A Hybrid Machine Learning Approach for Improving Mortality Risk Prediction on Imbalanced Data

Araek Tashkandi*
Institute of Computer Science
Georg-August-University Goettingen
Goettingen, Germany
araek.tashkandi@cs.uni-goettingen.de
astashkandi@uj.edu.sa

Lena Wiese
L3S Research Center / Knowledge Based Systems Group
Leibniz University Hannover
Hannover, Germany
wiese@l3s.de

ABSTRACT

The efficiency of Machine Learning (ML) models has widely been acknowledged in the healthcare area. However, the quality of the underlying medical data is a major challenge when applying ML in medical decision making. In particular, the imbalanced class distribution problem causes the ML model to be biased towards the majority class. Furthermore, the accuracy will be biased, too, which produces the Accuracy Paradox. In this paper, we identify an optimal ML model for predicting mortality risk for Intensive Care Units (ICU) patients. We comprehensively assess an approach that leverages the efficiency of ML ensemble learning (in particular, Gradient Boosting Decision Tree) and clustering-based data sampling to handle the imbalanced data problem that this model faces. We comprehensively compare different competitors (in terms of ML models as well as clustering methods) on a big real-world ICU dataset achieving a maximum area under the curve value of 0.956.

CCS CONCEPTS

• **Information systems → Decision support systems**; Clustering; • **Computing methodologies → Machine learning algorithms**; • **Applied computing → Health informatics**.

KEYWORDS

Machine Learning, Imbalanced Data, Risk of Mortality, Gradient Boosting Decision Tree, Under-sampling, Decision Support System.

ACM Reference Format:
Araek Tashkandi and Lena Wiese. 2019. A Hybrid Machine Learning Approach for Improving Mortality Risk Prediction on Imbalanced Data. In *The 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019), December 2–4, 2019, Munich, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3366030.3366040

*is also affiliated to University of Jeddah, College of Computer Sciences and Engineering, Jeddah, Kingdom of Saudi Arabia.

iiWAS2019, December 2–4, 2019, Munich, Germany

ACM ISBN 978-1-4503-7179-7/19/12…$15.00
https://doi.org/10.1145/3366030.3366040

1 INTRODUCTION

Healthcare is an important area to reap the benefits of advanced Artificial Intelligence and big data. Several Machine Learning (ML) models can be utilized for analyzing big medical data and for implementing an accurate, intelligent medical decision support system. Different intelligent predictions can be achieved such as disease diagnoses, medical prognoses, and individualized medication plans. Accuracy is a crucial requirement for such a system. The selection of an ML model influences system accuracy. Moreover, the quality of the used medical dataset is another influential factor for accuracy and is the main challenge when implementing such a system.

In this paper, we implement and experimentally evaluate a predictive model for assessing the risk of mortality for the Intensive Care Unit (ICU) patients by leveraging the efficiency of the machine learning model. The main challenge we face is the class imbalance in the dataset: In the analyzed mortality dataset the occurrence of suffered (deceased) cases is less frequent than the survived cases. An imbalanced dataset makes the ML model biased towards the majority class. Hence, it is difficult for the model to learn from such data to predict the minority class (which in our use case is the positive class or the class of interest). Therefore, building a model requires efforts to overcome the class imbalance; solving this problem promotes the learning process for the model and optimizes the prediction accuracy.

The approaches to handle this problem are categorized into three categories (according to Galar *et al.* [12]): algorithm level approaches, data level approaches, and cost-sensitive learning methods. The data level approach covers the data sampling methods (to balance the dataset) which can further be divided into over-sampling and under-sampling methods. The algorithm level approach develops an algorithm that adapts to the characteristics of the imbalanced data. The cost-sensitive learning method is a hybrid of both data and algorithm level approaches with different classification costs of the classes.

To overcome the imbalanced class problem our ML model encounters, we develop a hybrid approach based on data and algorithm level approaches. To approve the suitability of our approach, we compare it with other approaches. We validate our approach on a real-world, commonly used medical dataset: the Medical Information Mart for Intensive Care (MIMIC-III) dataset [19]. The main contributions of this paper are dedicated to handling the imbalanced data problem when predicting risk of mortality of ICU patients that incorporates 1) proposing a hybrid approach based on ensemble ML model and under-sampling the dataset 2) developing a method to implement clustering-based under-sampling to balance the data 3)

providing an approach to optimize the performance of K-means++ under-sampling. We conduct an empirical analysis to other models and under-sampling methods to validate the effectiveness of our approach.

This paper is organized as follows. First, we briefly describe the dataset used for our intended prediction task in Section 2. We next provide a comparative analysis of different ML models for the task at hand and choose the best-performing candidate in Section 3. Section 4 provides a description as well as analysis of the proposed clustering-based under-sampling method to handle the imbalanced data. Next, Section 5 presents the performance optimization approach and the accompanying results in conjunction with an in-depth comparison to other clustering candidates. Section 8 provides a survey of related approaches. Finally, Section 9 concludes the paper.

2 THE DATA SET

In this paper, we use the real-world critical care database Medical Information Mart for Intensive Care (MIMIC) which is provided by [19]. The data is collected from patients admitted to critical care units at the Beth Israel Deaconess Medical Center in Boston, Massachusetts in June 2001 to October 2012. It is a publicly available, widely used, and patient privacy protected dataset. We use MIMIC-III which is the most recent and significantly extended data set and thus has not been used much in evaluations so far. MIMIC-III comprises over 61,000 hospital admissions to critical care units of 53,423 adult admissions and 7870 neonate admissions. In our study, we are interested in predicting the risk of mortality for adult patients. Thus, only the data of adult patients (aged 15 years or above) are included. We extracted the high-quality entries that have a sufficient amount of data measurements resulting in a data set size of 32,635 patients. Our selection of predictor variables was inspired by previous work by Lee *et al.* [22]. The predictor variables contain data from the first 24 hours of each ICU stay. We extracted 76 predictor variables including minimum and maximum values of some vital signs from every 6 hours of the 24 hours in the ICU stay and some minimum and maximum lab variables. Most notably, in the extracted MIMIC-III dataset we observe a high imbalance between the negative class (there are 28,974 survived patients) and the positive class (there are only 3,661 suffered patients out of the total 32,635). The ratio of the instances of the negative class to the positive class is hence 89:11. We investigate in this paper the effect of class imbalance on the analyzed machine learning models and compare different approaches to handle the class imbalance.

