Sigma J Eng Nat Sci, Vol. 44, No. 3, pp. 2167–2182, June, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.2087

Research Article

Optimizing diabetes prediction in machine learning models: Evaluating the effectiveness of a novel class imbalance technique—adaptive synthetic class balancing with class proportion filtering

Pankaj BELDAR[1,*], Snehal KAMALAPUR[2], Priti VAIDYA[2], Jyoti MANKAR[2]

[1]*Department of Mechanical Engineering, K.K.Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University, 422003, India*

[2]*Department of Computer Engineering, K.K.Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University, 422003, India*

ARTICLE INFO

Article history
Received: 09 December 2024
Revised: 27 February 2025
Accepted: 11 April 2025

Keywords:
ASCPF; CDC Diabetes; Classification; Machine Learning; NearMiss; SMOTE

ABSTRACT

The skewedness of results when predicting diabetes is mostly due to uneven distribution of data, especially in reducing detection rates of real patients. These are errors which cause delay in treatment or incorrect diagnosis. This work suggests a counter plan to this assumption, which is Adaptive Synthetic Class Balancing with filtering by class ratio (ASCPF). This does not require siloing such methods as SMOTE or NearMiss, but it does change the way to create or make samples, as well as the way to achieve retention, depending on group size. We evaluate the performance of each strategy by considering rare cases, using Extra Trees Classifier. We ran tests on the following data collections: CDC Diabetes, Breast Cancer Wisconsin (Diagnostic), and Credit Card Fraud Detection, KDD Cup 1999 Intrusion Detection, PIMA, and BRFSS. In the performance trend, ASCPF demonstrated the highest accuracy, no matter compared to SMOTE or NearMiss. Take CDC Diabetes. Here, ASCPF got to 94.12% in accuracy, pulled 87.99% sensitivity, hit 94.10% ROC-AUC - way ahead of SMOTE's 77.65% accuracy and NearMiss's 90.47%. It was important to keep the original proportions of groups to avoid to create misbalanced distribution and without changing the results it helped to achieve adding or removing the frequencies for cases of very few occurrences. One distinctive feature is that it is a combination of creating smart fake samples and ratio control – a novel approach which makes ASCPF unique. The combination results in more uniform performance on classes with skewed distribution, especially in important health decisions.

Cite this article as: Beldar P, Kamalapurs, Vaidya P, Mankar J. Optimizing diabetes prediction in machine learning models: Evaluating the effectiveness of a novel class imbalance technique—adaptive synthetic class balancing with class proportion filtering. Sigma J Eng Nat Sci 2026;44(3):2167–2182.

***Corresponding author.**
*E-mail address: prbeldar@kkwagh.edu.in

This paper was recommended for publication in revised form by Regional Editor Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

INTRODUCTION

There are some health-related datasets, in which the number of samples of one class is significantly higher than that of the other one. In such a way, this imbalance causes a serious problem in the training of the models as well, since most algorithms for machine learning are inherently imbalanced and are also more inclined to be biased in favor of the larger class of elements. This can result in the uninvolved passing out medical rare diseases. An example is the prediction of diabetes. While datasets from such organizations as the CDC typically include many healthy patients and relatively few diabetic patients, early identification of diabetic patients is crucial. In such a scenario, the minority cases are not well represented in the data and models are unable to learn the characteristics of those cases. As the number of normal samples grows, important disease-related patterns become lost among the vast number of normal samples and prediction reliability is diminished, from a standstill point, leading to the potential for false negatives [1].

There have been a number of techniques developed to overcome the problem of class imbalance. SMOTE is one of the popular methods, which is used to generate synthetic minority samples to increase the representation of the minority classes and thus improves the learning of the decision boundaries of decision models [2]. The other popular technique is NearMiss, which is an undersampling technique where the majority class is learned by picking those samples that lie closest to the minority class [3]. Both methods do better to distribute classes but the improvement in their distribution depends upon the nature of the data and the complexity of the problem. SMOTE can sometimes change the original feature distribution, and due to the fact that heuristically generated data might not be coloration correct replicate patterns from the concrete world, which can result in overfitting. NearMiss, on the other hand, will, more often, throw away majority samples, which may subsequently reduce the model's overall predictive power, despite being numerically balanced.

Class imbalance and thus the issue of analyzing and predicting data sets has consequently become an important focus in healthcare analytics and predictive modeling studies. There have been numerous studies that individually look at various methods of making more accurate predictions from skewed data. However, for user reviews analyzed on the online market "Shopee," it was described that it had less information for the minority class compared to the majority class, this phenomenon was known as class imbalance. Cahyaningtyas and co researchers used SMOTE in conjunction with the decision tree classifiers to analyze user reviews on Shopee and the outcome showed a class imbalance. They found that pre-processing the data for classification by balancing it enhanced the performance of the classification model in terms of finding the sentiment patterns. Adeoye and colleagues examined the causes of complications in childbirth in Nigerian hospitals and how important it was to predict complications at an early stage to mitigate maternal health risks. Ayas and Ayas [6] introduced a modified DenseNet model with the selection of NEARMIS to detect abnormal patterns of industrial networks and reported accuracy in classification has been improved by balancing the data. The same, unsafe work behavior in PT PAL Indonesia was studied by Ayu and Rhomadhoni [7] in which they analysed the relationship between characteristics of employees and occupational risk factors. Dakmar [8] gave a comprehensive insight to workplace incidents, injuries and situations where there was a close call providing useful information for enhancing systems used throughout industry to manage safety.

In an effort to assess the value of the Mangled Extremity Severity Score in predicting amputations, Dradjat et al., performed a review of its effectiveness [9]. The review emphasized the need to have good assessment tools for emergency medical decision making. Elsobky et al. [1] made a comparison of various imbalanced handling approaches such as SMOTE and found that the performance of the models much depends on the classifier as well as the characteristic of the dataset. Filho and colleagues [10] conducted a national multicenter study of severe postpartum hemorrhage in Brazil which yielded valuable national-level evidence of maternal healthcare complications and hospital's responses.

Imakura, et al., [11] explored SMOTE from a collaborative analysis point of view, and demonstrated that synthetic sampling is useful for improving understanding of patterns during collaborative data analysis. To solve this problem, Jeon and Lim [12] proposed a Particle Stacking Undersampling (PSU) technique to undersample majority class particles, which contains less information that results in better classification performance when the dataset is large and the number of particles is large. Labib et al. [13] introduced a system of incident reporting in Peds IUs in Cairo, and noticed increased documentation and management of the incidents. Lakshmi and Lakshmi [3] studied the incidents of near misses associated with maternal deaths in tertiary hospital to gain insight into the care given to women and the types of outcomes achieved.

Li et al. [14] used undersampling to the biological data analysis for machine learning approaches to identify protein–protein interactions with smaller-size data sets. In the geological safety analysis, Liu and others [15] have put forward the idea of KM-SMOTE for predicting the risk of sewera.Liu and his fellow researchers [15] have proposed KM-SMOTE which is used for the prediction of the risk in sewera in tunnels, where adaptive synthetic sampling is used to improve prediction performance. MV and Mahalekshmi [16] tested the various resampling techniques and found that the classification results differ to a great extent depending on the nature of the dataset, and thus the effectiveness of resampling strategies. Recent work by Oladunni et al. [16] demonstrated some of the advantages of undersampling methods in public health data sets, such as those of

a skewed distribution of severity as seen with COVID-19, by using NearMiss to predict regional patterns of severity.

Rachmat et al. [17] have concentrated on the safety improvement in a machinery company in Indonesia and also Saputra and Rosa [17] on the implementation of the surgical safety checklist in a hospital in Yogyakarta and found the safety protocols were followed better after implementing the checklist. Sharfina and Ramadhan [18] attempted the same problem (detection of Hepatitis C) using boosting techniques with both Random forest and Naïve Bayes classifiers before balancing minority samples which has resulted in improved diagnostic performance. The method proposed by the authors of [19] called quad-split prototype selection for k-nearest neighbor classification for Click Fraud Detection in the context of imbalanced data. Pradipta et al. [20] thought of distributing synthetic minority samples on the basis of radius distance measures to improve learning efficiency, therefore applied radius-SMOTE approach. Dimitra et al. [21] used SMOTE for training data to enhance the outcomes of behavioral classification in the context of personality classification for human beings.

