NOVA IMS Information Management School

MDDM

Master's Degree Program in
Data Driven Marketing

Evaluating Ensemble Neural Networks as an Alternative to Tree-Based Ensemble Methods for Heart Disease Prediction Using Oversampling Methods

Emrecan Duran

Master Thesis
presented as partial requirement for obtaining a Master's Degree in Data-Driven Marketing

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

Evaluating Ensemble Neural Networks as an Alternative to Tree-Based Methods for Heart Disease Prediction Using Oversampling Methods

by

Emrecan Duran

Master Thesis presented as a partial requirement for obtaining the Master's degree in Data-Driven Marketing, with a specialization in Data Science for Marketing.

Supervised by

Diana Orghian, PhD, NOVA Information Management School

Flávio Luís Portas Pinheiro, PhD, NOVA Information Management School

November, 2024

Statament of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism, any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisbon, 30th November 2024

Dedication

To my mother,

Your endless love and support have been my guide throughout my life. With sincere gratitude and respect for all that you have given up and for believing in my goals, I dedicate this thesis to you.

With all my love,

Emrecan

Abstract

Ensemble learning enhances predictive accuracy by combining multiple models, but it often struggles with imbalanced data, which can lead to biased results. To address this challenge, this study explores whether Ensemble Neural Networks (ENN) can be an alternative model to tree-based methods like Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB) in predicting cardiovascular disease (CVD) and whether it can provide improved results. Unlike single neural networks, ENN combines multiple neural network architectures, like how tree-based models use ensembles of decision trees. This approach might allow ENN to better capture and understand data patterns. To mitigate class imbalance, oversampling techniques such as Random oversampling (ROS), Synthetic minority oversampling technique (SMOTE), Borderline-smote (B-SMOTE), and Adaptive synthetic sampling (ADASYN) are applied. Performance is evaluated using accuracy, F-score, geometric mean (G-mean), and area under the curve (AUC) on three CVD datasets: Heart Disease Health Indicators, Framingham, and Statlog. Results show that ENN, when combined with SMOTE and B-SMOTE, offers a strong alternative for imbalanced classification tasks, though tree-based methods remain more robust in terms of overall performance.

Keywords: imbalanced learning; oversampling; tree-based ensemble algorithms; ensemble neural networks

Sustainable Development Goals (SDG):

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AB	Adaptive Boosting
ADASYN	Adaptive Synthetic Sampling
AI	Artificial Intelligence
AUC	Area Under the Curve
B-SMOTE	Borderline Synthetic Minority Oversampling Technique
CVD	Cardiovascular Disease
DL	Deep Learning
ENN	Ensemble Neural Networks
G-mean	Geometric Mean
GB	Gradient Boosting
ML	Machine Learning
NN	Neural Networks
RF	Random Forest
ROS	Random Oversampling
SMOTE	Synthetic Minority Oversampling Technique
XGB	Extreme Gradient Boosting

1 Introduction

In recent years, machine learning has revolutionized numerous fields by enabling systems to make accurate predictions and uncover patterns in data (Jordan & Mitchell, 2015; Goodfellow, Bengio, & Courville, 2016). Among modern machine learning techniques, ensemble learning has proven to be one of the most effective methods. What makes these methods powerful is combining the predictive capabilities of several models to enhance overall performance (Dietterich, 2000; Rokach, 2010). However, one of the challenges these powerful methods face is handling data where the class distribution is imbalanced. An imbalance in data refers to a situation when the representation of one class is noticeably lower than others. Such data tends to prioritize the majority and ignore the minority group. Therefore, the presence of imbalance can produce biased results and inaccurate predictions in practical machine learning (ML) scenarios (Susan & Kumar, 2021, pp. 1–2). Medical diagnoses, anomaly detection, and fraud detection are some examples of critical events that are rare in nature. In such areas, achieving a balanced data distribution is crucial to ensure accurate and reliable results (Kaur et al., 2019).

