained models. Concerning the general pre-processing steps, we perform the selected feature extraction and feature selection algorithms always for both models. Since the selection of an appropriate evaluation metric is crucial for modeling class imbalance (see Section II-A), we provide the most common evaluation metrics in our experiments, for example, balanced accuracy, precision, recall, and F1-score. Nevertheless, the comparison of different approaches and models is solely based on the MCC. Due to the simplicity and the interactive nature of our experiments, we split the data sets into training and test sets. This split is performed "stratified", leaving the same proportion of minority instances in the training and the test set.

B. DATA SETS

Concerning the data sets used in our experiments, we use the data sets provided in the library imbalanced learn [37].[4] The 27 available data sets are taken from various sources, for example, the UCI,[5] and can be downloaded and used directly within the API of the library.

C. PYTHON WEB APPLICATION

For the visual and interactive evaluation of class imbalance countermeasures, we developed a Python web application.[6] Within this web application, we provide techniques for all approaches using their default parameters from the corresponding libraries, for example, feature extraction and feature selection methods, multiple sampling variants, ensemble classification models, adapted classification models, thresholding and calibrations. A full list of available approaches, classification models and other techniques is available in the Appendix. For more specific comparison of the trained models, we provide different local and global explanations in the section "Comparison of created models". In general, one can globally inspect a model by comparing the "weights"[7] of different models.

Hardware: The app is designed to work on normal desktop computers or notebooks, for example, the authors used a Lenovo notebook type T470p with Ubuntu 20.04.3 LTS, Intel Core i7-7700HQ CPUE with 2.8 GHz and 4 cores, 16 GB of RAM.

V. RESULTS AND DISCUSSION

In Table 4, we present the analysis of the interpretability of selected class imbalance countermeasures for both AI users and AI experts. As a short recap (for details see Section II-D), we listed the three sub-problems associated with a method's or models' interpretability:

1) Accuracy - Prediction performance is given, instead of trivial model
2) Efficiency - Time to understand a method or model
3) Understandability:
   - AI user - Grasp the process of a method or model's outcome or prediction (why?)
   - AI expert - Understand and reproduce the output of a method or model (how?)

Given the information in Table 4, we point out, that the interpretability of almost all countermeasures heavily depends on the model used within a countermeasure. For example,

[4] https://imbalanced-learn.org/stable/data sets/index.html
[5] https://archive.ics.uci.edu/ml/index.php
[6] https://gitlab.know-center.tugraz.at/dcemernek/cicm-paper

[7] Used in library eli5 as a term for explaining model-specific parameters, for example, the feature-importance of tree-based models

TABLE 5. Weights of selected features before and after resampling using SMOTE.

Feature	Before resampling	After resampling
13	0.5649	0.8494
4	0.1670	0.0337
174	0.0835	0
239	0.0525	0
127	0	0.0077
8	0	0.0552

feature selection with recursive Feature Elimination (RFE) can both be interpretable, when using logistic regression as a wrapper and non-interpretable, when using random forest. In the following subsections, we provide a detailed analysis of this overview.

A. RESAMPLING DATA SPACE

Following the mathematical analysis of [23], we first show the influence of resampling on the feature importance of the arrhythmia dataset for a selected model (i.e., decision tree) in Table 5. The selected resampling technique, SMOTE, gives similar weights to all features while adding new instances of the minority class. This addition of new data instances significantly changes the importance of certain features, including diminishing the weights of some features. Our findings of this experiment align with the study of [38] that resampling has significant impacts on models and should be avoided in tasks where interpretability is a constraint.

We use another example to show how the models themselves can change if resampling is used. We therefore visualize two decision trees built on the arrhythmia data set. For the first decision tree in Figure 3(a) the SMOTE algorithm was used before training, and in Figure 3(b) we see a decision tree, without any pre-processing. From the differing models, we conclude that the decision tree without pre-processing is even simpler, given by:

- Fewer leaves (nine compared to ten)
- Lower depth (four compared to six)
- Checks fewer variables (seven compared to nine)

Sampling approaches have positive effects on the performance of classification models, which was already shown by many authors before [2], [6], [39], [40]. However, the effects on the trained models are often neglected, since as we showed on the comparison of the trained models with and without resampling. From our experiments, we can not conclude that the trained models are always more complex, but the importance of features, the predictions, and the prediction probabilities of models do change, since underneath sampling approaches change the class distribution. To emphasize our argumentation, we build on the findings in literature. For example, [41] stated that under-sampling has a significant impact on the results of random forests, since especially for smaller data sets, removing a few observations has a high impact on the error curve of the trained model. Additionally, under-sampling is the disadvantage of the potential disposal of useful information by removing instances of the majority class [14]. For over-sampling experiments, [40] suggested that for decision trees, the measured syntactic complexity is higher for models where over-sampling was used. The syntactic complexity was represented by the mean number of induced rules and a mean number of conditions per rule of the induced models. The drawbacks of over-sampling depend on the used variant, for example, since random over-sampling creates exact copies of minority instances, the chances for over-fitting are increased, and for focused over-sampling variants more specific decision regions are created [14]. Regarding the impact on trained models, we follow [6] arguing that the derivation of knowledge that has been generated with re-sampling approaches is not recommended.

TABLE 6. Sampled rows of the E. coli dataset before feature extraction.