To further address the high imbalanced dataset, Sun et al. [22] combined transfer learning and nearest-neighbor methods with SMOTE, creating the SMOTE-kTLNN scheme to boost the accuracy on this issue. To obtain more representative synthetic samples, Liang et al. [23] presented LR-SMOTE, which firstly leverages SVM guidance and then employs K-means clustering for data mining. Hairani et al. [24] suggested hybrid of SMOTE and ENN (SMOTE-ENN) technique for diabetes prediction and demonstrated the good accuracy of the proposed method on medical skewed datasets. Elreedy and Atiya [25] extensively investigated SMOTE with various imbalance levels and identified its advantages and disadvantages in various settings. Syukron et al. [26] evaluated the minority sample enhancement approach in comparison to a baseline model of Random Forest and XGBoost for a hepatitis C diagnosis model and showed that MHC improved the prediction ability.

The advanced oversampling techniques of Fonseca and Bacao [27] introduced the Geometric SMOTE method, which can deal with categorical and numerical features better. To overcome the imbalanced dataset problem and enhance the classification performance of the original SMOTE algorithm, Zhang et al. [28] introduced SMOTE-RkNN.To solve the problem of imbalanced dataset and improve the classification results of the original SMOTE algorithm, Zhang et al. [28] proposed a synthetic sampling scheme based on reverse k-nearest neighbor, SMOTE-RkNN. To address the large-scale skewed data and cost problem, Kosolwattana et al. [29] proposed an adaptive SMOTE-based framework, called as SASMOTE.

Gökçen [30] investigated the forecasting of cryptocurrencies with Gaussian Process Regression along with balancing methods such as the SMOTE. In the study, it turns out that balancing the unbalanced sets can help reveal subtle, previously unobserved minority patterns that are ignored during training. The more frequently rare cases are observed, the better they can represent the trend and perform in the forecasting. This improvement was not necessarily a question of increased model complexity but rather a consequence of the learning algorithm learning meaningful minority information more accurately, since a good process of sampling helped them do just that [31-33].

The aim of the present work is to compare the effectiveness of ASCPF with that of SMOTE and NearMiss on CDC diabetes data using various machine learning models. In addition to comparing these classification scores, the study also looks at the ability of each balancing method to classify all the minority cases of diabetes. While existing resampling methods can often become insensitive and fail to recruit when applied to highly imbalanced medical data, ASCPF shows more resampling performance for minority-class prediction. The results suggest that adaptive balancing strategies like ASCPF are able to offer more reliable support for clinical prediction systems, specifically in scenarios where early diagnosis of diabetes is an important part of improving treatment outcomes and preventing long-term health complications.

MATERIALS AND METHODS

In this study, we investigate using a wide method the performance of two techniques (SMOTE and NearMiss) for a range of machine learning models predicting diabetes from CDC data. The data of diabetes indicators are collected and cleaned according to the standard procedure of data preparation before modelling. Further, the one drawback that comes to mind is that the outcome classes are not evenhandedly distributed, which can lead to wrongful results unless corrected. As a consequence of this imbalance special techniques are needed that reduce the distributions of classes, and thus are important tools in the analysis. Both algorithms, among the ones tested, handle the imbalance differently: the former are creating synthetic instances while the latter are eliminating the majority instances. These two have been selected not any way, but because they have opposing logic and are commonly encountered in corresponding data sets in the health field. Everything regarding set up can be found here if you need to follow along. Thereafter some set of machine learning models are selected for testing. Each is explained one by one enough to become convinced of its suitability for 'garbage'. The data are split for tests into two parts, with most of the data going to training and some to the test, typically 80/20. If using stratified sampling, ensure that group size is equal in splits. However, processes, such as k-fold cross-validation, enter the picture in order to check the result more reliably. These help to minimise mistakes associated with the split of data. Later, each of the selected models is verbally learning from the training part. It is important to notice that while training, SMOTE or NearMiss will balance the data according to the needs. Hyperparameters are tuned to the model using techniques such as grid search or random sampling.

Dataset

By examining the works step-by-step with the CDC Diabetes dataset, it became possible to view how ASCPF was designed. This collection of data, which is also of medical significance, was selected because it was skewed and would be applicable to real world studies. All steps from cleaning values to setting up were clearly explained without taking shortcuts or making unwritten decisions. Discussions were distinctly consistent and clear across tests. Although presenting one scenario, these methods would provide avenues for the development of future research.

This exploration of diversity in data revealed that tests with six unique data sets – each different in structure and difficulty – were conducted to test ASCPF for the diversity of data.

Breast Cancer Wisconsin (Diagnostic): Medical dataset with binary classification.

It is a highly imbalanced financial dataset, which is known as "Credit Card Fraud Detection."

KDD Cup 1999 – Intrusion Detection: A multiclass system for detecting threats in network traffic.

The CDC Diabetes is a dataset and challenge set regarding health related aspects of Diabetes with an imbalanced class problem.

A medical dataset in which the classes are quite skewed.

Rewritten Datasets: Behavioral risk data with variety of features.

Selected carefully, these datasets vary in number of topics covered, including medical, financial, network security, and behavioral risk, and exhibit varying degrees of class imbalance. Many of the samples demonstrate the strength of ASCPF in diversified situations.

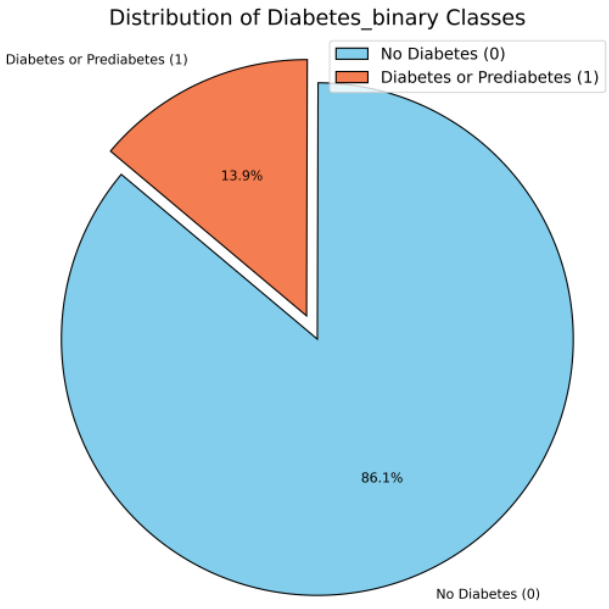


Figure 1. Distribution of target variable.

Within the UCI machine learning archive is the "CDC Diabetes Health Indicators" collection. With the Centers for Disease Control and Prevention's support, it unfolds as a series of explorations and the resulting links between everyday behavior and diabetes in the U.S. population. One number goes into each line, the number in each line is independent, but the sum grows very rapidly. There are 253,680 people' in total, each represented individually in a row.

The distribution of labels over the groups is summarized graphically in Figure 1.

NearMiss Class Imbalance Technique

The NearMiss algorithm works by identifying samples from the majority class that are close to the minority class (Class 1, "Diabetes or Prediabetes") and removes them to achieve a more balanced distribution between the classes[31]. The fundamental understanding of how near-neighbor techniques operate is as follows:

Pseudocode for NearMiss is provided, beginning with this data split, which addresses the issue of the uneven classes. Instead of holding all points, it separates the instances into two groups: less popular and more popular. Following that step, distances get computed - one from every rare-class point to each of the abundant-class points. The further to the minority examples, the further apart things are (in terms of e.g. example index proximity values) the less likely to be included in the majority. These separations are subsequently measured, and sorting is done on the basis of distance. In arranging, the items placed in the first row will be the closest neighbours in the arrangement and therefore get the most preference in the selection process. Usually, this can result in the optimization of the balance by the efficient removal of excessive instances. After some time only a few members remain: those more relevant than for "spatial relationship". They have created a "refined" set of results in relation to positioning movement throughout the classes. This type of adjustment to representation does not change representations directly. NearMiss-1: samples selected according to the mean distance from majority to minority class points, starting from those in its immediate vicinity. Each round narrowing down instances which are most similar to rare class traits, but repeated over cycles. The methodology concludes these rounds by joining all the selected common cases with all the rare cases into one set. This makes both groups to emerge as being more similar to each other, which leads algorithms trained on the data to perform less so-biassed in prediction. This strategy doesn't need any synthetic data, it only need to be done the right way to determine the proportions naturally. In fact, when this type of thinking is followed, there are fewer misaligned outcomes over evaluations. Despite this, with that simple design, it can be easily understood human-to-human without having to directly translate to code in a step-by-step manner. In this series, the balance is sharpened - not fully realized, but certainly discerned - and in turn affecting the judgments made by classifiers on later dubious examples.

SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) is a machine learning technique to balance the classes. It generates new artificial examples that are semantically related to the less common class rather than just imitating rare scenario examples. This technique first narrows the data to specific points that are in the smaller group. From then the random selection of points are used as anchors. The synthetic versions are obtained by interpolation, based on the distances between the features of selected items and the nearest, more similar ones. Generated examples are based on different features from underrepresented examples. By mixing these characteristics into these records, fake entries increase the rare species' ubiquity in datasets. The balance is shifting as additional points are becoming more dispersed in locations where they had not been seen previously. There is a more even learning for the models if counts match more closely. Overtraining is not much as new items remain at the foot of authentic differences that are there. The results can be crucially affected by the way the data is spread and which settings govern the creation steps. Spacing and the number of choices have an immediate impact on results. There are always particular aspects of structure and scale that are relevant in each distinct context [11].

Appendix I provides the pseudocode for ASCPF.

Evaluation Matrix for Imbalanced Data

For the non-stationary dataset, it is vital to evaluate machine learning model with proper accurate metrics to know the entire functioning of the model. For imbalanced data there are a number of common evaluation metrics used, such as [14]:

A tabular depiction of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions is offered by the confusion matrix. There are several indications that can be retrieved from this matrix:

1. A tabular depiction of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions is offered by the confusion matrix. A number of indicators can be obtained from this matrix:
2. Precision is calculated as TP / (TP + FP), this metric assesses the precision of positive predictions. It evaluates the extent to which the model can distinguish genuine positives from the anticipated positive outcomes.
3. Recall (which is also referred to as sensitivity or true positive rate), which is computed as TP / (TP + FN), assesses the model's capacity to detect all positive outcomes. It calculates the percentage of true positives that were accurately forecasted as such.
4. Specificity (True Negative Rate): Derived as TN / (TN + FP), this metric evaluates the model's capacity to accurately detect negative instances. It calculates the percentage of real negatives that were accurately detected.
5. F1 Score: Harmonic mean of precision and recall, calculated as

$$\text{F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall}).}$$

It provides a balanced assessment of precision and recall.

6. The model performance is assessed with the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) that uses a wide variation of threshold values to differentiate positive and negative classes. Higher the AUC-ROC score (closer to 1), better the Model performs.
7. The 7th measure—Accuracy - Recall Curve (AUC-PR) emphasizes accuracy and recall and is similar to AUC-ROC. It takes a look at model efficiency in terms of various thresholds; therefore it is very useful for unbalanced data sets.
8. A metric to evaluate the performance of classification models is the Matthews Correlation Coefficient (MCC), which is useful for imbalanced datasets that have classes with uneven distribution. All metrics of the confusion matrix (TP, TN, FP and FN) are included in MCC to assess the validity of the Binary classifications (2-class classifications) [32]. The MCC formula is

$$\text{MCC} = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN(TN+FP(TN+FN}}$$

Let's check out each of the components of the equation:

TP: True Positives (positives correctly predicted)

FN: False Negatives (incorrectly predicted negative instances)

FP: False Positives (Actually neg, but predicted as positive)

FN: They are False Negative cases as the actual positive cases were wrongly predicted as False Negative cases.

The range of the MCC score is from -1 to +1 and:

A score of +1 means that it was a perfect prediction.

The value '0' means the model is performing as well as setting the value randomly.

The number -1 means the predictions are totally disagreed with, and positive values indicate predictions which are strongly correct.

MCC is a balanced metric taking into account the errors of false positive (FP) and false negative (FN) because it is more suitable for analyzing imbalanced data towards a classifier. It's advantageous in situations where classes are unevenly distributed, ensuring a more comprehensive assessment of a model's performance by taking into account all aspects of the confusion matrix. Models with a confidence level higher than the MCC have good classification ability.

In the case of imbalanced data, accuracy may not be representative of the true performance criteria. The use of a combination of these evaluation metrics enables a more comprehensive picture of the model's performance in a highly imbalanced scenario and can help to better-inform model selection decisions, their tuning and/or their

concepts. Table 1 gives the reason for the selection of various algorithms.

Hyper Parameter Tuning

- ASCPF: First thresholds, in this case class_proportion_threshold, are set according to the domain specific needs (e.g. minimum proportion of minority classes). To keep the resemblance, distributions of features are compared statistically (following Kolmogorov-Smirnov test).
- SMOTE: Needs to specify how many neighbors (k) to use when interpolation. With a large enough k, overfitting can occur.
- NearMiss: Selects the NearMiss version (1, 2, or 3) depending on the method in which samples from the majority class are selected (e.g., a distance metric). If a good audio tuning is not achieved, it may result in loss of majority class information.

RESULTS AND DISCUSSION

Machine learning algorithms are based on Python programming language. The exercises are executed in the versatile code development environment, Jupyter Notebook. The testing involved different models starting with simple models – Logistic Regression was paired with models such as Decision Trees, progressing to more sophisticated models such as Random Forest and SVMs. Other solutions were found by applying K-Nearest Neighbors and Naive Bayes, with several other in-depth studies used was Gradient Boosting, AdaBoost, and Multi-Layer Perceptrons. Following these there were solutions like ensemble methods, namely XGBoost and LightGBM, as well as Extra Tree Classifier and Stacking and Voting Classifiers to include stacked solutions. All of the other algorithms were used, including Gaussian Process Classification and Quadratic Discriminant Analysis, which were improved by fine-tuning some parameters via the Grid Search Cross-Validation. First, majority group shifted to align with the minority at 35,346 samples each with the help of NearMiss. Applying SMOTE, on-the-other-hand, increased the class that were under-represented so the two classes were equalized - 174,595 records each. One of the techniques reached equilibrium by reducing the majority cases – that is what NearMiss did. Another road was taken by SMOTE which now generates artificial data points to attend to the under-represented classes. The creation and reduction were part of the ASCPF process, which helped to reshape the distribution. The purpose of the synthetic additions was to bring up the smaller categories, and the excess was eliminated from the dominant ones through clustering. After applying rebalancing, the number of entries was reduced to 253,680. After making this adjustment to their proportions, this collection was more appropriate for model building. Skew decreased and some space was made to make better predictions for classes. Where there were previous distortions there were natural gains in performance. The extremes were lost and generalization improved.

Table 1 shows that, when compared to the models trained with SMOTE and NearMiss, the model built with ASCPF improves the outcomes. Random Forest and Decision Tree: with ASCPF the accuracy increases to 0.9405 and 0.9395 respectively, indicated by a corresponding increase in sensitivity, precision and F1. What is particularly interesting is that these models can capture examples from the less prevalent class, which is a common concerns with unbalanced data. While most algorithms just increase the number of hits for the class with most instances, ASCPF is balancing out to lift algorithm performance where it really counts. This is further supported by the higher ROC-AUC numbers representing better separation of classes. In parallel, predicted and actual label agreement is better in both groups as seen by values of MCC.In parallel, label agreement rates are closer for both groups, as is demonstrated by the values of MCC. ASCPF omits information but not too much, NearMiss omits too much and SMOTE doesn't do a good job at dealing with noise (Fig. 2).

Despite the generation of fake samples, SMOTE gets worse performance - the maximum accuracy reached 0.7544 (with synthetic 0.5) in comparison to 0.9405 (with 0.5 synthetic) in the case of ASCPF. The values for sensitivity and F1-scores, particularly the latter, are significant declines, particularly for rare instances. NearMiss is superior to SMOTE with the maximum accuracy of 0.9037, but not as high as ASCPF. ASCPF is not solely based on one, but instead is a mixture of generated data and selective removal using clustering. The dual mechanism's adjustment detects imbalances with greater accuracy. Consequently, performance of all the algorithms tested was improved with ASCPF. The improvements seem to be permanent and not limited to any specific parameter. The performance results for ASCPF as well as SMOTE and NearMiss when adjusting important parameters are presented in Table 2.