3 CHOOSING THE BEST ML MODEL

The first part of our investigation is assembling the basic learning model and choosing a good candidate to build the prediction on. The results of a comprehensive comparison are presented in this section.

3.1 Ensemble Machine Learning Model in Comparison to other Models

Different machine learning models are available to achieve our task of mortality risk prediction. It is generally assumed that ensemble models (that are more complex models built up from a set of simpler models) achieve a much better prediction performance. We aim to verify this assumption on our chosen dataset. The following methods are included in our comparison:

- Decision Tree (DT). The DT builds a classification model in the form of a tree. The medical predictor variables are used as nodes for the DT. The different values of the predictor variable constitute the tree branches. The leaf nodes are the value of the risk prediction; in case of a binary classification either yes or no. By following the nodes of the tree starting from the root node to each leaf nodes of a DT, a decision can be taken – in our case, either a patient has a death risk or not. DT haven been used for medical decision making for example in [22].
- Logistic Regression (LR). The task of predicting the risk of mortality can be described as a classification task. A patient might have a risk of death or does not (either 1 or 0). Logistic Regression (LR) estimates the probability that an instance belongs to a particular class (i.e., the probability of mortality risk). The instance/patient belongs to the positive class (has a risk) when the probability is greater than 50% otherwise it belongs to the negative one (has no risk). LR has been used for medical prediction for example in [21].
- K-Nearest Neighbor (KNN). Prediction of the class label for an instance/patient denoted x is based on the training instances. The prediction of the risk of mortality of patient x is derived from the patients that have similar medical records. The user-defined positive integer k identifies the k neighbors nearest to x from which the predicted class of x is assigned. Hence, distances between x and all the training instances are computed. Many related approaches use KNN for different medical predicting purposes such as [18, 26].
- Gradient Boosting Decision Tree (GBDT). Multiple Decision Trees (DT) are ensembled to produce a more robust model. First of all boosting refers to hypothesis boosting that is "any ensemble method that can combine several weak learners into a strong learner" [13]. The popular boosting methods are AdaBoost (short term of Adaptive Boosting) and Gradient Boosting. AdaBoost sequentially builds a better predictor based on the previous one by adjusting (i.e., increasing) the weight of the misclassified training instances. Similar to AdaBoost, Gradient Boosting in every iteration builds a better predictor by correcting its predecessor. However, it is not based on updating the weight of the instances. It is based on fitting to the *residual errors* of the previous predictor. Each additional DT optimizes the classification error of the overall model. Moreover, GBDT reduces bias and variance. More details on Gradient Boosting can be found in [9, 10]. GBDT has previously been used for medical decision making for example in [5].

3.2 Result of the Model Comparison

In order to choose the best-performing model, we compare our candidate ML models – the Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbor (KNN), and Gradient Boosting Decision Tree (GBDT) – on the original dataset (that is, the imbalanced

Table 1: Overview of the model parameters

DT	From the available options for the splittin criterion (information gain, gain ratio and Gini index), we chose the gain ratio as the best splitting criterion. The max depth of 41 is the optimal value experimentally determined from 1 until 100. After testing the minimum samples needed to do a split from the range 1 to 100, the value of 11 is chosen.
LR	For the regularization parameter, we used Lambda of zero.
KNN	We tested the values between 1 to 50 for the number of the nearest neighbors k; k=21 is optimal. The weighted vote approach to combine the class labels outperforms the majority vote for our case. The used distance measure is Euclidean distance.
GBDT	To avoid any over-fitting that GBDT might produce, we use a small number of trees (20 trees) and small maximal depth (5 levels) of each tree.

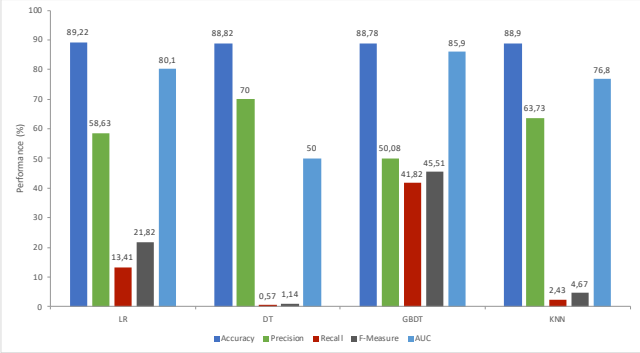

Figure 1: The Prediction Performance of the Models on the Imbalanced Dataset.

data) by different accuracy metrics (see Figure 1). In all cases, 10-fold cross-validation is used for evaluating the models. All the parameters of each model are selected by using a grid search; an overview of the chosen settings for each model is given in Table 1. All the data pre-processing and model implementation, training, and testing were performed in RapidMiner studio version 9.2 in the Educational edition. The computation took place under Windows 10 with an Intel i5-7300U CPU at 2.70 GHz with 32 GB available RAM.

From Figure 1, we notice that all the models have high and similar accuracy. Moreover, we find that the models were more successful in predicting survival cases (the negative class) rather than the suffered cases (the positive class). The reason lies in the imbalanced dataset we use. This problem of imbalanced class distribution causes the classifier to be extremely biased toward the majority class. The high predictive accuracy is a sign for overall prediction of the majority class which is the survived case. This situation – where the higher accuracy metric is not an indicator of an excellent classifier performance – is called *Accuracy Paradox* [31]. It is paradoxical when accuracy is not a good metric for the predictive model because the accuracy is biased to the majority class.