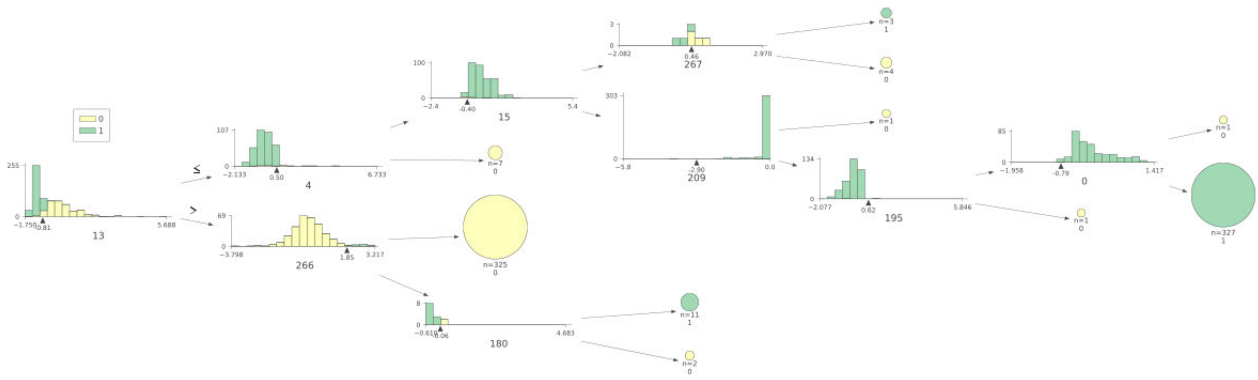
mcg	gvh	lip	chg	aac	alm1	alm2	Label
0.38	0.40	0.48	0.50	0.63	0.25	0.35	-1
0.34	0.51	0.48	0.50	0.44	0.37	0.46	-1
0.74	0.78	0.48	0.50	0.75	0.54	0.15	-1
0.32	0.42	0.48	0.50	0.35	0.28	0.38	-1
0.35	0.48	0.48	0.50	0.56	0.40	0.48	-1
0.54	0.49	0.48	0.50	0.40	0.87	0.88	-1
0.46	0.59	0.48	0.50	0.36	0.76	0.23	-1
0.27	0.42	0.48	0.50	0.37	0.38	0.43	-1
0.34	0.46	0.48	0.50	0.52	0.35	0.44	-1
0.39	0.41	0.48	0.50	0.52	0.72	0.75	-1

TABLE 7. Sampled rows of the E. coli dataset after feature extraction, for which the feature values completly changed in comparison to the raw feature values from Table 6.

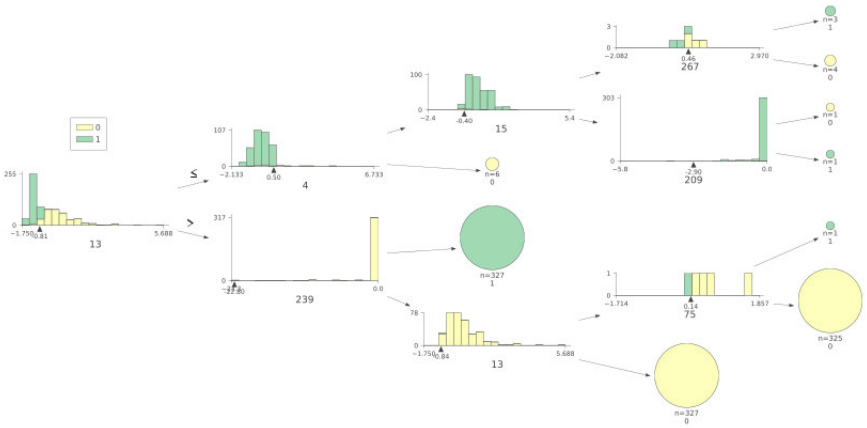
mcg	gvh	lip	chg	aac	alm1	alm2	Label
-0.3721	-0.4308	0	0	0.8667	-0.5526	-0.2222	-1
-0.4961	0.2462	0	0	-0.4000	-0.2368	0.0833	-1
0.7442	1.9077	0	0	1.6667	0.2105	-0.7778	-1
-0.5581	-0.3077	0	0	-1.0000	-0.4737	-0.1389	-1
-0.4651	0.0615	0	0	0.4000	-0.1579	0.1389	-1
0.1240	0.1231	0	0	-0.6667	1.0789	1.2500	-1
-0.1240	0.7385	0	0	-0.9333	0.7895	-0.5556	-1
-0.7132	-0.3077	0	0	-0.8667	-0.2105	0	-1
-0.4961	-0.0615	0	0	0.1333	-0.2895	0.0278	-1
-0.3411	-0.3692	0	0	0.1333	0.6842	0.8889	-1

B. FEATURE EXTRACTION

For classification models sensitive to feature scaling or normalization, for example, logistic regression or SVM, feature extraction is obligatory. In general, we emphasize, that feature extraction has an impact on interpretability, since the original feature values change, and therefore, at least, AI users cannot understand the model's prediction, without additional methods. We demonstrate this concept by performing the feature extraction on ecoli protein dataset, Table 6 and Table 7 show the original and changed feature values respectively. While certain robust methods for feature extraction enhance model performance by eliminating low-entropy features, they may pose challenges in interpretability for AI users. This is evident in Table 7, where these techniques result in significant alteration of feature values. The substantial transformation introduced by feature extraction methods can impede the straightforward understanding of the relationships between input features and model



(a) Decision tree trained on the data set arrhythmia pre-processed with the SMOTE algorithm



(b) Decision tree trained on the data set arrhythmia

FIGURE 3. Decision tree trained with/out SMOTE algorithm.

outcomes. The absence of the raw feature values, hinders interpretability, since AI users cannot understand the model's predictions.