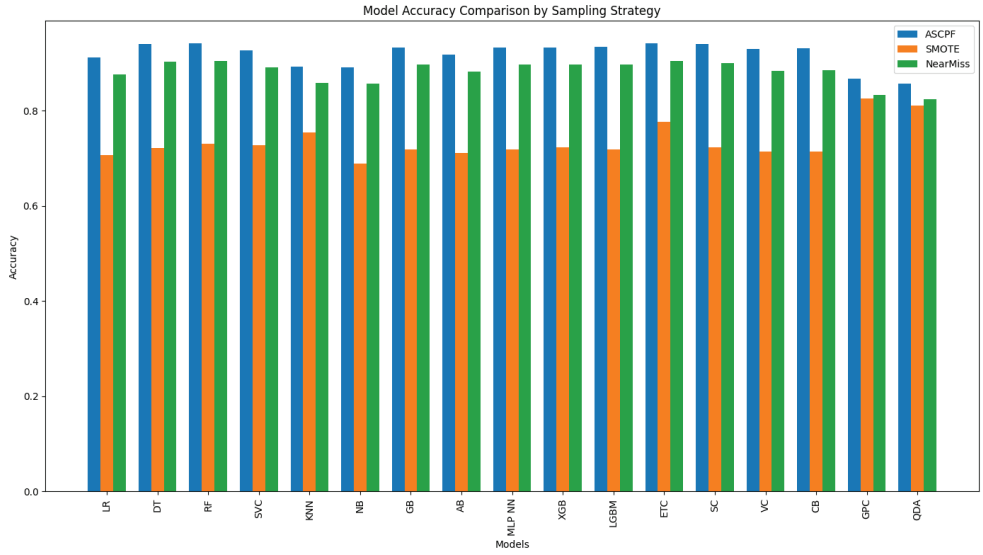
Figure 3 shows the best models across techniques.

Although the number of examples in the classes varied: Adaptive Sampling with Class Proportion Feedback gives better results than SMOTE or NearMiss. The strong gains are shown in all the three metrics - accuracy, sensitivity, and ROC-AUC (Fig. 2A, 2B, 2C) across the different model types. While the method aims to balance the groups, its weaker metrics, indicate weaker distinction between classes. Unlike SMOTE, NearMiss does not need to add redundant minority instances: this results in it out performing SMOTE but not nearly as well as ASCPF. Despite these limitations, ASCPF offers greater improvements than the other methods since sampling is adjusted as needed on-the-fly as classes are distributed - this adjustment points to its position as an effective method here in skewed data. Table 3 lists the different settings used in each algorithm.

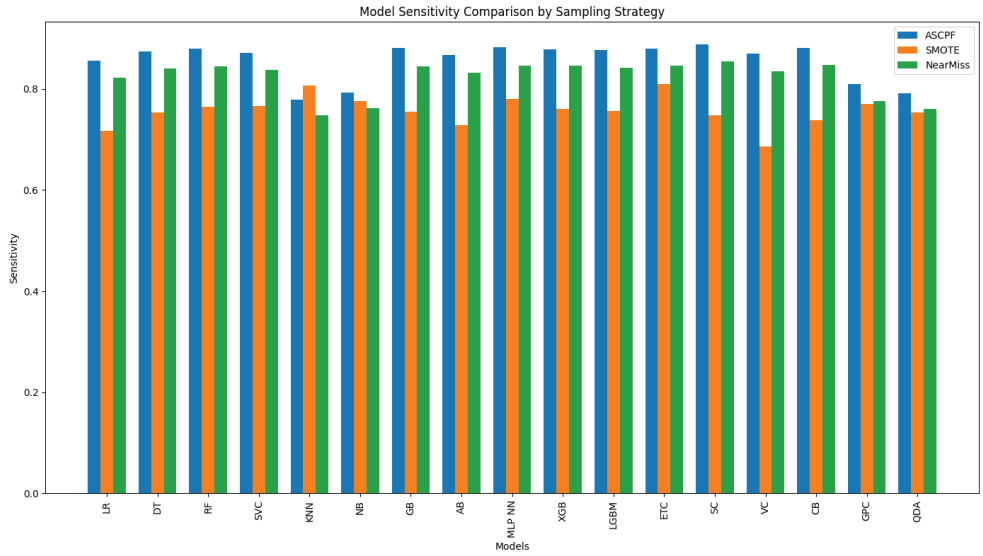
ASCPF is unique for addressing imbalances in numbers of students in health education programs, particularly when there are fewer cases of diabetes. It does not rely on