The area of interest of this study cardiovascular disease (CVD) is one of the fields where medical diagnoses are applied. CVD is used to point out any condition that influences the heart and blood arteries. This general statement covers all heart diseases (National Heart, Lung, and Blood Institute, n.d.). Over time, due to its various symptoms, CVD becomes a public concern, causing 18 million deaths each year, as stated by the World Health Organization. Key factors contributing to CVD include unhealthy eating, not exercising, and smoking (World Health Organization, n.d). In the last century, CVD has transformed from a common disease in developed countries to a leading global health problem. Firstly, CVD started to be seen in Western developed countries due to lifestyle. Afterward, it came out as the main reason for early death worldwide, especially affecting low-income countries. Lack of financial resources, changes in lifestyle, and limitations on proper treatment in low-income countries have a big impact on the transition (Teo & Rafiq, 2021, p. 733). Especially in third-world countries, widespread heart diseases cause increased costs and the possibility of receiving late results. This urgency pushes healthcare

institutions to innovative approaches such as ML for early diagnosis and cost reduction (Javaid et al., 2022). In addition to ML applications that form the foundation of this study, further strategies could be pursued to make health services and tools accessible worldwide such as transferring the technologies from high to low-income countries with collaboration and partnerships and making the prices affordable for required tools. Bringing these into action will also speed up the achievement of Sustainable Development Goal 3.4, which aims for global health and well-being (World Heart Federation, 2023).

The demand for accurate predictive models to diagnose rare diseases underscores the importance of developing models that can reliably identify all relevant classes. Such models are essential for providing precise and efficient results in ML applications. To this end, various resampling strategies are applied to mitigate the impact of imbalanced data. One of these strategies is oversampling which involves creating samples for the minority class to ensure a more balanced distribution. In contrast, the undersampling strategy removes samples from the majority class, which may result to a loss of important information but helps balance the dataset; it is particularly beneficial when the dataset is large. Another strategy is combining both oversampling and undersampling, aiming for optimal class distribution without significant data loss. Advanced techniques like cost-sensitive learning make the algorithm give more weight to errors made on the minority class, reducing the chances of misclassifying it. This increases the algorithm's sensitivity to minority instances and improves its performance in situations with severe class imbalances. To deal with the imbalanced data in this study, the oversampling strategy will be applied to increase the representation of minority samples without discarding any valuable data from majority samples. Random oversampling (ROS), Synthetic minority oversampling (SMOTE), Borderline-smote (B-SMOTE), and Adaptive synthetic sampling (ADASYN), which will be used in this study, are some of the commonly applied oversampling strategies to tackle class imbalance effectively. These strategies are either generating synthetic samples or replicating existing minority class instances to obtain balanced data.

As well as employing and emphasizing the importance of oversampling strategies in ensemble learning, this study's main objective is to determine whether there is an alternative model that works in the same way as ensemble tree-based models and produces better results. This alternative model is an ensemble neural network (ENN). The idea behind of the model is to integrate multiple neural networks similarly to how ensemble models combine several distinct models. In this way, ENN might handle data patterns more effectively and achieve a more reliable prediction model by utilizing the strengths of different neural network architectures. The well-established ensemble tree-based models that will be compared to the ENN in this study are Random Forest (RF), Adaptive Boosting (AB), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB).

While the corresponding models are versatile enough to be used in multi-task classification or regression tasks, this study will focus specifically on their application to binary classification. The baseline performance of models will be evaluated on imbalanced data by using CVD benchmark datasets from Kaggle and UCI Machine Learning Repository with different imbalance ratios, number of samples, and features. Following preprocessing steps and applying oversampling techniques, GridSearchCV will be used to determine the optimal hyperparameter combination. To provide a comprehensive evaluation of model performances, various metrics will be employed, including accuracy, F-score, geometric mean (G-mean) and, area under the receiver operating characteristic Curve (AUC) scores.