We can conclude from this case that having a highly accurate model is not enough indication of a useful model. In related work, Valverde-Albacete *et al.* [31] state that a predictive classifier model with a low accuracy may have an even higher predictive power than a model with high accuracy. In particular, they stress that this applies to the highly *imbalanced* or *skewed* training data where the classifier produces a highly accurate result by assigning all the cases to the majority class. We can confirm this widely observed phenomenon on our chosen data set. For instance, even though DT and KNN have high accuracy, they have a very low Recall (that measures how often a positive class instance is truly predicted as a positive one). Following the same argumentation, Hoens *et al.* [16] and Chawla [3] state that predictive accuracy is inappropriate when data is imbalanced. They recommend alternative metrics to evaluate the classifier performance on the imbalanced dataset. They recommend ROC curves, Precision and Recall, and F-measure. Furthermore, He *et al.* [15] state that accuracy is sensitive to the class distribution while Precision and Recall are not. Therefore, we should consider other metrics to evaluate our models besides accuracy. In particular, we claim that, regarding the risk of mortality prediction, the cost of misclassification of different classes (the positive and negative class) are different: the cost of misclassifying a patient with mortality risk as not having the risk is higher than misclassifying the healthy patient as having the risk. In other words, the cost of false negative (FN) is higher than the cost of false positive (FP) due to wronglyfully disregarding a patient's severe health state. In evaluating our model, we should hence consider the metrics that reflect this. We conclude that Recall is more crucial for our case than Precision. Rather than considering the overall accuracy of our model we focus on Recall, AUC (area under the ROC curve), and F-measure as classification performance metrics.

Looking at Figure 1, we find that GBDT has the best performance trade-off through all the metrics and the highest AUC. Specifically, for the Recall – which we consider a critical metric – the GBDT has the highest value. Moreover, in the overall comparison, the LR has the second highest Recall and AUC values. The DT gives the random guessing value in AUC and the lowest Recall value. The KNN also has a low Recall and low AUC values in comparison to GBDT and LR.

Hence, in comparison to the other models, the GBDT already gives high predictive performance even without any performance optimization regarding the imbalanced data problem. Furthermore, the GBDT significantly outperforms the DT which confirms the effectiveness of the ensemble technique on the imbalanced dataset. Therefore, we identified GBDT as the best model for our prediction task. Moreover, we hypothesize that GBDT will provide even higher predictive performance with a balanced dataset.

4 K-MEANS++ CLUSTERING-BASED UNDER-SAMPLING FOR IMBALANCED DATA

To overcome the imbalanced class problem, our approach relies on two components: we analyze the GDBT approach (that uses the power of the ML ensemble technique) in conjunction with a

data sampling technique. Hence we now comprehensively test the chosen ML ensemble model GBDT on a balanced dataset to predict the risk of mortality.

4.1 Balancing the Data Set

After choosing an appropriate model, the second part of our approach is identifying a good re-sampling method. As already mentioned, in the used MIMIC-III dataset there are 28,974 survived patients and only 3,661 suffered patients out of 32,635; the original ratio of the instances of the negative class (survived patients) to the positive class (suffered patients) is hence 89:11. We re-sample the imbalanced dataset to balance the distribution of the classes: our goal is to have a balanced dataset with 1:1 ratio of the classes.

Under-sampling refers to the fact that only some instances of the majority class are chosen for the training data set. A significant drawback of under-sampling is that it might remove some useful information. Random under-sampling (that is, choosing training instances from the majority class at random) is one approach for under-sampling the majority class that has this problem. Therefore, the under-sampling has to be done carefully. In our approach we analyze the deployment of clustering algorithms prior to under-sampling; this approach avoids the deletion of important samples that occurs with random under-sampling.

In our first test, we use the common K-means clustering before under-sampling the majority class. To equalize the classes to a 1:1 ratio we select from the majority class samples of the same size as the minority class. Thus, the k value (the number of the clusters) equals the size of the minority class (i.e., k=3,661). K-means clustering under-sampled the majority class (negative class) into k clusters. Then, only the centroids of the clusters are used as representatives for this class. The outcome data are combined with all the instances of the minority class (positive class). The initial cluster centers are determined by using K-means++ algorithm [2] (see Section 6 and Table 3 for details).

4.2 Result of K-means++ Clustering-based Under-sampling

We test the effect of learning from a balanced dataset by ML models for predicting the risk of mortality. Our main focus is on the ensemble learning model (the GBDT) applied on a K-means++ based re-sampled dataset (see Section 4.1). Furthermore, the other ML models are also applied in conjunction with the K-means++ under-sampling to observe their performance as shown in Figure 2. The 10-fold cross-validation is used for evaluating the models and the sampling method.

Comparing the predictive performance of the models on the imbalanced dataset (Figure 1) with their performance on the balanced dataset (Figure 2) we conclude that in general all the performance metrics improved with the balanced dataset. Specifically, the balanced dataset helps to improve the Recall, which is crucial for this study. Thus, the balanced dataset improves the prediction of the minority class (the suffered patient cases). Moreover we can conclude that under-sampling can avoid the accuracy paradox which occurred on the imbalanced dataset. The highest performance improvements were obtained by the GBDT, which outperforms all the other models.

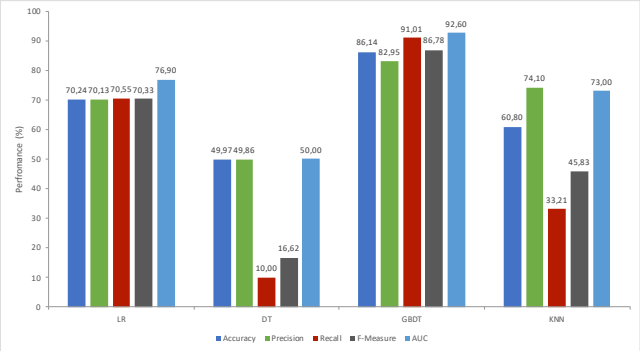

Figure 2: The Prediction Performance of the Models with the Balanced Dataset by K-means++ Under-sampling where k = Size of the Minority Class.