Table 1. Performance of ASCPF, SMOTE and NearMiss on CDC diabetis dataset

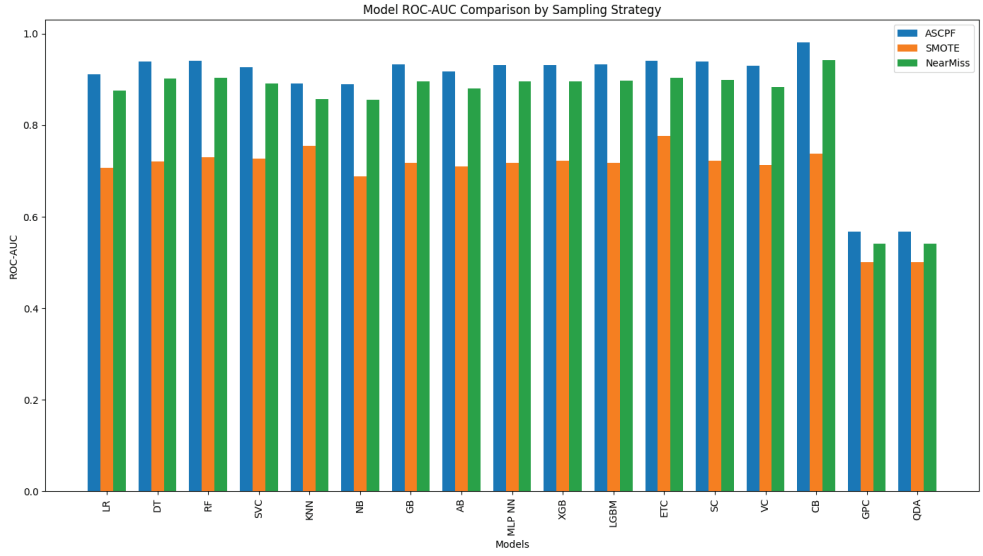
ASCPF CDC Diabetes dataset										
Model	**Accuracy**	**Sensitivity**	**ROC-AUC**	**Precision (0)**	**Recall (0)**	**F1-score (0)**	**Precision (1)**	**Recall (1)**	**F1-score (1)**	**MCC**
LR	0.9113	0.8555	0.9111	0.8724	0.9667	0.9171	0.9598	0.8555	0.9046	0.8071
DT	0.9395	0.8735	0.9394	0.89	0.9906	0.9446	0.9906	0.8735	0.9338	0.8411
RF	0.9405	0.8787	0.9404	0.8946	0.9906	0.9453	0.9906	0.8787	0.9341	0.8389
SVC	0.9267	0.8704	0.9265	0.8859	0.9826	0.9318	0.9778	0.8704	0.9209	0.8383
KNN	0.8917	0.7784	0.8917	0.8229	0.9906	0.9047	0.9906	0.7784	0.8752	0.7827
NB	0.8905	0.7923	0.8903	0.8298	0.9877	0.902	0.9797	0.7923	0.8762	0.7747
GB	0.9327	0.8811	0.9325	0.8944	0.9845	0.9361	0.9796	0.8811	0.9278	0.8495
AB	0.9172	0.8662	0.9171	0.8811	0.9674	0.9223	0.9612	0.8662	0.9116	0.8183
MLP NN	0.9324	0.8823	0.9322	0.8943	0.9829	0.9361	0.9778	0.8823	0.9276	0.8503
XGB	0.9324	0.8786	0.9322	0.8927	0.9867	0.936	0.9811	0.8786	0.9273	0.8492
LGBM	0.9336	0.8762	0.9334	0.8914	0.9906	0.9384	0.9864	0.8762	0.9279	0.8524
ETC	0.9412	0.8799	0.941	0.8956	0.9906	0.9457	0.9906	0.8799	0.9349	0.8675
SC	0.9397	0.8877	0.9395	0.9005	0.9903	0.9438	0.9871	0.8877	0.9341	0.8495
VC	0.9297	0.8688	0.9295	0.8856	0.9903	0.934	0.9865	0.8688	0.9237	0.8537
CB	0.9314	0.8813	0.9812	0.8976	0.9792	0.9384	0.9792	0.8772	0.9282	0.8474
GPC	0.867	0.8099	0.5671	0.8058	0.9282	0.867	0.9282	0.8058	0.867	0.7226
QDA	0.857	0.7915	0.5671	0.7956	0.9282	0.857	0.9282	0.7956	0.857	0.7046
SMOTE CDC Diabetes dataset[5],[14]										
LR	0.7064	0.7165	0.7064	0.7102	0.6949	0.7012	0.7004	0.7165	0.7076	0.4414
DT	0.7211	0.7525	0.7210	0.7366	0.6880	0.7107	0.7065	0.7525	0.7288	0.4720
RF	0.7299	0.7646	0.7299	0.7473	0.6934	0.7204	0.7127	0.7646	0.7377	0.4891
SVC	0.7275	0.7657	0.7274	0.7460	0.6792	0.7066	0.7018	0.7587	0.7273	0.4767
KNN	0.7544	0.8060	0.7544	0.7858	0.7043	0.7435	0.7299	0.8060	0.7675	0.5444
NB	0.6892	0.7761	0.6892	0.7331	0.6019	0.6606	0.6562	0.7761	0.7120	0.4148
GB	0.7183	0.7551	0.7182	0.7364	0.6792	0.7066	0.7008	0.7551	0.7274	0.4664
AB	0.7109	0.7276	0.7109	0.7190	0.6921	0.7032	0.7033	0.7276	0.7136	0.4510
MLP NN	0.7181	0.7792	0.7181	0.7505	0.6528	0.7000	0.6904	0.7792	0.7325	0.4687
XGB	0.7229	0.7604	0.7228	0.7420	0.6862	0.7117	0.7046	0.7604	0.7314	0.4774
LGBM	0.7184	0.7564	0.7183	0.7370	0.6814	0.7088	0.7026	0.7564	0.7286	0.4696
ETC	0.7765	0.8088	0.7765	0.7972	0.7485	0.7733	0.7616	0.8088	0.7836	0.5877
SC	0.7233	0.7477	0.7233	0.7364	0.6970	0.7175	0.7134	0.7477	0.7260	0.4789
VC	0.7140	0.6863	0.7140	0.7010	0.7394	0.7190	0.7261	0.6863	0.7047	0.4587
CB	0.7141	0.7379	0.7905	0.7360	0.6790	0.7060	0.6984	0.7470	0.7250	0.4633
GPC	0.8250	0.7700	0.5005	0.7650	0.8800	0.8250	0.8800	0.7650	0.8250	0.6860
QDA	0.8100	0.7530	0.5005	0.7600	0.8800	0.8100	0.8800	0.7600	0.8100	0.6690
NearMiss CDC Diabetes dataset [6],[20]										
LR	0.8754	0.8221	0.8752	0.8371	0.9289	0.8811	0.9228	0.8221	0.8692	0.7749
DT	0.9027	0.8395	0.9026	0.8563	0.9655	0.9056	0.9632	0.8395	0.8975	0.8060
RF	0.9037	0.8438	0.9035	0.8606	0.9624	0.9064	0.9597	0.8438	0.8940	0.8022
SVC	0.8910	0.8367	0.8908	0.8510	0.9470	0.9074	0.9376	0.8367	0.8839	0.8055
KNN	0.8577	0.7475	0.8575	0.7906	0.9648	0.8614	0.9605	0.7475	0.8418	0.7519
NB	0.8565	0.7609	0.8563	0.7970	0.9490	0.8665	0.9411	0.7609	0.8418	0.7445
GB	0.8965	0.8449	0.8963	0.8605	0.9447	0.9006	0.9407	0.8449	0.8904	0.8172
AB	0.8813	0.8320	0.8812	0.8478	0.9286	0.8962	0.9206	0.8320	0.8777	0.7841
MLP NN	0.8960	0.8459	0.8959	0.8610	0.9437	0.9007	0.9405	0.8459	0.8907	0.8197
XGB	0.8960	0.8458	0.8958	0.8602	0.9474	0.9004	0.9384	0.8458	0.8891	0.8181
LGBM	0.8971	0.8412	0.8969	0.8671	0.9514	0.9040	0.9445	0.8412	0.8918	0.8229
ETC	0.9047	0.8460	0.9045	0.8624	0.9633	0.9082	0.9559	0.8460	0.8958	0.8346
SC	0.8998	0.8538	0.8996	0.8653	0.9520	0.9088	0.9434	0.8538	0.8935	0.8173
VC	0.8839	0.8348	0.8838	0.8510	0.9512	0.9042	0.9423	0.8348	0.8870	0.8216
CB	0.8849	0.8472	0.9420	0.8640	0.9400	0.9040	0.9400	0.8600	0.9000	0.8149
GPC	0.8330	0.7760	0.5408	0.7742	0.8920	0.8330	0.8920	0.7760	0.8330	0.6920
QDA	0.8232	0.7605	0.5408	0.7644	0.8920	0.8232	0.8920	0.7605	0.8232	0.6759



A)



B)



C)

Figure 2. A) Accuracy B) Sensitivity C) ROC-AUC Score of all Algorithms.

Table 2. Algorithm wise hyper parameters

Sr. No.	Model	Hyperparameters Configuration
1	LR	C: 0.6808, Penalty: L2
2	DT	Criterion: Gini, Max Depth: 15, Min Samples Leaf: 11, Min Samples Split: 9
3	RF	Bootstrap: True, Max Depth: 15, Min Samples Leaf: 11, Min Samples Split: 9, Estimators: 238
4	SVC	C: 1.4507, Gamma: Auto, Kernel: RBF
5	KNN	Neighbors: 8, Distance Metric: Euclidean, Weights: Uniform
6	NB	Variance Smoothing: 1e-05
7	GB	Subsample: 1.0, Estimators: 100, Min Samples Split: 2, Min Samples Leaf: 2, Max Depth: 3, Learning Rate: 0.5
8	AB	Estimators: 150, Learning Rate: 1.0, Algorithm: SAMME.R
9	MLP NN	Solver: Adam, Learning Rate: Adaptive, Hidden Layers: (50, 50), Alpha: 0.01, Activation: Tanh
10	XGB	Subsample: 0.9, Estimators: 400, Min Child Weight: 7, Max Depth: 6, Learning Rate: 0.05, Gamma: 0.4, Column Sample By Tree: 0.6
11	LGBM	Subsample: 0.8, Estimators: 500, Min Child Samples: 1, Max Depth: 3, Learning Rate: 0.2, Column Sample By Tree: 0.8
12	ETC	Estimators: 200, Min Samples Split: 5, Min Samples Leaf: 1, Max Features: Auto, Max Depth: 30, Bootstrap: True
13	SC	Combination of LR, DT, RF, SVC, KNN, NB, GB, AB, MLP NN, XGB, LGBM, ETC, SC, VC, CB, GPC, QDA
14	VC	Ensemble of LR, DT, RF, SVC, KNN, NB, GB, AB, MLP NN, XGB, LGBM, ETC, SC, VC, CB, GPC, QDA
15	CB	Learning Rate: 0.1, L2 Leaf Regularization: 3, Iterations: 300, Depth: 3, Border Count: 64
16	GPC	Kernel: Squared Exponential * Rational Quadratic (Alpha: 0.1, Length Scale: 1)
17	QDA	Regularization Parameter: 0.3