C. FEATURE SELECTION

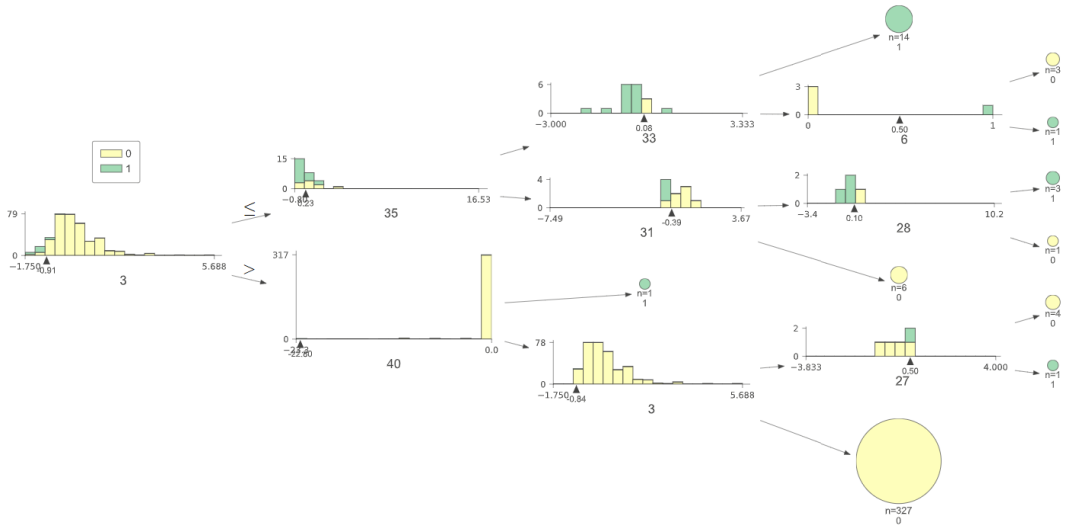
Feature selection is beneficial, especially in an imbalanced learning setting. Due to the lower number of minority instances, these instances are easily regarded as noise, but by removing irrelevant features this risk is reduced [2]. Concerning interpretability, Figure 4(a) shows a decision tree trained on the arrhythmia dataset without feature selection, achieving a MCC test score of 0.713. Figure 4(b) shows a decision tree trained with feature selection achieving an MCC of 0.77. Both models use the same number of features, but partially different ones. This example provides visual evidence for our argument that feature selection has no adversary impact on the interpretability of a model. Feature selection filters out the insignificant features, thus helping the model to distinguish between the majority and minority classes, resulting in an improved MCC. An appropriate feature selection method applies the reduction to only relevant features, leading to simpler models or situations where simpler and interpretable models can be built, which again are easier to understand by both AI users and AI experts. We repeated our experiment on the same data set with another interpretable model, namely logistic regression. Therefore, we observed a significant improvement of 42% in the MCC test score after using feature selection (MCC = 0.226 without feature selection and MCC = 0.654 after feature selection. PAM results are in agreement with the results obtained for MCC before and after feature selection (PAM = 0.22 and PAM = 0.59 respectively).

D. COST-SENSITIVE LEARNING

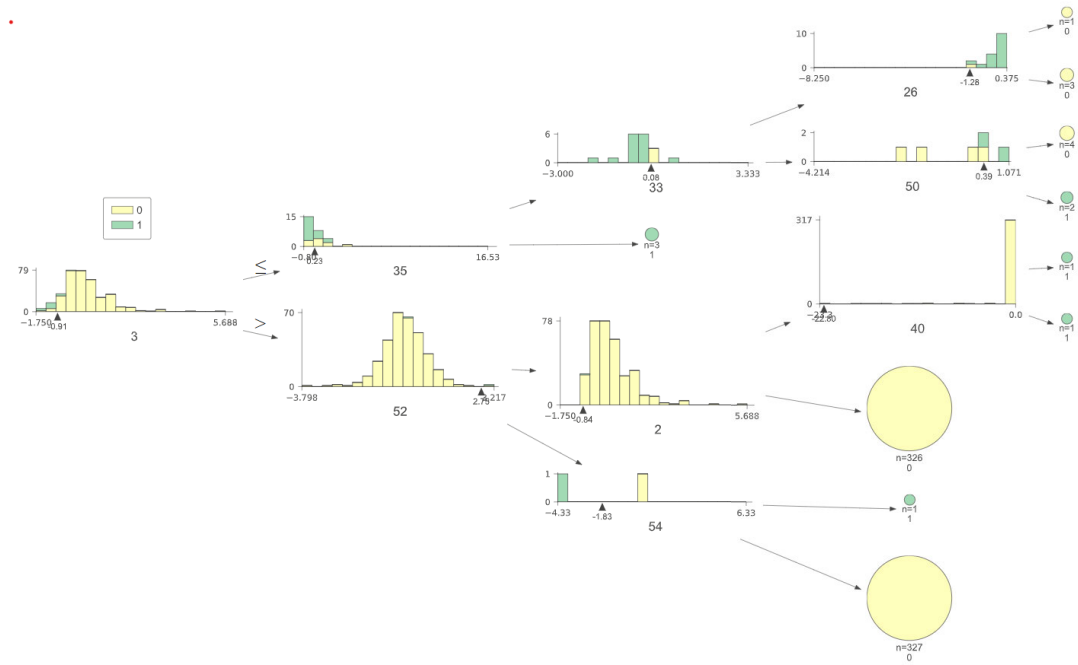
The experiments for cost-sensitive learning investigated the impact of the class weights' parameter, and the post-processing calibration of prediction probabilities, for classifiers supporting it.

1) INTEGRATION OF WEIGHTS

In this subsection, we break down the results for a normal and a model that integrated weights, to transparently show the effects on the confusion matrix and ultimately the predictions for the integration of weights. We trained both logistic regression and decision tree on the yeast_me2 data set. Unfortunately, both models received weak results, so we turned to the random forest model. The "normal" random forest achieved a MCC test score of 0.213 and 0.354 for



(a) Decision tree trained on the data set arrhythmia with feature selection using f_classif



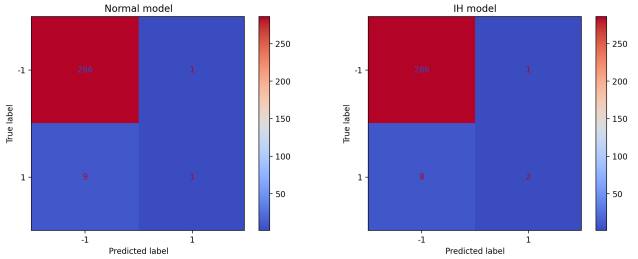
(b) Decision tree trained on the data set arrhythmia with no feature selection

FIGURE 4. Decision tree trained with/out feature selection algorithm.

both the balanced and balanced-subsample options for the yeast_me2 data set. The confusion matrix for the normal model is presented in Figure 5(a), and the corresponding confusion matrix for the model with the balanced option is presented in Figure 5(b). The confusion matrices are only differing for the "True-Negative" column, where the normal model correctly predicted only one minority example, whereas the balanced model correctly predicted two minority examples.

In Table 8, we provide an overview of the weights (in that case "feature importance") for the features in the two models. For this table, we intend to point out, that the weights for features zero and two are higher in the balanced model. Additionally, the weights for features one, three, and four differ, resulting in a slightly different model.