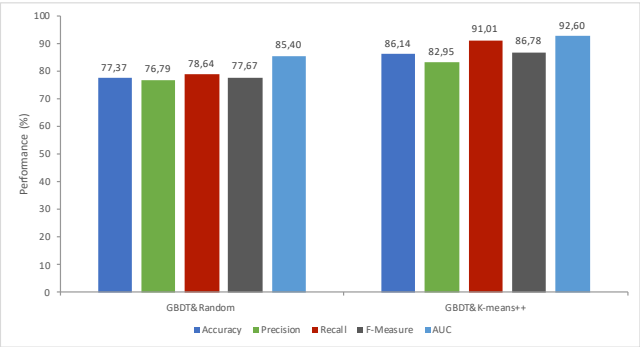

Figure 3: Comparison of GBDT with Random Under-sampling and with The K-means++ Under-sampling where k = Size of the Minority Class.

From the previous tests, we find that GBDT outperforms the other models in both cases – with the imbalanced dataset and with the balanced dataset (i.e., based on K-means++ under-sampling). Thus, the efficiency of the ensemble model GBDT is approved in both cases. Furthermore, we observe that our approach for implementing K-means++ under-sampling to create a balanced dataset significantly lessens the impact of the accuracy accuracy paradox for all the models.

4.3 Comparison to Random Under-Sampling

While fixing GBDT as the best-performing model, we compare our K-means++ under-sampling approach with another under-sampling method: we also tested the GBDT on random under-sampling. In the random under-sampling, the majority class is randomly under-sampled to the same size as the minority class (i.e., 3,661). Then, the remaining majority class instances are combined with all the instances of the minority class to produce a balanced dataset. The comparison result is given in Figure 3.

We can draw the following conclusions. Overall, the K-Means++ clustering-based under-sampling for the majority class outperforms the random under-sampling (see Figure 3). The K-means++ under-sampling improves all the accuracy metrics. The accuracy improved

Table 2: Clustering Run Time for Different Cluster Sizes k.

k	run time
10	2 seconds
100	19 seconds
500	13 minutes
900	40 minutes
1,830	2 hours and 44 minutes

by 11.34% and the Recall improves with K-means++ by 15.73%. K-means++ based under-sampling produces an AUC of 0.926 while the random under-sampling gives an AUC of 0.854; hence K-means++ improves the AUC by 8.43%. The reason for these performance improvements could be ascribed to the main disadvantage of random under-sampling where we lose potentially relevant information from the omitted samples. However, by our K-means++ under-sampling approach, we retain more relevant information of the majority class.

5 IMPROVING THE PERFORMANCE OF K-MEANS++ CLUSTERING-BASED UNDER-SAMPLING

One weakness of the proposed K-means++ cluster-based under-sampling implementation is the long run time to build the clusters. K-means++ clustering groups similar instances of the input data set. The similarity is defined by the distance measure between the instances and the centroid of the clusters. With a large cluster number (we have 3,661) it takes a long time to build the clusters and the centroids (by K-means++ algorithm). The K-means++ clustering of the entire majority class of 28,974 patients on our test system takes 7 hours and 11 minutes.

We hence propose a method to improve the long run time of the previously implemented K-means++ under-sampling (where k equals the size of the minority class). It is based on combining the K-means++ under-sampling (with smaller k) and random under-sampling. We proceed as follows. First, the majority class is clustered into k clusters by K-means++. Then, equally sized subsets are randomly selected from each cluster where the size of these subsets in total equals the size of the minority class (i.e., 3,661). It might happen that the size of a cluster is smaller than the size of the subset that needs to be extracted; in such a case, the remaining number of majority class instances is extracted from the other clusters that have a larger number of instances.

We test different amounts of clusters k = 10, 100, 500, 900, and 1,830 (that is, half of the minority class). The run time was much less than the previous approach (see Table 2); that is, with smaller k we can achieve significant run time improvements.

Apart from the run time, the question arises what effect the amount of clusters has on the prediction performance. We test the ensemble model GBDT with those balanced datasets that are obtained by different k (see Figure 4). The predictive performance of the previous approach (with k=size of the minority class) is higher than this combined approach for any of the tested values of k. However, in comparison to the random under-sampling (Figure 3), the combined approach (for any tested values of k) improves the

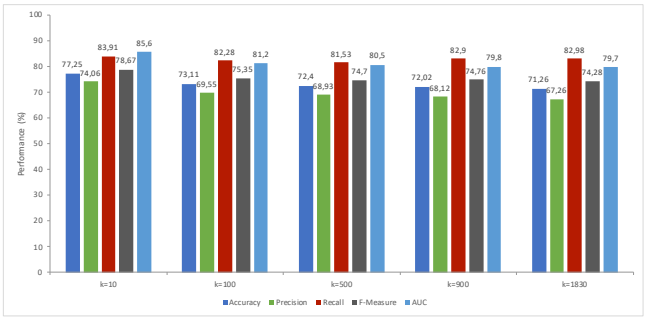

Figure 4: K-means++ Under-sampling with Different Numbers of Clusters.

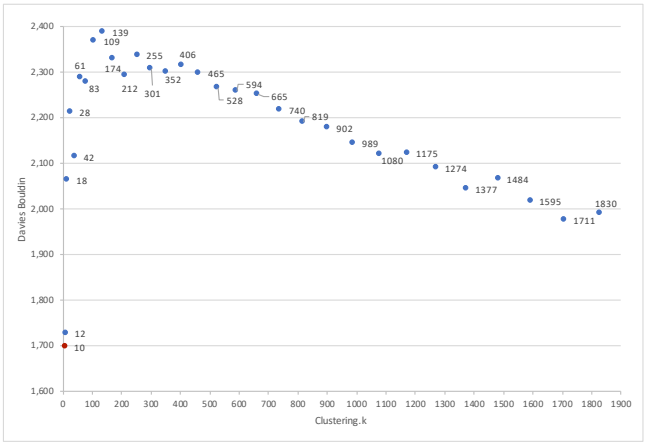

Figure 5: Davies Bouldin index for Different Numbers of Clusters.

recall – i.e., the prediction of the risk of mortality for the patients who are actually in risk.