Applying this methodology, this study evaluated the effectiveness of ENN in addressing class imbalance in CVD prediction across several benchmark datasets. While ENN demonstrated strong potential, particularly in balancing precision and recall, it generally did not outperform tree-based models in terms of overall performance. However, ENN, when combined with effective resampling techniques such as SMOTE and B-SMOTE, showed notable improvements in handling class imbalance, making it a competitive alternative in managing imbalanced classification tasks, particularly in the Heart Disease Health Indicators dataset.

2 Literature Review

2.1 Machine Learning: Foundation of Artificial Intelligence

Artificial intelligence (AI) made significant progress and become widely applicable in today's age, driven by technological advances and the increase in the amount of data available. Digital advertising (Lee & Cho, 2019), fraud detection (Ryman-Tubb, Krause, & Garn, 2018), recommendation systems (Aher & Lobo, 2013), autonomous vehicles (Aradi, 2022), spam detection (Crawford et al., 2015), and many other technologies are already integrated into our lives. Even though these technologies often operate invisibly in the background of our devices and activities (Helm et al., 2020), their impact is substantial.

With the increasing integration of AI technologies into everyday activities, the role of ML in analyzing and interpreting complex data continues to grow. ML enables computers to learn from data, tackling problems too complex for direct human interpretation, and is crucial for extracting valuable insights and improving decision-making. Although ML is key to developing and optimizing algorithms, it is important to explore various methods to achieve optimal results. No single approach fits every scenario; the determination of method depends on the specific scenario and objectives.

While the importance of ML grows in handling complex data, understanding its fundamental categories becomes essential for effectively applying these techniques to different problems. ML can be broadly categorized into two types based on the nature of the problem: supervised and unsupervised learning. In the context of supervised learning, both input features (independent variables) and target labels (dependent variables) are required. The goal is to develop a relationship from inputs to outputs, enabling the model to predict target labels using the input features in the dataset.

To deepen the understanding of supervised learning, it is essential to explore its primary subcategories: classification and regression. Classification problems involve predicting discrete outcomes and are categorized into binary and multi-class classification. For example, this study

utilizes binary classification, where the target labels are either 1 (the presence of heart disease) or 0 (the absence of heart disease). On the other hand, regression problems are concerned with predicting continuous variables, such as housing prices or weather conditions.

In contrast, unsupervised learning does not rely on predefined target labels. Instead, it is used to uncover patterns, relationships, or structures within the data without specific outcome variables. For instance, clustering customers into categories such as bronze, silver, and gold is an example of unsupervised learning, as it involves grouping based on inherent patterns rather than predefined labels.

In supervised learning, various individual models are often employed, and the one that delivers the optimal results for the specific task is selected for future predictions. Expanding on this concept, ensemble learning takes this approach further by combining the outputs of several models. Ensemble learning is a broad term for methods that combines predictions from several different separate models, known as base learners, to make a final decision. Ensemble methods typically use base learners, each of which is a single model, and the combination of these models aims to enhance overall performance over using a single model. This method also helps to improve the strength of the model and generalization ability (Sagi & Rokach, 2018, pp. 1–2).

Bagging: Bootstrap Aggregating

Bagging (Bootstrap Aggregating) follows principles of ensemble learning by using multiple base learners. It creates multiple subsets of training data through bootstrap sampling, where random samples with replacement are extracted from the original dataset and processed in parallel (Dong et al., 2020, p. 242). Training each subset independently produce different results in the prediction phase which are then aggregated (e.g., averaging or voting). An extension of bagging, Random Forest, utilize several decision trees as base learners.

Random Forest. The concept of random forest (RF) builds on the earlier work of Tin Kam Ho on random decision forests in 1995 (Ho, 1995), with the first formal paper on RF presented by Breiman in 2001 (Breiman, 2001). RF stands as one of the established algorithms in ensemble learning that use the characteristics of bagging. It also uses bootstrap sampling, like bagging,

and incorporates random feature selection. This indicates that each tree in the forest is build using a randomly selected subset of training data and a subset of features. And each tree is divided into nodes depending on the selected characteristics. This division is determined by the Gini impurity or entropy, which are measures used for classification problems. The process of splitting nodes continues until the maximum depth is reached or there is no sufficient data left in the leaf. Each tree votes for the class. The most voted class becomes the final prediction (Biau & Scornet, 2016).