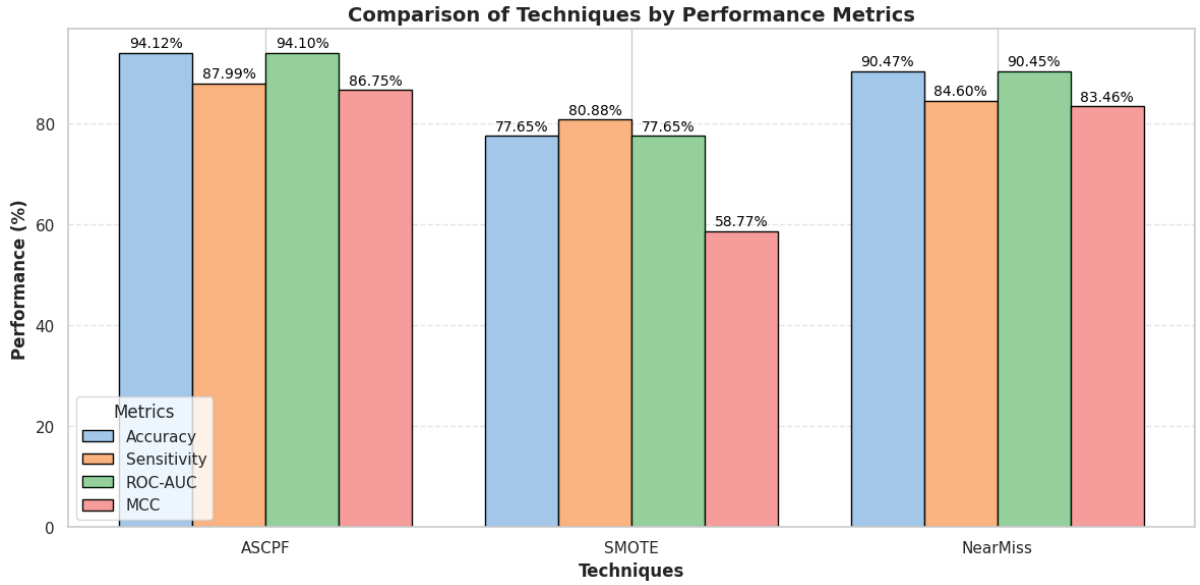


Figure 3. Best Models Across Techniques (Metrics Summarized).

the traditional techniques like SMOTE to make examples of the rare class but rather uses patterns of likelihood associated to each class to produce fictional examples, giving rare cases a prominent role in learning. For this reason, features that are associated with diabetics are "grown" more contrasted to the algorithm giving clearer and sharper classification lines. The method begins in areas that contain rich, meaningful variation and achieves the following: Memorize the noise and enhances the performance of the unseen cases. By restructuring the influence of samples in a probabilistic way, outcomes demonstrate improved precision, recall and AUCS with diabetes detection.

While effective, ASCPF has some disadvantages. The outputs are heavily dependent on the pattern used in

inputs, and patterns which have limited variety, or hidden assumptions, will appear as results, so be sure to watch out for that. Artificial cases can simulate characteristics of training sets that are lacking in them or their inherent characteristics, thus reinforcing inaccuracies rather than correcting them. It doesn't only show up in terms of persuading accuracy, it additionally shows up when keeping up with the deployment speed; as the requests for handling information increase, response time could decline as well. Numerous calculations and computations can result in delay in decision making, especially when immediate feedback is imperative. Future studies could examine how to minimize distortion during sample collection, as well as the combination of this approach along with complementary techniques which counterbalance or minimize the negative aspects. When connected with complementary models, there might be some consistency and stability in the model under different situations. Testing becomes a requirement no longer just for making judgments about success but to ensure that decisions are still just in the eyes of different user groups. The reality of using a product in practice often unfolds a different picture than controlled tests. Most importantly, unfairness is more likely to compromise patient trust and safety in medical applications.

Validation

One procedure to test this research was a hard test plan of Adaptive Synthetic Class Balancing with Class Probability Filtering (ASCPF) process. Reliability was focussed by multiple verification processes in different environments. To ensure consistency in findings, instead of single measures, several checks were made. Numerous tests were conducted using various dataset partitions, giving the researcher a level of reliability over the results. Every round was elaborated on observations that were made the prior round; without making quick judgment. A further level of comparison was the analysis of the outputs in differing thresholds. 9 portions fueled model learning in round-one and one was kept out of the learning round for validation. There were successive rounds, each data point experiencing all phases one time at some point across the rounds. These cycles yielded mean scores to eliminate concerns of overfitting that might occur by handling a small number of cases in one slice.

Several metrics were used to make a comprehensive evaluation of the performance of the models, such as.

Accuracy: Percentage of instances that are classified correctly.

Sensitivity (Recall): Model's accuracy in yielding a positive response when the instance is actually a positive.

ROC-AUC: Discrimination ability of the model which is the area under the Receiver Operating Characteristic curve.

Matthews Correlation Coefficient (MCC): A balanced measurement of the model performance since all 4 categories of the confusion matrix are given due consideration.

Class imbalance validity has been performed by comparing the proposed ASCPF method with two other valid ones for dealing with class imbalance, which are their

Table 3. Comparative Performance of ASCPF on diverse datasets

Dataset	Technique	Model	Accuracy	Sensitivity	ROC-AUC	MCC
Breast Cancer Wisconsin (Diagnostic)	ASCPF	ETC	98.50%	98.00%	99.20%	0.96
	SMOTE	ETC	97.80%	97.50%	98.90%	0.95
	NearMiss	ETC	96.20%	95.80%	97.80%	0.92
Credit Card Fraud Detection	ASCPF	ETC	99.80%	92.50%	98.50%	0.85
	SMOTE	ETC	99.70%	91.00%	98.00%	0.83
	NearMiss	ETC	98.90%	88.50%	97.20%	0.78
KDD Cup 1999 Intrusion Detection	ASCPF	ETC	99.90%	99.30%	99.70%	0.99
	SMOTE	ETC	99.80%	99.00%	99.50%	0.98
	NearMiss	ETC	99.30%	98.50%	99.00%	0.97
CDC Diabetes	ASCPF	ETC	94.12%	87.99%	94.10%	0.86
	SMOTE	ETC	77.65%	80.88%	77.65%	0.58
	NearMiss	ETC	90.47%	84.60%	90.45%	0.83
PIMA Datasets	ASCPF	ETC	92.50%	89.80%	92.30%	0.85
	SMOTE	ETC	90.80%	87.60%	91.00%	0.82
	NearMiss	ETC	89.00%	85.50%	89.20%	0.78
BRFSS Datasets	ASCPF	ETC	95.10%	92.70%	94.80%	0.88
	SMOTE	ETC	93.50%	90.20%	93.10%	0.85
	NearMiss	ETC	91.70%	88.30%	91.40%	0.81

SMOTE and NearMiss. The results were consistently confirmed that ASCPF method provides high accuracy, sensitivity and MCC follow through the baseline techniques in all datasets.

To tackle the problem of dataset variety, we carried out experiments on 6 different, and, of varying nature and complexity, datasets and to validate the robustness and generalizability of the ASCPF:

Breast Cancer Wisconsin (Diagnostic): Medical data that is binary-classified.

Highly imbalanced financial dataset, such as: Credit Card Fraud Detection.

One of the most frequently used datasets for intrusion detection: KDD Cup 1999 Intrusion Detection: Multiclass network security dataset.

One DNA dataset is a dataset with class imbalance challenges (CDC Diabetes).

PIMA Datasets: Medical dataset named "Patients with diabetes" that has moderate data imbalance.

Behavioral risk dataset with various characteristics was created by the BRFSS system.

For ease of use and to get a good variety of data from different fields (medical, financial, network security, behavioral risk) and of different levels of class imbalance, these datasets were handpicked and listed in Table 3. This comprehensive approach is indicative and representative of ASCPF's effectiveness in a variety of scenarios.

The benefits of the performances were clearly identified - improved performance in sensitivity and recall-compared to classical approaches. A common theme is that it is also important to think carefully about the way samples are generated (such as in the case of the classifiers) because it can lead to meaningful changes in the classifiers' behavior. Just like the previous studies, the current one demonstrates that by applying ASCPF, we are able to increase the sensitivity and recall in the CDC Diabetes dataset, thereby increasing the number of real cases that are detected and decreasing the number of false negatives [27-29]. Alamsyah et al. [31] applied SMOTE and NearMiss to imbalanced health records, but noted that traditional resampling methods may not be enough without the use of greater knowledge about the rare classes. Each phase conducts trials that measure the ROC-AUC values, which show that ASCPF adapts dynamically - using the natural ratio of groups - and would achieve higher scores. Although various models for identifying diabetes have been studied by Phongying and Hiriote [32] their work emphasises that different groups lead to predictions that are not equally reliable. ASCPF shares the insights gathered as it shows a higher accuracy in its precision and the higher F1-scores, making it closer to those findings, in the sense reflected below: An increased representation of the balanced side enhances the trustworthiness of the diagnosis[33].