From Figure 4 we find that number of clusters of k=10 provides the best performance metrics for this study. The value of k=10 optimizes the AUC and F-measure compared to the random under-sampling. We evaluate the clusters that are created from different amounts k of clusters by the Davies Bouldin (DB) index [6]. The DB index is a ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within a cluster is the standard deviation of the distance between the cluster center (centroid) and all the samples of this cluster. The separation between two clusters is the distance between their centroids. The lowest DB index produces the most proper clustering.

The result of the DB index for different k values from 10 to 1,830 is shown in Figure 5. We find that k=10 has the lowest (i.e., the best) DB index. In conclusion, the combined approach between K-means clustering and random under-sampling with k=10 provides the lowest run time and outperforms the random under-sampling performance.

As a result, balancing the dataset by under-sampling has a significant impact on optimizing the model prediction performance of

both classes (the majority and the minority class). The tested under-sampling methods (i.e., K-means++ and random under-sampling and the hybrid of them) improve the performance of prediction on the imbalanced dataset. However, our approach of the standalone K-means under-sampling was the best. The representative selection of the majority class is the major strategical decision of under-sampling. In the proposed methodology of K-means++ cluster-based under-sampling, the k centroids from the k clusters are selected as representatives of the majority class. This approach avoids throwing away the important information of the majority class, which occurs with random under-sampling.

Furthermore, we can conclude that the combination of K-means++ and random under-sampling into a more efficient variant does not outperform the originally proposed K-means++ method. Even though we implement K-means++ clustering for smaller k, the subsequent random representative selection causes information loss. Yet with our study, we were able to identify an optimal k-value that in the combined variant together with random under-sampling improves the run time of the proposed method (because of the smaller k value) and at the same time optimizes the prediction performance in comparison to the pure random under-sampling.

6 COMPARISON TO OTHER CLUSTERING METHODS

Several other clustering methods are available and could be used for under-sampling the data set. For further approval of our chosen approach we compare it with other clustering approaches for under-sampling: K-medoids, DBSCAN, and K-means (without using K-means++ algorithm). We briefly assess the difference of the four approaches:

- K-means: Clustering the data starts with randomly selected k initial centroids for k clusters. Then, a distance measure is used to assign all the data points to the nearest cluster centroid. After assigning all the data points to the clusters, the centroid of each cluster is recalculated by averaging the attributes of data instances of this cluster. This recalculation ends when the centroids no longer change or when the maximum number of optimization iterations is reached.
- K-means++: It is the approach that explained in Section 5. It is based on K-means. The only different between K-means and this method is that the k start points are determined using K-means++. The initial cluster centers iteratively are chosen by taking the distance to the previously chosen cluster centers into account. The drawback of this approach is an increased runtime for this cluster initialization step.
- K-medoids: instead of computing an artificial centroid as the average of all cluster members, a "real" element called medoid is chosen as the representative of the cluster. Points are assigned to clusters such that the distance from the medoid to all the data points in a cluster is minimal.
- DBSCAN [8]: DBSCAN stands for density-based spatial clustering of applications with noise. In contrast to K-means and K-medoids (that is, distance-based approaches where the clustering depends on the distance between the cluster center or centroid and each data point) DBSCAN is a density-based clustering where the clusters are defined by the density of the points. DBSCAN relies on density reachability and density connectivity among the data points. A point P is density reachable by a point Q, if the distance between them is within a distance bounded by a value ϵ. Moreover, the amount of point's neighbors which are within the ϵ distance should be above a specified threshold: the parameter minPts denotes the minimum number of data points to define a cluster. In case no sufficiently sized cluster can be found for a point, this point is considered a noise point. Thus, the two main parameters of DBSCAN are ϵ and minPts.

Table 3: Overview of the clustering model parameters

K-medoids	The cluster number is k=10. The maximal number of runs of the K-medoids with the random initialization of the start points is 1. The maximal number of optimization iterations for one run of K-medoids is 10. The numerical measure used to find the nearest neighbors is Euclidean distance.
K-means	The cluster number is k=10. The maximal number of runs of the K-means with the random initialization of the start points is 1. The maximal number of optimization iterations for one run of K-means is 10. The numerical measure used to find the nearest neighbors is Euclidean distance.
K-means++	The cluster number is k=10. The k start points are determined using K-means++. The maximal number of optimization iterations for one run of K-means++ is 10. The numerical measure used to find the nearest neighbors is Euclidean distance.
DBSCAN	In order to have a similar cluster number for fair comparison we set minPts=5 and ϵ=1.0; this results in 12 clusters of which one cluster consisting only of the noise points is excluded. The numerical measure used to find the nearest neighbors is Euclidean distance.

We assess the two additional models (K-medoids and DBSCAN) in a similar setting as chosen for the runtime-optimized K-means++ as analyzed in Section 5. That is, we obtain a certain set of clusters and randomly choose elements of these clusters to obtain an under-sampled majority class. We start the comparison with a small cluster number; then, if any approach outperforms our runtime-optimized approach in term of accuracy or the runtime, we test it with the same setting of the original approach in Section 4. We represent all the experimentally selected parameters of each clustering model in Table 3.