Furthermore, in Table 9 we provide the prediction probabilities for the test instances for both models. With these prediction probabilities, we see that for the test instance



(a) Original data (b) Balanced data

FIGURE 5. Confusion matrix of the imbalanced and balanced (IH model = imbalance handling model) test instances using random forest model for the data set yeast_me2.

TABLE 8. Comparison of model weights and their standard deviation (Std) for the different features for the normal and the balanced model for the yeast_me2 data set.

Feature	Normal model		Balanced model	
	Weight	Std	Weight	Std
0	0.2865	0.0859	0.3010	0.1455
1	0.1884	0.0645	0.1398	0.0883
2	0.1691	0.0598	0.3211	0.1355
3	0.1592	0.0625	0.0921	0.0446
6	0.1247	0.0542	0.0868	0.0582
7	0.0591	0.0319	0.0501	0.0345
4	0.0122	0.0149	0.0072	0.0125
5	0.0007	0.0033	0.0019	0.0027

TABLE 9. Excerpt of prediction probabilities for the test instances 105-114 for the normal and the balanced model for the yeast_me2 data set. Column "Mismatch" indicates if there is a mismatch (1) between the prediction of the normal and the balanced model.

Index	Label	Mismatch	Normal model			Balanced model		
			-1	1	pred	-1	1	pred
105	-1	0	1	0	-1	1	0	-1
106	1	1	0.6800	0.3200	-1	0.4500	0.5500	1
107	-1	0	0.9400	0.0600	-1	0.8900	0.1100	-1
108	-1	0	1	0	-1	1	0	-1
109	-1	0	0.9900	0.0100	-1	0.9900	0.0100	-1
110	-1	0	1	0	-1	1	0	-1
111	-1	0	1	0	-1	1	0	-1
112	-1	0	0.9800	0.0200	-1	0.9900	0.0100	-1
113	-1	0	1	0	-1	1	0	-1
114	-1	0	0.9700	0.0300	-1	0.9900	0.0100	-1

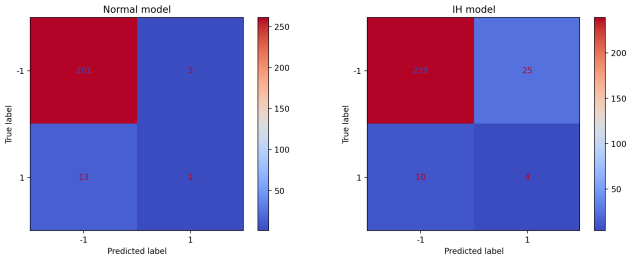
with index 106, the two models make differing predictions, indicated by the column "Mismatch" equals 1. The normal model has a prediction probability for the majority class of 0.68 vs. 0.32 for the minority class. The balanced model has a prediction probability of 0.44 for the majority and 0.56 for the minority class, leading to a correct prediction for this instance for the balanced model. In Table 10 we present the explanation for both predictions, showcasing that the differing weights of variables are the cause for these differing predictions.

2) CALIBRATION

Calibration, as a prediction post-processing strategy for handling imbalanced data [22], has no effect on the model structure or model (hyper-)parameters. Therefore, we only show the effects of calibration on the confusion matrix and prediction probabilities. For example, for the data set solar_flare_m0 we presented the confusion matrix for the normal model in Figure 6(a), and the confusion matrix for the calibrated model in Figure 6(b). The calibration of the model

TABLE 10. Explanation of the predictions for the test instance with index 106 for the normal and the balanced model for the yeast_me2 data set. The column "Target" is the predicted target class, and the column "Weight" represents feature importance.

Feature	Feature value	Normal model		Balanced model	
		Target	Weight	Target	Weight
<BIAS>	1	-1	0.9649	1	0.5017
3	0.2700	-1	0.0106	1	-0.0745
1	0.6100	-1	0.0077	1	-0.1037
4	0.5000	-1	0.0004	1	0.0004
5	0	-1	-0.0007	1	0.0018
7	0.2200	-1	-0.0102	1	0.0054
6	0.5200	-1	-0.0491	1	0.0256
0	0.6900	-1	-0.1083	1	0.0791
2	0.3800	-1	-0.1353	1	0.1141



(a) Baseline model (b) Calibrated model

FIGURE 6. Confusion matrix of the test instances for the baseline and calibrated (IH model = imbalance handling model) random forest model for the data set solar_flare_m0.

led to an additional three test instances, that are correctly classified as minority, but also led to additional 22 test instances incorrectly classified as minority instances. The change of prediction probabilities is presented in Table 11, where the probabilities for the test instance with index 19 changes from majority class in the normal model, to the prediction of minority class in the calibrated model.

E. ENSEMBLE METHODS

Within this section, we try to investigate the question, how simple, interpretable models perform compared to the complex, and class imbalance-specific models. To compare logistic regression, we added a feature extraction step, namely RobustScaler[8] to scale the features accordingly. We use the random forest model as our baseline, and compared logistic regression and decision tree as interpretable models, with the complex and class imbalance-specific ensemble methods. In Table 12 we provided the results for these experiments. For the eleven data sets, for which the baseline model achieved a MCC test score below 0.5, we also tested all available complex models (column "Best-Complex"). For the remaining 16 data sets, for which the baseline already was above a MCC test score of 0.5, we only compared the baseline model with decision tree and logistic regression. From the overall 27 data sets, the simple models performed better than the baseline for 15 data sets, indicating, that the simple models are at least a good starting point for further

[8]MinMaxScaler tends to favor scaling for outliers, and StandardScaler assumes that the data is normally distributed

TABLE 11. Excerpt of prediction probabilities for the test instances with index from index 19-28 for the normal and the calibrated model for the solar_flare_m0 data set. Column "Mismatch" indicates if there is a mismatch (1) between the prediction of the normal and the balanced model.