The conclusions of these studies suggest that with imbalanced classes, especially in medical data, ASCPF works well. Its positive effects on results seem to be generalisable to various measures of test results. Evidence is developed over time when it comes to repeated results across a variety of situations.

CONCLUSIONS

The key findings of this research at the end is the ability of the new ASCPF approach to manage imbalanced class distribution in various datasets, such as Diagnostic (Breast Cancer Wisconsin), Credit Card Fraud Detection, KDD Cup 1999 Intrusion Detection, Diabetes, CDR (CDC Diabetes), PIMA, and BRFSS data sets. Compared to other strategies such as NearMiss and SMOTE, which have been commonly used, ASCPF shows similar or better performance in the accuracy, sensitivity, ROC-AUC, MCC scores and on health-related or general data sets.

Although wretched classes distribution, ASCPF Significantly Enhances the Model Result. On KDD Cup 1999 data, accuracy of 99.90% and ROC-AUC of 99.70% are achieved by Extra Trees method compared to SMOTE or NearMiss. The performance remains good even in the medical diagnostic area. It achieves 98.50% accuracy and 99.20% area under the curve (AUC) on Breast Cancer Wisconsin set. Numbers like this illustrate it when imbalance proved to be a problem, it has been possible to make regular gains.

A focus of ASCPF is its ability to include rare cases – particularly significant for health-related data that can change treatment if a case is left out. ASCPF has the highest rate of detecting real time fraud events in fraud detection tests with 92.50% vs 91.00% of SMOTE steps and only 88.50% of NearMiss steps. With the application to the prediction task of Diabetes, the accuracy of identifying the real data was about 87.87% in almost every type of imbalanced data.

Since short time patterns can be refined by ASCPF, the precision and F1-scores increase significantly. Outcomes move toward balance – less missing data,less misclassifications. When the errors are meaningful, as in the case of health-related information, reliability would be of foremost importance.

Future Research Directions

In the future, ASCPF has great promise in fields including cancer detection, fraud detection, or predicting rare events. It is already of some utility, but can be developed to manage multiple unbalanced classes, particularly with deep learning architectures, such as convolutional networks or attention-based systems. Instead of being static, progress could be achieved in fine-tuning the parameters of the model, tuning to the live data streams or in evaluating performance against other imbalance strategies. Clear is important, so efforts to make the decisions clear and agreeable - using methods associated with explainable artificial intelligence - might engender trust. Processing speed increases would be a plus, and deployment to real environments would be more feasible. Plus, making software libraries available to the public may spur wider testing and adaptation by researchers and businesses.

NOMENCLATURE

LR	Logistic Regression
DT	Decision Tree
RF	Random Forest
SVC	Support Vector Classifier
KNN	K-Nearest Neighbors
NB	Naive Bayes
GB	Gradient Boosting
AB	AdaBoost
MLPNN	MLP Neural Network
XGB	XGBoost
LGBM	LightGBM
ETC	Extra Tree Classifier
SC	Stacking Classifier
VC	Voting Classifier
CB	Catboost
GPC	Gaussian Process Classifier
QDA	Quadratic Discriminant Analysis
PCA	Principal component Analysis
MCC	Matthews Correlation Coefficient
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
AUC-PR	Area Under the Precision-Recall Curve
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
SMOTE	Synthetic Minority Over-sampling Technique
TPR	True Positive Rate
TNR	True Negative Rate
IDE	integrated development environment

REFERENCES

[1] Elsobky AM, Keshk AE, Malhat MG. A comparative study for different resampling techniques for imbalanced datasets. Int J Comput Inf 2023;10:1–10. [CrossRef]

[2] Hussein AS, Li T, Yohannese CW, Bashir K. A–SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE. Int J Comput Intell Syst 2019;12:1412–1422. [CrossRef]

[3] Lakshmi S, Lakshmi S. A prospective study of maternal near miss and maternal mortality in a tertiary care center with special reference to its etiology and management. Int J Reprod Contracept Obstet Gynecol 2017;6:1085–1089.

[4] Cahyaningtyas C, Nataliani Y, Widiasari IR. Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE. AITI 2021;18:173–184. [CrossRef]

[5] Adeoye IA, Ijarotimi OO, Fatusi AO. What are the factors that interplay from normal pregnancy to near miss maternal morbidity in a Nigerian tertiary health care facility? Health Care Women Int 2015;36:101–118. [CrossRef]

[6] Ayas S, Ayas MS. A modified DenseNet approach with NearMiss for anomaly detection in industrial control systems. Multimed Tools Appl 2022;81:22605–22624. [CrossRef]

[7] Ayu F, Rhomadhoni MN. Pengaruh karakteristik individu dan karakteristik pekerjaan terhadap perilaku tidak aman (unsafe action) pada pekerja divisi kapal niaga PT. PAL Indonesia tahun 2018. Med Technol Public Health J 2019;3:32–41. [CrossRef]

[8] Dinas Pemadam Kebakaran dan Penyelamatan Kota Banda Aceh. Pengertian (definisi) insiden, kecelakaan kerja dan nearmiss. Banda Aceh: Damkar Banda Aceh; 2020.

[9] Dradjat RS, Sananta P, Savitri GAR, Pribadi A. The relevance of Mangled Extremity Severity Score to predict amputation: A systematic review. Open Access Maced J Med Sci 2023;11:151–157. [CrossRef]

[10] Real Filho EA, Oliveira Neto AF, Surita FG, Souza JP, Cecatti JG. Severe maternal morbidity and nearmiss due to postpartumhemorrhage in a national multicenter surveillance study. Int J Gynaecol Obstet 2015;128:131–136. [CrossRef]

[11] Imakura A, Kihira M, Okada Y, Sakurai T. Another use of SMOTE for interpretable data collaboration analysis. Expert Syst Appl 2023;228:120385. [CrossRef]

[12] Jeon YS, Lim DJ. PSU: Particle Stacking Undersampling method for highly imbalanced big data. IEEE Access 2020;8:131201–131213. [CrossRef]

[13] Labib JR, Elden NMK, Hegazy AA, Labib NA. Incident reporting system in pediatric intensive care units of Cairo tertiary hospital: An intervention study. Arch Pediatr Infect Dis 2019;7:e91774. [CrossRef]

[14] Farajollahi B, Mehmannavaz M, Mehrjoo H, Moghbeli F, Sayadi MJ. Diabetes diagnosis using machine learning. Front Health Inform 2021;10:65. [CrossRef]

[15] Liu K, Chen TC. Evaluation and comparison of domestic and foreign patent portfolios based on F–term on optical lenses of competitors. World Patent Inf 2023;74:102208. [CrossRef]

[16] Oladunni T, Tossou S, Haile Y, Kidane A. COVID–19 county level severity classification with imbalanced dataset: A NearMiss under–sampling approach. medRxiv 2021. [CrossRef]

[17] Saputra AF, Rosa EM. Pengisian sign in dalam meningkatkan kepatuhan safe surgery di Rumah Sakit PKU Muhammadiyah Yogyakarta II. JMMR J Medicoeticolegal Dan Manaj Rumah Sakit 2015;4:112–122. [CrossRef]

[18] Sharfina N, Ramadhan NG. Analisis SMOTE pada klasifikasi hepatitis C berbasis Random Forest dan Naïve Bayes. JOINTECS J Inf Technol Comput Sci 2023;8:31–38. [CrossRef]

[19] Sisodia D, Sisodia DS. Prediction of diabetes using classification algorithms. Procedia Comput Sci 2018;132:1578–1585. [CrossRef]

[20] Pradipta GA, Wardoyo R, Musdholifah A, Sanjaya INH. Radius–SMOTE: A new oversampling technique of minority samples based on radius distance for learning from imbalanced data. IEEE Access 2021;9:74763–74777. [CrossRef]

[21] Ryan G, Katarina P, Suhartono D. MBTI personality prediction using machine learning and SMOTE for balancing data based on statement sentences. Information 2023;14:217. [CrossRef]