Regarding the optimal parameters for DBSCAN, we tested different values of the parameters minPts and ϵ. We observed that with smaller ϵ, the cluster of the noise points is getting larger (that is, an extensive amount of data of the majority class is not clustered). For instance, with ϵ=0.1, the entire majority class is considered noise points. We furthermore tested ϵ=0.5 with minPts=5 (which gives 29 clusters, and the noise cluster contains 4,285 instances) as well as ϵ=0.7 with minPts=5 (which gives 17 clusters, and the noise cluster contains 749 instances). Consequently, we selected the value of ϵ and the minPts shown in Table 3 that gives a cluster number (12

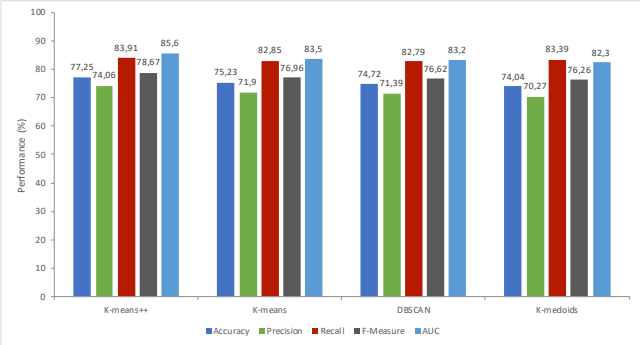

Figure 6: Comparison of our Approach (K-means++) with other Clustering Approaches for Under-Sampling: K-means, K-medoids and DBSCAN.

Table 4: The run time for clustering model for under-sampling

K-medoids	1 hour and 25 minutes
K-means	1 second
K-means++	2 seconds
DBSCAN	1 hour and 13 minutes

clusters including one cluster of noise points) similar to the other clustering models.

We test the ensemble model GBDT with those balanced datasets that are obtained by the under-sampled majority class (by combining each of the different clustering models with random under-sampling) and the entire minority class (see Figure 6). The 10-fold cross-validation is again used for evaluating the models with the different clustering methods. We find that our approach of runtime-optimized K-means++ under-sampling (as described in Section 5) outperforms all the other clustering-based under-sampling approaches. We are ordering the clustering models for under-sampling from the best to the worst: K-means++, K-means, DBSCAN then K-medoids. We noticed how using K-means++ to determine the initial points of the clusters (i.e., as we have done in our approach) improves the accuracy in comparison to K-means where the initial points are determined randomly. The Recall is almost the same between the models (because the minority class the positive cases is the same between the models). However, the other metrics are different among the models. Yet, the difference is not big.

One could argue that the performance of our main approach in Section 4 is significantly higher than the performance of runtime-optimized K-means++ (see Section 5). Hence, we could think of applying a similar approach to the other cluster models (that is, with the amount of clusters k equal to the minority class size). However, we argue that it is not computationally feasible and hence it is not worthwhile to try it. The reason goes back to the computational cost for each clustering models. The run time that each clustering model takes to sample the majority class is shown in 4. We can notice the clear cost-effectiveness of our chosen approach. Even with the small cluster number for K-medoids (10 clusters) and DBSCAN (11 clusters plus one cluster of noise points), the run time was too long in comparison with our approach using K-means++. There is only a small difference between K-means and our chosen approach using K-means++ regarding the run time. However, the resulting accuracy enhancement is worth this small computational cost.

As a result, we find that our approach of runtime-optimized K-means++ outperforms the other clustering-based under-sampling models (K-medoids and DBSCAN) in term of accuracy metrics and remarkably in the computational cost.

7 APPROACHES FOR SELECTING THE MAJORITY CLASS REPRESENTATIVES FROM THE K-MEANS++ CLUSTERS

After having assessed the computation-time performance of K-means++, we now want to optimize the selection of the data instances from the majority class clusters. As stated before the selection of the majority class representatives is a critical point during the under-sampling process. Therefore, rather than randomly choosing such representatives, we consider testing another approach for election – the main idea is based on selecting the nearest neighbors of each centroid.

We cluster the majority class into k clusters. The value k is either equal to the size of the minority class or equal to 10; recall that 10 was approved to be the optimal cluster number for our data in Section 5. Then, for each cluster, we calculate the distance between the cluster's centroid and the cluster's points (by Euclidean distance). Afterward, we select from each cluster the Top1 or TopN nearest neighbors to the centroid. To have a balanced dataset with 1:1 ratio, the selected data points from the majority class should be equal to the size of the minority class. Consequently, the number of the nearest neighbors is the ratio of the size of the minority class to the cluster numbers. In case the cluster numbers equal to the size of the minority class, we select the Top1 nearest neighbor to the centroid. Otherwise, we take the TopN nearest neighbors to the cluster centroids with $N = 366$.

We compare this approach to the previous approaches from Section 4 and Section 5. All the approaches use K-means++ to cluster the majority class for under-sampling, however, the selection of the majority class representatives follows different approaches. The settings of the different approaches are summarized in Table 5. The resulting under-sampled data of the majority class from these different approaches are combined with the minority class before starting the classification. We compare the K-means++ under-sampling method with the different approaches for selecting the majority class representatives on GBDT by 10-fold cross-validation.

Comparing the results in Figure 7, we find that the approach of selecting the TopN nearest neighbors to the centroid in general outperforms all the other approaches. Only the recall of our primary method (selecting the cluster centroids from Section 4) is higher by 3.78%. Predicting the positive class of patient at risk of death is crucial in our case – and hence we consider the approach with highest recall the most appropriate for us. Nonetheless, the approach of selecting the TopN nearest neighbors to the centroid (with a small k) is a great competitor to the method of selecting the cluster centroids. It has a good balance between the accuracy metrics and a short computational time (a few seconds) in comparison to the long time (more than 7 hours) of the approach with highest recall.

Table 5: The different approaches for K-means++ clustering for under-sampling

Approach	Number of clusters (k)	Instance selection
K-means++ and Centroids	k = size of the minority class	Clusters' centroids (method in Section 4).
K-means++ and Random sampling	The optimal k (k=10)	Randomly from each cluster select number of instances = size of the minority class/number of clusters k (method in Section 5).
K-means++ and Top1 centroids' nearest neighbor	k = size of the minority class	Select the nearest neighbor to the cluster centroid.
K-means++ and TopN centroids' nearest neighbors	The optimal k (k=10)	Select the TopN nearest neighbors to the cluster centroid, TopN = size of the minority class/number of clusters k.