Index	Label	Mismatch	Normal model			Calibrated model		
			-1	1	pred	-1	1	pred
19	1	1	0.5608	0.4392	-1	0.4754	0.5246	1
20	-1	0	1	0	-1	0.9923	0.0077	-1
21	-1	0	0.2282	0.7718	1	0.0208	0.9792	1
22	-1	0	0.9867	0.0133	-1	0.9910	0.0090	-1
23	-1	0	0.9525	0.0475	-1	0.9869	0.0131	-1
24	-1	0	1	0	-1	0.9923	0.0077	-1
25	-1	0	1	0	-1	0.9923	0.0077	-1
26	-1	0	0.9800	0.0200	-1	0.9904	0.0096	-1
27	-1	0	1	0	-1	0.9923	0.0077	-1
28	-1	0	1	0	-1	0.9923	0.0077	-1

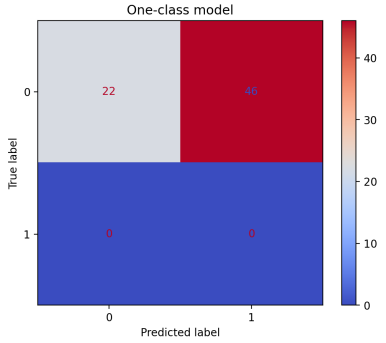


FIGURE 7. Confusion matrix for an isolation forest trained on the ecoli data set only learning the majority class. Since the minority class is never correctly predicted, the MCC is zero.

model selection and (hyper)-parameter tuning. At least for these data sets, the "general" trade-off between accuracy and interpretability is not true [31].

F. ONE-CLASS LEARNING

Finally, we examine the inner workings of one-class learning with a focus on the comparison of the two main different approaches, learning only the minority vs. learning only the majority class. For each of the 27 data sets we trained both representatives, namely Isolation Forest and OneClass-SVM and analyzed the MCC test scores. We tested both the learning of the minority and the majority class. The training of the majority class almost always resulted in a MCC test score of 0.0.[9] We provide an example of the confusion matrix for the ecoli data set with the isolation forest only trained for the majority class in Figure 7, in which it is shown that the minority class is completely missed, resulting in a MCC of 0. Due to this, we only report the MCC test scores for the training of the minority class (see Table 13). Concerning the other data sets, it appears that these algorithms need further inspection and tuning to learn the minority instances. To improve their performance, the structure of the minority instances is important information, that should be integrated in the form of background knowledge into classifiers [8].

[9]MCC is 0 if one of the four values of the confusion matrix is zero, then MCC is also zero.

G. SUMMARY

Although feature extraction is not changing the relationship between of features itself, the effect of feature transformation must at least be considered. For example, by providing the parameters of feature scaling, such as the min and max values used for a Min/Max-Scaler to AI users.

Although feature selection changes the number of features, it does not change the distribution of remaining feature values or the distribution of instances. For interpretability, the key aspect is that an appropriate feature selection method applies the reduction to only relevant features, leading to simpler models or situations where simpler and interpretable models can be built. This provides full interpretability for both AI users and AI experts.

For re-sampling approaches, the argument of [42], that over-sampling does not add information and under-sampling actually removes information, and research strategies should concentrate on ML algorithms that can deal most efficiently with whatever data they are given, is absolutely valid. From the perspective of interpretability and based on our experiments, we can reinforce this finding by stating that models that were created using sampling methods should not be used in a setting where interpretability is a requirement. For these settings, approaches that do not change the distribution of input data should be preferred [14].

For both cost-sensitive approaches, we conclude that careful monitoring of confusion matrices is required to avoid massive shifts of errors from the minority to the majority class. Assigning costs, or benefits, to the minority and majority instances is not a trivial task. For example, an explicit description of the costs of misclassification is not available in many situations [12], and furthermore, a wrong cost matrix could lead to the situation that some class labels are never predicted [43]. The simplest solution that requires no explicit costs or domain knowledge is assigning weights to different labels by their inverse frequency, for example, via the class weight parameter of scikit-learn (what we did within our experiments). The integration of this parameter into a model selection process is recommended. Especially, for data sets for which models with default parameters achieve lower MCC scores, it is a simple and interpretable way to take class imbalance into account. As our experiment also showed, the use of weights has minor effects on the

TABLE 12. Comparison of baseline model (random forest) with simple, complex, and class imbalance-specific models. MCC for the baseline, decision tree and logistic regression are available in columns "MCC-Baseline", "MCC-DT", and "MCC-LR". Column "MCC Best-Simple-Change" indicates the change from the best simple model compared to the MCC test score of the baseline model. For the data sets, for which the baseline achieved a MCC test score below 0.5, we also tested all available complex, and class-imbalance-specific models: Column "Best-Complex" indicates the name of the best performing complex model, its MCC test score in column "MCC Best-Complex" and the change compared to the MCC test score of the baseline model in column "MCC Best-Complex-Change".

Name	MCC-Baseline	MCC-DT	MCC-LR	MCC Best-Simple-Change	Best-Complex	MCC-Best-Complex	MCC Best-Complex-Change
ozone_level	-0.008	0.134	0.089	+0.142	EasyEnsemble	0.235	+0.243
abalone_19	0	-0.009	0	+0.000	RandomUnderSampleAda	0.128	+0.128
arrhythmia	0	0.713	-0.036	+0.713	Balanced Bagging	1	+1.000
yeast_ml8	0	-0.033	-0.022	-0.022	EasyEnsemble	0.122	+0.122
coil_2000	0.027	0.055	0.057	+0.030	EasyEnsemble	0.164	+0.137
solar_flare_m0	0.11	0.068	0.135	+0.025	RandomUnderSampleAda	0.191	+0.081
abalone	0.127	0.169	0	+0.042	EasyEnsemble	0.358	+0.231
scene	0.208	0.145	0.083	-0.063	BalancedRandomForest	0.287	+0.079
yeast_me2	0.213	0.127	-0.015	-0.086	RandomUnderSampleAda	0.365	+0.152
wine_quality	0.367	0.497	0.155	+0.130	LightGBM	0.368	+0.001
oil	0.394	0.293	0.411	+0.017	RandomUnderSampleAda	0.42	+0.026
us_crime	0.544	0.343	0.472	-0.072	-	-	-
satimage	0.63	0.483	0.106	-0.147	-	-	-
mammography	0.638	0.686	0.52	+0.048	-	-	-
ecoli	0.646	0.477	0	-0.169	-	-	-
isolet	0.728	0.563	0.776	+0.048	-	-	-
libras_move	0.763	0.708	0.763	+0.000	-	-	-
sick_euthyroid	0.801	0.769	0.641	-0.032	-	-	-
webpage	0.803	0.686	0.746	-0.057	-	-	-
spectrometer	0.828	0.712	0.828	+0.000	-	-	-
car_eval_4	0.84	0.959	0.76	+0.119	-	-	-
protein_homo	0.869	0.743	0.857	-0.012	-	-	-
car_eval_34	0.874	0.854	0.829	-0.020	-	-	-
thyroid_sick	0.891	0.956	0.689	+0.065	-	-	-
optical_digits	0.902	0.819	0.842	-0.060	-	-	-
letter_img	0.938	0.942	0.701	+0.004	-	-	-
pen_digits	0.995	0.942	0.788	-0.053	-	-	-