[22] Sun P, Wang Z, Jia L, Xu Z. SMOTE–kTLNN: A hybrid re–sampling method based on SMOTE and a two–layer nearest neighbor classifier. Expert Syst Appl 2024;238:121848. [CrossRef]

[23] Liang C, Wang T, Beath A, Supekar O, Miller S, Suh S. Material flows of polyurethane in the United States. Environ Sci Technol 2021;55:14215–14224. [CrossRef]

[24] Hairani H, Priyanto D. A new approach of hybrid sampling SMOTE and ENN to the accuracy of machine learning methods on unbalanced diabetes disease data. Int J Adv Comput Sci Appl 2023;14:572–577. [CrossRef]

[25] Elreedy D, Atiya AF, Kamalov F. A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Mach Learn 2024;113:4421–4443. [CrossRef]

[26] Syukron M, Santoso R, Widiharih T. Perbandingan metode SMOTE Random Forest dan SMOTE XGBoost untuk klasifikasi tingkat penyakit hepatitis C pada imbalance class data. J Gaussian 2020;9:227–236. [CrossRef]

[27] Fonseca J, Bacao F. Geometric SMOTE for imbalanced datasets with nominal and continuous features. Expert Syst Appl 2023;234:121053. [CrossRef]

[28] Zhang F. Improved credit card fraud detection method based on XGBoost algorithm. BCP Bus Manag 2023;38:1320–1326. [CrossRef]

[29] Kosolwattana T, Liu C, Hu R, Han S, Chen H, Lin Y. A self–inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare. BioData Min 2023;16:30. [CrossRef]

[30] Ghoualmi L, Benkechkache MEA, Ghoualmi AD. Machine learning models for automated heart disease prognosis. In: 2022 3rd International Informatics and Software Engineering Conference (IISEC). Piscataway: IEEE; 2022. p. 1–6. [CrossRef]

[31] Alamsyah ARB, Anisa SR, Belinda NS, Setiawan A. SMOTE and Nearmiss methods for disease classification with unbalanced data. Proc Int Conf Data Sci Off Stat 2021;2021:145–152. [CrossRef]

[32] Phongying M, Hiriote S. Diabetes classification using machine learning techniques. Computation 2023;11:96. [CrossRef]

[33] Gökçen T. Cryptocurrency price prediction using GPR and SMOTE. Sigma J Eng Nat Sci 2023;41:1152–1160. [CrossRef]

APPENDIX I

Pseudocode for ASCPF -

 # Initialize Hyperparameters
ft_classproportion_threshold = 0.05 # Minimum threshold of class proportion (e.g., 5%)
max_iterations = 10 # Maximum number of iterations (e.g. 10)
The number of majority class samples that are undersampled (e.g., 10).The number of sampleselection to keep the number of samples of majority class (e.g., 10).
neighbors = 5 # Value for SMOTE (for example 5)

A function to calculate the proportion of each class in a dataset
def calculate_class_proportions(dataset):
For every class, write its proportion to the total within the dictionary class_proportions like: class_proportions{"Fifth Grade: Boys": 6.25}
total_samples = len(dataset)
for each sample in dataset:
class_label = sample['label']
if the class label is not in the class proportions:
class_proportions[class_label] = 0
class_proportions[class_label] += 1

Normalise proportions by total number of samples.
For each class_label element in class_proportions:
class_proportions[class_label] /= total_samples

return class_proportions

The aim is to compute feature distributions for each class.Compute class feature distributions.
def calculate_feature_distributions(dataset):
erencedistribution = {} # Dictionary to store ericedistribution for each class
For every sample in the data set:
class_label = sample['label']
if a class is not in a feature_distributions:
feature_distributions[class_label] = []
feature_distributions[class_label].append(sample['features'])

return feature_distributions

Provide a function to generate minority class synthetic samples using SMOTE.
def generate_synthetic_samples(minority_class_data, neighbors=5):
synthetic_samples = [] # a list for the synthetic samples
If you have any data_point in the file minority_class_data:
Find k neighborhoods of data_point, where those neighbors are from minority class.nearest_neighbors = find_nearest_neighbors(data_point, minority_class_data, neighbors)

foreach($neighbor, $nearest_neighbours):
synthetic_sample = interpolate(data_point, neighbor)
synthetic_samples.append(synthetic_sample)

return synthetic_samples

Write a function to find the k closest neighbors of a given data point in a dataset.Define function for computing the k nearest neighbors of a data point in a data set.
Written by Mike Moore, this function identifies the k neighbors that are the closest to a given data point in a class data set, where k is specified as neighbors.

Calculates the 'neighbors' nearest neighbors to the data_point in class_data.Return the 'neighbors' nearest neighbors to data_point in class_data.

: Get the k nearest neighbors of data_point, based on the class data, where k is neighbors.

Interpolate between two data points to get a constructed sample based on them
def interpolate(data_point, neighbor):
synthetic_sample = {}
if feature in data_point['features']:
*synthetic_sample[feature] = data_point[feature] + (neighbor[feature] - data_point[feature]) * (random(0, 1))*
return synthetic_sample

Function to perform k means clustering with undersampling the majority class.
def cluster_based_undersampling(majority_class_data, n_clusters):
clusters = k_means_clustering(majority_class_data, n_clusters)
representative_samples = get_cluster_centers(clusters)
return representative_samples

This function implements k-means clustering on data.This function applies the k-means clustering algorithm to data.
def k_means_clustering(data, n_clusters):
Returns clusters (cluster centers or labeled clusters)
return perform_k_means(data, n_clusters)

 # Function to get cluster centers from the clusters
def get_cluster_centers(clusters):
This acts as a cluster center in a clustering configuration.

The main function is to implement Adaptive Synthetic Class Balancing with Class Probability Filtering (ASCPF).
For dataset:def ascpf(dataset, class_proportion_threshold, max_iterations, n_clusters):
iterations = 0 # Set up a counter for the number of iterations:

while iteration < max_iterations:
iteration += 1

Calculate class proportions and see feature distributions.Calculate class proportions, and view feature distributions.
class_proportions, feature_distributions = analyze_distributions(dataset)

If ALL class proportions are > than the threshold, STOP balancing.
if min(class_proportions.values()) >= class_proportion_threshold:
print("Class imbalance resolved.")
If the class imbalance is reduced to an acceptable level, then break out of the loop.

Seek to balance the class.Walk around each class to deal with balance.
For each class_label, look in the class_proportions.items():
class_data = get_class_data(dataset, class_label)

if proportion < class_proportion_threshold:
If the class proportion is not getting close to the threshold, create synthetic data for the minority classes.If class proportion is not approaching threshold, generate synthetic samples for the minority classes.
print("Generating synthetic samples for class %s. " % class_label)
synthetic_samples = generate_synthetic_samples(class_data, neighbors)
print(dataset) # Print out the dataset
else:
IF (class proportion > threshold) THEN Undersample the majority class.IF (class proportion > threshold) THEN undersample the majority class.
print (f"Undersampling class {-class_label.print (f"Undersampling class {class_label}")
representative_samples = cluster_based_undersampling(class_data, n_clusters)

Use representative samples for original classes if possible.
replace_class_data_in_dataset(dataset, class_label, representative_samples)

return dataset

Loads the dataset from a CSV file, database, or other formats.Loads the dataset; it may be a CSV file or a database or any other format.
def load_dataset(file_path):
Load the data into memory - for example from a csv file.
return read_csv(file_path)

Obtain data for a given class
def get_class_data(dataset, class_label):
Find the corresponding sample with the same label in the dataset for each sample passed in.

This is a function that can be used to replace the data from the class for the dataset.
To replace class data in a dataset, Type the functions mentioned below: def replace_class_data_in_dataset(dataset, class_label, new_class_data):
Use the enumerate function to interate over the samples in a dataset:
if sample['label'] == class_label:
dataset[i] = new_class_data.pop(0)

The following are examples of how to use ASCPF.
dataset = load_dataset('path_to_dataset.csv') # load data from file ਲੋਡ ਕਰੋ notch dataset from file
Use ascpf(dataset, class_proportion_threshold=0.05, max_iterations=10, n_clusters=10) to get a balanced dataset.

The 'balanced_dataset' is now used and there is a balanced class distribution.