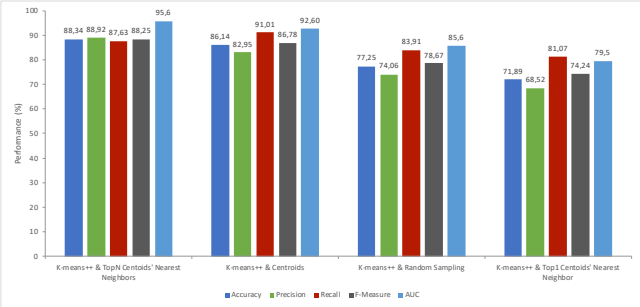

Figure 7: Comparison of K-means++ Under-sampling with Different Approaches for Selecting the Majority Class Representatives.

Unexpectedly, the approach that combines K-means++ and random sampling outperforms the method of choosing the Top1 nearest neighbors. The reason might correspond to the optimal k value that the hybrid method of K-means++ and random sampling used.

As a conclusion, the cluster numbers k and the approach to select the instance are crucial influencers on selecting the majority class representatives (from the K-means++ clusters) and the model accuracy. Selecting the nearest neighbors to the centroids works best with a small k (i.e., we choose TopN). Whereas, with a significantly larger k value selecting the centroids is the better choice than choosing the Top1 nearest neighbor.

In general, all the approaches that used K-means++ clustering-based under-sampling with different methods for selecting the majority class representatives are significantly improving the prediction accuracy of the GBDT on imbalanced data (in comparison to Figure 1). Moreover, they (except the Top1 centroids' nearest neighbor approach) outperform the other clustering models that we tested in Section 6.

8 RELATED WORK

Automated data analytics with machine learning support will play a major role in future medicine. In particular, the notion of predictive, preventive, personalized and participatory (so-called P4) medicine – a term coined by Leroy Hood et al. [17] – is seen as a major paradigm shift from a reactive medicine towards a proactive medicine. Historical data of patients are valuable information in order to assess health conditions as well as decide on further treatment of the target patient. Predictions (based on historical patient data) support doctors in identifying health risks before symptoms become obvious and prepare an appropriate treatment in advance. This results in better plannability of treatments and gives patients the opportunity to prepare for times of illness. In particular, ML methods can act as an enhancement for a Clinical Decision Support System (CDSS). CDSSs have widely been conjectured to identify optimal treatments when considered by experienced medical staff as an extra source of information [25] – in addition to their personal professional expertise.

Applications of ML in medical use cases require a high reliability of the models. In particular, the models have to be able to handle the class imbalance problem. The investigation of this issue is the major focus of our work in this article. To overcome the imbalanced class problem we rely on the data sampling method – specifically, clustering-based under-sampling and ensemble ML. There are many related approaches having applied the clustering-based under-sampling technique to class-imbalanced data. We survey the most significant of them here. Lin et al. [23] proposed a clustering-based under-sampling method based on K-means. They set the number of the majority class clusters equal to the minority class size. Then, the selection of the majority class representatives follows two strategies: using the cluster centers or using the nearest neighbors of the cluster centers. Ofek et al. [27] also used the clustering approach for under-sampling. They aim to consider both computational cost and predictive performance. They cluster the minority class instances then select for each cluster a similar instance number from the majority class. Tsai et al. [30] propose an integrated clustering-based under-sampling method with instance selection algorithms. Kumar et al. [20] use K-means clustering for under-sampling the majority class then use C4.5 as the learning algorithm. Lin et al. [23] used a similar clustering under-sampling approach as us (i.e., K-means and cluster centers are the representatives of the majority class) but they did not empower it by the ensemble ML model. Similarly, [28] apply k-means in conjunction with kNN for text classification. These previous research works mainly focus on the clustering-based under-sampling for pre-processing the dataset; then, the learning process from the data is done by applying ordinary ML models. The high performance of our approach relies on the clustering-based

under-sampling in conjunction with an ensemble ML model. In addition, to the best of our knowledge, our approach is the first one to comprehensively analyze the behaviour of different clustering methods in conjunction with different ML models on the MIMIC-III dataset.

Haixiang et al. [14] and Galar et al. [11] give a survey of the ensemble methods that are used for imbalanced class problem. The ensemble-based classifiers are usually combined either with data re-sampling methods or a cost-sensitive strategy to learn from imbalanced data. When embedding a data pre-processing (re-sampling) technique in an ensemble learning algorithm, each classifier in the ensembles trained with the different (manipulated) training set. For instance, SMOTEBoost combines SMOTE data sampling method with boosting ensemble algorithm [4]. DYCUSBoost integrates dynamic clustering and under-sampling with Adaboost [24]. In our approach (Section 4), we avoid this computational complexity. Rather than including the data re-sampling in each iteration of the ensemble model, it is performed once before applying the ensemble model. Moreover, the achieved result – specifically, the high prediction performance of the critical cases of patients at risk (i.e., the rare cases of the minority class) – is promising. In particular, in the recent review done by Haixiang et al. [14] there are 218 papers that proposed ensemble models for imbalanced data out from the 527 reviewed articles. The only two papers that used GBDT combined it with the cost-sensitive approach. Again, to the best of our knowledge, there is so far no other extensive research proposing and analyzing GBDT with clustering-based under-sampling for imbalanced data.

9 CONCLUSION AND FUTURE WORK

Imbalanced data are a severe problem that affects the Machine Learning (ML) model performance. It causes a biased model and biased accuracy. In this study, we aimed to predict the risk of mortality on an intensive care unit (ICU) data set. The main struggle was the imbalanced data which is a common problem of such a real-world dataset. We proposed a method to mitigate this problem. Our approach is based on implementing an ensemble Machine Learning model on a balanced dataset. The balanced dataset is obtained by K-means++ clustering-based under-sampling of the majority class (to equalize the size of the majority class to the size of the minority class). The centroids of the clusters are selected as representatives of the majority class. This under-sampling method prevents losing critical information from the dataset (which is a central problem of the random under-sampling). The proposed approach provides higher accuracy in comparison to its competitors. Specifically, the hybrid between ensemble ML model and the balanced dataset significantly improves the prediction accuracy of risk of mortality for the patients who are truly in risk.