TABLE 13. Comparison of one-class learning approaches with the baseline model. "MCC-Baseline" contains the MCC test score for the isolation forest is depicted in column "MCC-IF", and for the one-class SVM in column "MCC-OCSVM". The column "MCC Best-Change" indicates the difference between the MCC test score of the default model with the best one-class learning approach. The final row contains the number of data sets for which the corresponding approach: Isolation Forest or one-class SVM approach outperformed the baseline model.

Name	MCC-Baseline	MCC-IF	MCC-OCSVM	MCC Best-Change
pen_digits	0.995	0.833	0.673	-0.162
letter_img	0.938	0.316	0.407	-0.531
optical_digits	0.902	0.57	0.558	-0.332
thyroid_sick	0.891	0.105	0.215	-0.676
car_eval_34	0.874	0	0	-0.874
protein_homo	0.869	0.028	0.286	-0.583
car_eval_4	0.84	0	0.272	-0.568
spectrometer	0.828	-0.069	-0.014	-0.842
webpage	0.803	0.011	0.039	-0.764
sick_euthyroid	0.801	0.176	0.288	-0.513
libras_move	0.763	0.221	0.649	-0.114
isolet	0.728	0.297	0.468	-0.260
ecoli	0.646	0.582	0.327	-0.064
mammography	0.638	0.17	0.197	-0.441
satimage	0.63	0.466	0.525	-0.105
us_crime	0.544	0.175	0.128	-0.369
oil	0.394	-0.033	0.066	-0.328
wine_quality	0.367	-0.032	-0.071	-0.399
yeast_me2	0.213	0.079	0.057	-0.134
scene	0.208	0.017	0.061	-0.147
abalone	0.127	0.351	0.382	+0.255
solar_flare_m0	0.11	-0.001	0.087	-0.023
coil_2000	0.027	-0.008	0.017	-0.010
yeast_ml8	0	0	0	+0.000
arrhythmia	0	0.148	0.036	+0.148
abalone_19	0	0.056	0.084	+0.084
ozone_level	-0.008	0.105	0.116	+0.124

inner workings and structure of trained models, although these effects would require additional investigation to gain a better understanding. With the integration of weights, the calibration of models is understandable since it does not modify a model's structure. Furthermore, its effects can be visually represented and analyzed, and the impact of calibration on model behavior can be easily demonstrated and understood by both AI users and AI experts.

The presence of class imbalance encouraged the use of ensemble methods and also led to the adaptation of existing classification algorithms to appropriately consider the challenges imposed by class imbalance. Given the main concept of ensemble methods, the aggregation of the predictions of a group of multiple predictors, almost always involving bagging, is not interpretable. Although it is shown that ensemble models work [26], they typically fulfill the accuracy criteria. Due to their nature, they typically do not fulfill the understandability requirement for both AI users and AI experts. AdaBoost-related ensemble methods are not understandable in terms of their resampling of the data space approach [1]. The same is true for algorithms that involve bagging, for example, random forest and its derivates. Additionally, most ensemble methods only fulfill the accuracy criteria when using a large number (even 100s) of models, which jeopardizes the efficiency criteria since it is not possible to understand an ensemble method within a reasonable time.

We conclude with our findings concerning the interpretability of one-class learning. In relation to interpretability, learning only the majority class makes it easier to understand the behavior and predictions of a model, and it is a useful and interpretable approach for anomaly detection. However, for classification, the tested approaches face the same obstacles concerning interpretability as "common" classification models. To gain a more profound understanding of this research field, further investigation would be necessary, as [44] already stated that explainable or interpretable one-class learning is necessary.

TABLE 14. The table contains the names, technique and type of all used classification algorithms, feature selection, feature extraction methods. The Name column contains links to the api-documentation of the corresponding approach.

Name	Technique	Type
f_classif	Feature Selection	Univariate
mutual_info_classif	Feature Selection	Univariate
Boruta	Feature Selection	All-relevant feature selector
MinMaxScaler	Feature Extraction	Scaler
StandardScaler	Feature Extraction	Scaler
RobustScaler	Feature Extraction	Scaler
DecisionTree	Classification Algorithm	Basic
LogisticRegression	Classification Algorithm	Basic
AdaBoost	Classification Algorithm	Ensemble
RandomForest	Classification Algorithm	Ensemble
HistGradientBoosting	Classification Algorithm	Ensemble
GradientTreeBoosting	Classification Algorithm	Ensemble
BalancedBaggingClassifier	Algorithmic classifier modification	Ensemble (Balanced Bagging)
BalancedRandomForestClassifier	Algorithmic classifier modification	Ensemble (Balanced Bagging)
EasyEnsembleClassifier	Algorithmic classifier modification	Ensemble (Balanced Bagging)
RUSBoostClassifier	Algorithmic classifier modification	Ensemble (Balanced Bagging)
OneClassSvm	Algorithmic classifier modification	One-class-learning
IsolationForest	Algorithmic classifier modification	One-class-learning
SVM	Classification Algorithm	Kernel based methods

TABLE 15. The table contains the names, technique and type of all used resampling approaches. The Name column contains links to the api-documentation of the corresponding approach.

Name	Technique	Type
ClusterCentroids	Under-sampling	Prototype generation
Condensed-NearestNeighbour	Under-sampling	Prototype selection
EditedNearestNeighbours	Under-sampling	Prototype selection
RepeatedEdited-NearestNeighbours	Under-sampling	Prototype selection
AllKNN	Under-sampling	Prototype selection
InstanceHardnessThreshold	Under-sampling	Prototype selection
NearMiss	Under-sampling	Prototype selection
Neighbourhood-CleaningRule	Under-sampling	Prototype selection
OneSidedSelection	Under-sampling	Prototype selection
RandomUnderSampler	Under-sampling	Prototype selection
TomekLinks	Under-sampling	Prototype selection
RandomOverSampler	Over-sampling	Basic
SMOTE	Over-sampling	SMOTE variants
SMOTEN	Over-sampling	SMOTE variants
ADASYN	Over-sampling	SMOTE variants
BorderlineSMOTE	Over-sampling	SMOTE variants
KMeansSMOTE	Over-sampling	SMOTE variants
SVMSMOTE	Over-sampling	SMOTE variants
SMOTEENN	Hybrid	Combination of over- and under-sampling
SMOTETomek	Hybrid	Combination of over- and under-sampling

H. LIMITATIONS

In this section we disclose and address possible limitations of our experiments and evaluation.

a: MODEL SELECTION VIA TRAIN-TEST SPLIT AND NOT WITH CROSS-VALIDATION

Full cross-validation with multiple parameters was not useful due to the interactive nature of our experimental approach. A user should interact and understand both data and the model, which, in our opinion, is not possible with complex hyperparameter tuning.

b: USAGE OF DEFAULT PARAMETERS FOR ALL MODELS AND CLASS IMBALANCE APPROACHES

Simpler models, have fewer parameters, resulting in a potential advantage over more complex models in an advantage, since more complex models provide more options to tune. From a prediction performance perspective, this aspect is true, but from an interpretability aspect additional tuning of hyper-parameters of a black-box model does not make it "more interpretable".

c: NOT ALL REPRESENTATIVES FOR CLASS IMBALANCE APPROACHES ARE AVAILABLE

For most of the missing approaches, for example, active learning or costs for cost-sensitive learning, it is not feasible due to a lack of domain knowledge for the used data sets. Even for domain owners, estimating costs is a complex task. For feature selection, for example, we want to show the basic effect that feature selection could have in a class imbalance setting, and therefore available options seem sufficient.

d: NOT ALL EXPLANATION METHODS ARE AVAILABLE

Concerning the available options, we decided to use pre-existing options from specific libraries, for example, eli5 or tree-interpreter. There exist many other techniques, for example SHAPE, LIME or partial dependency plots, but ultimately, these are only explanations of pre-trained models, with the aforementioned limitations.

VI. CONCLUSION

The purpose of this paper is to investigate the interpretability of class imbalance countermeasures from the perspective of AI users and AI experts. To be considered interpretable, a countermeasure, or more generally a model, must fulfill three criteria, namely, accuracy, efficiency, and understandability. Accuracy is important since one can always create a trivial, interpretable model with no relation to the data. Efficiency focuses on the amount of time needed, to grasp the model, since every (complex) model is understandable with infinitive time. For AI users, a model is understandable as they understand how a model's prediction is materialized. For AI experts, a model is understandable if they can fully trace the process that created a given model. Our experiments indicated that two of the most common class imbalance countermeasures, namely, resampling and ensemble methods, are not interpretable for both AI users and AI experts. Resampling is not understandable due to the random sampling of the data. Ensemble methods are also not understandable due to their nature, with various models or involved re-sampling mechanisms. To be accurate, most ensemble methods require numerous models, which jeopardizes both efficiency and again understandability. Due to the transformation of raw feature values within feature extraction, AI users cannot understand the model's prediction, or only with the help of the original feature values. The two representatives of one-class learning are not interpretable. One-class-SVM, due to the complex feature transformations, is not understandable by AI users, and isolation forest is not understandable by both AI users and AI experts, due to the involved bagging algorithm. Given that the used method or model is interpretable, we conclude that the remaining class imbalance countermeasures feature selection and the two representatives of cost-sensitive learning, namely, integration of weights and calibration are interpretable. Feature selection generally reduces the overall feature set to the most relevant features, leading to simpler, less complex models, without jeopardizing interpretability. For the integration of weights, the usage of different weights for minority and majority classes improves the performance of models without an impact on their interpretability. Calibration has the additional requirement that the used model must support probability estimates, but despite that calibration is usable in settings where interpretability is important. Additionally, both cost-sensitive representatives require monitoring of changes in the confusion matrix, since they typically shift the classification error from minority to majority instances. Our results indicate that interpretability is still an under-explored topic in the field of class imbalance countermeasures, for which the pure prediction performance of class imbalance countermeasures seems to be the main criterion. However, experiments suggested that for 16 out of 27 data sets, simple, interpretable models, i.e., logistic regression and decision tree, can outperform complex models. With this publication, we also provide a guideline for both AI users and AI experts, on which class imbalance countermeasures are usable in settings where interpretability is of importance. Our findings propose, that there is great potential for interpretable class imbalance countermeasures. In one of our future works, we will further investigate the combination of feature selection with techniques of cost-sensitive learning.

APPENDIX

USED METHODS

We assemble a table consisting of all classification algorithms, feature selection, feature extraction in Table 14, and resampling methods in Table 15.

ACKNOWLEDGMENT

The authors would like to thank their reviewers for pointing them to the relevant work. Intrigued by the performance of PAM [17], they have added this metric to their web application to complement their existing measures.

REFERENCES

[1] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009, doi: 10.1142/S0218001409007326.

[2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Exp. Syst. Appl.*, vol. 73, pp. 220–239, May 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417416307175

[3] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105662. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494619304429

[4] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *J. Intell. Inf. Syst.*, vol. 46, no. 3, pp. 563–597, Jun. 2016, doi: 10.1007/s10844-015-0368-1.

[5] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem," *Int. J. Adv. Soft Comput. Appl.*, vol. 7, no. 3, pp. 176–204, 2015.

[6] C. Tantithamthavorn, A. E. Hassan, and K. Matsumoto, "The impact of class rebalancing techniques on the performance and interpretation of defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 46, no. 11, pp. 1200–1219, Nov. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8494821/

[7] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.

[8] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016, doi: 10.1007/s13748-016-0094-0.