One limitation of this balancing method is the long run time. We tested a new method to overcome this long run time problem. The method is based on combining the K-means++ under-sampling (of small k) with random under-sampling as well as with nearest-neighbor under-sampling. These approaches with the optimized k value still outperform the basic random under-sampling (without clustering) and overcome the long run time problem by reducing the execution time significantly.

Several options for further analysis and optimization arise. For example, by applying appropriate performance optimization for the clustering (for example parallel execution) a faster execution could be achieved. Feature selection is another step for performance optimization. High-dimensional data cause poor clustering performance. Thus, selecting only the essential features for clustering could significantly improve the performance; we will identify relevant features and analyze their impact in future work. Moreover, we analyzed different notions for the similarity or distance underlying the clustering from a performance perspective [29, 32]. The impact of different distance measures on the overall prediction performance could be analyzed more in detail. Further planned future research, in addition to optimizing the under-sampling approach, considers "hybrid" re-sampling: We for example plan to combine SMOTE for oversampling and K-means++ for under-sampling similar to [1, 7] and identify optimal settings for our use case.

REFERENCES

[1] Astha Agrawal, Herna L Viktor, and Eric Paquet. 2015. SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, Vol. 1. IEEE, 226–234.

[2] David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Philadelphia,USA, 1027–1035.

[3] Nitesh V Chawla. 2009. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*. Springer, New York, 875–886.

[4] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. 2003. SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*. Springer, Berlin, 107–119.

[5] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. 2016. Interpretable deep models for ICU outcome prediction. In *AMIA Annual Symposium Proceedings*, Vol. 2016. American Medical Informatics Association, Bethesda, Maryland, 371.

[6] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* PAMI-1, 2 (1979), 224–227.

[7] Georgios Douzas, Fernando Bacao, and Felix Last. 2018. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences* 465 (2018), 1–20.

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96(34). AAAI, California,USA, 226–231.

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 29 (2001), 1189–1232.

[10] Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis* 38, 4 (2002), 367–378.

[11] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 4 (2011), 463–484.

[12] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 4 (2012), 463–484.

[13] Aurélien Géron. 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., Sebastopol,California.

[14] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* 73 (2017), 220–239.

[15] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering* 21, 9 (2009), 1263–1284.

[16] T Ryan Hoens and Nitesh V Chawla. 2013. Imbalanced datasets: from sampling to classifiers. In *Imbalanced learning: foundations, algorithms, and applications*, Haibo He and Yunqian Ma (Eds.). John Wiley & Sons, New Jersey,USA, Chapter 3, 43–59.

[17] Leroy Hood and Mauricio Flores. 2012. A personal view on systems medicine and the emergence of proactive P4 medicine: predictive, preventive, personalized

and participatory. *New biotechnology* 29, 6 (2012), 613–624.

[18] Mark Hoogendoorn, Ali el Hassouni, Kwongyen Mok, Marzyeh Ghassemi, and Peter Szolovits. 2016. Prediction using patient comparison vs. modeling: A case study for mortality prediction. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2464–2467.

[19] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 9.

[20] N Santhosh Kumar, K Nageswara Rao, A Govardhan, K Sudheer Reddy, and Ali Mirza Mahmood. 2014. Undersampled K-means approach for handling imbalanced distributed data. *Progress in Artificial Intelligence* 3, 1 (2014), 29–38.

[21] Joon Lee, Joel A Dubin, and David M Maslove. 2016. Mortality Prediction in the ICU. In *Secondary Analysis of Electronic Health Records*. Springer, Cham, Switzerland, 315–324.

[22] Joon Lee, David M Maslove, and Joel A Dubin. 2015. Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS one* 10, 5 (2015), e0127428.

[23] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. 2017. Clustering-based undersampling in class-imbalanced data. *Information Sciences* 409 (2017), 17–26.

[24] Chen Lingchi, Deng Xiaoheng, Shen Hailan, Zhu Congxu, and Chang Le. 2018. DYCUSBoost: Adaboost-based imbalanced learning using dynamic clustering and undersampling. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 208–215.

[25] B Middleton, DF Sittig, and A Wright. 2016. Clinical decision support: a 25 year retrospective and a 25 year vision. *Yearbook of medical informatics* 25, S 01 (2016), S103–S116.

[26] Mohammad Amin Morid, Olivia R Liu Sheng, and Samir Abdelrahman. 2017. PPMF: A Patient-based Predictive Modeling Framework for Early ICU Mortality Prediction. *CoRR* abs/1704.07499 (2017), 10.

[27] Nir Ofek, Lior Rokach, Roni Stern, and Asaf Shabtai. 2017. Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* 243 (2017), 88–102.

[28] Jia Song, Xianglin Huang, Sijun Qin, and Qing Song. 2016. A bi-directional sampling based on K-means method for imbalance text classification. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, 1–5.

[29] Araek Tashkandi, Ingmar Wiese, and Lena Wiese. 2018. Efficient In-Database Patient Similarity Analysis for Personalized Medical Decision Support Systems. *Big Data Research* 13 (2018), 52–64.

[30] Chih-Fong Tsai, Wei-Chao Lin, Ya-Han Hu, and Guan-Ting Yao. 2019. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences* 477 (2019), 47–54.

[31] Francisco J Valverde-Albacete and Carmen Peláez-Moreno. 2014. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PloS one* 9, 1 (2014), e84217.

[32] Ingmar Wiese, Nicole Sarna, Lena Wiese, Araek Tashkandi, and Ulrich Sax. 2019. Concept acquisition and improved in-database similarity analysis for medical data. *Distributed and Parallel Databases* 37, 2 (2019), 297–321.