[9] S. Kang, "Model validation failure in class imbalance problems," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113190. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417420300166

[10] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," 2021, *arXiv:2108.00071*.

[11] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. London, U.K.: MIT Press, 2010.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5128907/

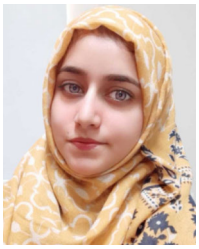
[13] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.

[14] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci.*, vol. 513, pp. 429–441, Mar. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025519310497

[15] A. Luque, A. Carrasco, A. Martín, and A. D. L. Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, Jul. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320319300950

[16] D. Cemernek, S. Cemernek, H. Gursch, A. Pandeshwar, T. Leitner, M. Berger, G. Klösch, and R. Kern, "Machine learning in continuous casting of steel: A state-of-the-art survey," *J. Intell. Manuf.*, vol. 33, no. 6, pp. 1561–1579, Aug. 2022, doi: 10.1007/s10845-021-01754-7.
[17] O. Aydemir, "A new performance evaluation metric for classifiers: Polygon area metric," *J. Classification*, vol. 38, no. 1, pp. 16–26, Apr. 2021.
[18] G. Mulugeta, T. Zewotir, A. S. Tegegne, L. H. Juhar, and M. B. Muleta, "Classification of imbalanced data using machine learning algorithms to predict the risk of renal graft failures in Ethiopia," *BMC Med. Informat. Decis. Making*, vol. 23, no. 1, pp. 1–17, May 2023.
[19] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678, doi: 10.1371/journal.pone.0177678.
[20] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, "Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 4, pp. 59–76, Nov. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8492368/
[21] G. M. Weiss, "Mining with rarity," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 7–19, Jun. 2004, doi: 10.1145/1007730.1007734. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1007730.1007734
[22] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–50, Jun. 2017, doi: 10.1145/2907070.
[23] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025519306838
[24] I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, *Feature Extraction* (Studies in Fuzziness and Soft Computing), vol. 207. Berlin, Germany: Springer, 2006, doi: 10.1007/978-3-540-35488-8.
[25] V. S. Sheng and C. X. Ling, "Thresholding for making classifiers cost-sensitive," in *Proc. Nat. Conf. Artif. Intell.*, vol. 1, 2006, pp. 476–481.
[26] L. Rokach, *Ensemble Learning* (Series in Machine Perception and Artificial Intelligence), vol. 85. Singapore: World Scientific, Mar. 2019. [Online]. Available: https://www.worldscientific.com/worldscibooks/10.1142/11325
[27] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, Jul. 2018, doi: 10.1002/widm.1249.
[28] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8466590/
[29] P. Hall and N. Gill, *An Introduction to Machine Learning Interpretability*. Sebastopol, CA, USA: O'Reilly, 2019.
[30] R. R. Hoffman, G. Klein, and S. T. Mueller, "Explaining explanation for 'explainable AI,'" in *Proc. Human Factors Ergonom. Soc. Annual Meeting*, vol. 1, 2018, pp. 197–201.
[31] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019. [Online]. Available: http://www.nature.com/articles/s42256-019-0048-x
[32] O. Loyola-González, "Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view," *IEEE Access*, vol. 7, pp. 154096–154113, 2019.
[33] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019, doi: 10.1016/j.artint.2018.07.007.
[34] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, Jan. 2021. [Online]. Available: https://jair.org/index.php/jair/article/view/12228
[35] A. Bibal and B. Frenay, "Interpretability of machine learning models and representations: An introduction," in *Proc. 24th Eur. Symp. Artif. Neural Netw.*, 2016, pp. 77–82.
[36] A. N. Richter and T. M. Khoshgoftaar, "Building and interpreting risk models from imbalanced clinical data," in *Proc. IEEE 30th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2018, pp. 143–150. [Online]. Available: https://ieeexplore.ieee.org/document/8576029/
[37] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365
[38] I. Alarab and S. Prakoonwit, "Effect of data resampling on feature importance in imbalanced blockchain data: Comparison studies of resampling techniques," *Data Sci. Manag.*, vol. 5, no. 2, pp. 66–76, Jun. 2022.
[39] V. García, J. S. Sánchez, A. I. Marqués, R. Florencia, and G. Rivera, "Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data," *Exp. Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113026. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417419307432
[40] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.
[41] P. Probst and A.-L. Boulesteix, "To tune or not to tune the number of trees in random forest?" *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6673–6690, May 2017.
[42] F. Provost, "Machine learning from imbalanced data sets 101," in *Proc. AAAI*, vol. 68, 2000, p. 3. [Online]. Available: https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-001.pdf
[43] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2001, vol. 17, no. 1, pp. 973–978.
[44] P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," 2021, *arXiv:2101.03064*.

DAVID CEMERNEK received the Magister (master's) degree in geoinformatics from Johannes Kepler University Linz, Austria, in 2009, and the master's degree in software engineering and management from the Technical University of Graz, Austria, in 2019. He is currently pursuing the Ph.D. degree with the Graz University of Technology, Austria. From 2016 to 2020, he was employed as a Data Engineer, a Data Scientist, and the Project Lead. From 2020 to 2022, he was the Strategic Area Manager of the Area Knowledge Discovery. In 2022, he became the Research Area Manager of the Area Data Management with the Know-Center GmbH. His research interests include machine learning, classification, class imbalance, interpretability, and feature selection.



SHAFAQ SIDDIQI is currently pursuing the Ph.D. degree in computer science with the Graz University of Technology, Austria, under the supervision of Prof. Stefanie Lindstaedt and Dr. Roman Kern. Her interest lies in heterogeneous data preprocessing and challenges with multi-modal and non-IID data. Her research focuses on exploiting data characteristics and ML signals for large-scale data cleaning.

ROMAN KERN received the Ph.D. degree from Graz University of Technology. He is currently a Computer Scientist and an Assistant Professor with the Institute for Interactive Systems and Data Science, Technical University of Graz. Additionally, he is also the Chief Scientific Officer with the Know-Center research center (Competence Center for Trustworthy AI). His research interests include natural language processing and machine learning, with a focus on causal data science. He applies these methods to achieve trustworthy AI in fields like digital libraries, intelligent transportation systems, and smart production.