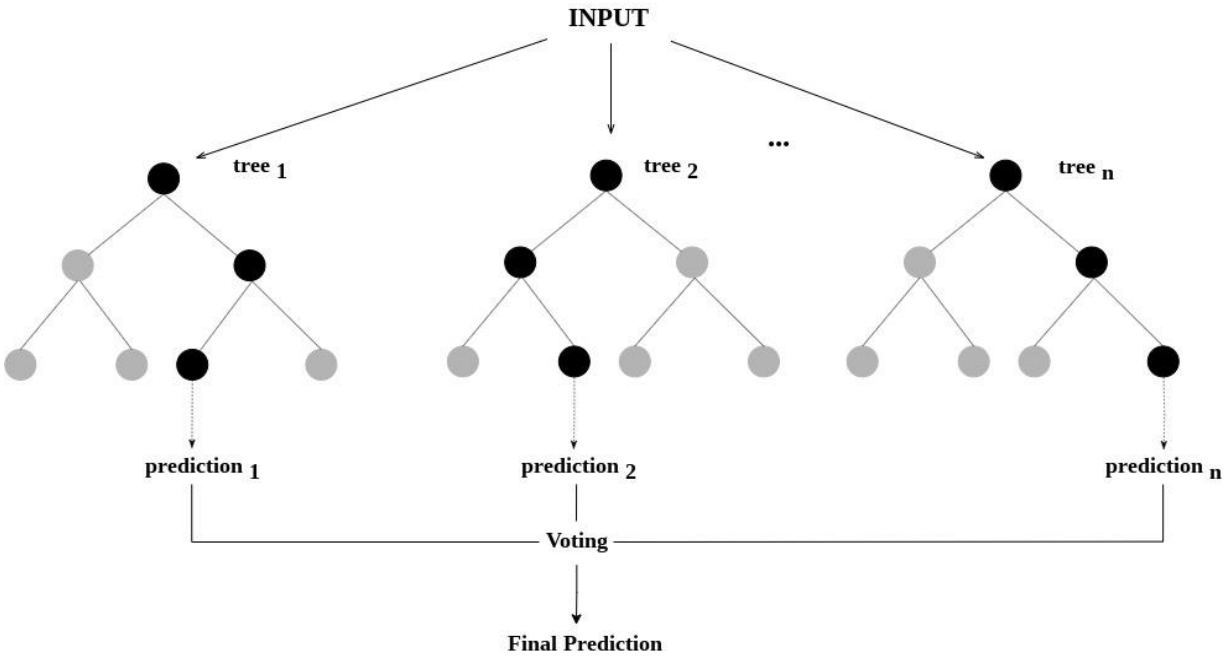

Figure 1 - Diagram of the Random Forest Algorithm

Note. The diagram illustrates how a RF model operates by combining multiple decision trees to make a final prediction in classification task. Each decision tree within the forest independently predicts a class label for the input data. The predictions from all the trees are then aggregated using a majority voting system, where each tree votes for a particular class. The class receiving the most votes from all trees is chosen as the final prediction.

Boosting

Unlike bagging, boosting strategy doesn't run in parallel; instead, it builds models sequentially. In this sequential approach, each model gives more importance to samples incorrectly classified by previous models. While more weight is given to incorrectly classified samples, correctly classified examples are given less weight. This leads to improved performance of weak learners (Schapire, 1990).

Adaptive Boosting. Robert Schapire and Yoav Freund proposed the AB in 1996. AB aims to create stronger and more accurate models by training multiple weak learners sequentially like boosting strategy. The point where it differs from boosting algorithms is to give more weight to misclassified or difficult-to-classify examples at each iteration. The algorithm dynamically sets both the weights of individual examples and the associated weak learners. The sequential process continues until a strong model is developed, usually identified by a defined quantity of weak learners (Wyner et al., 2017).

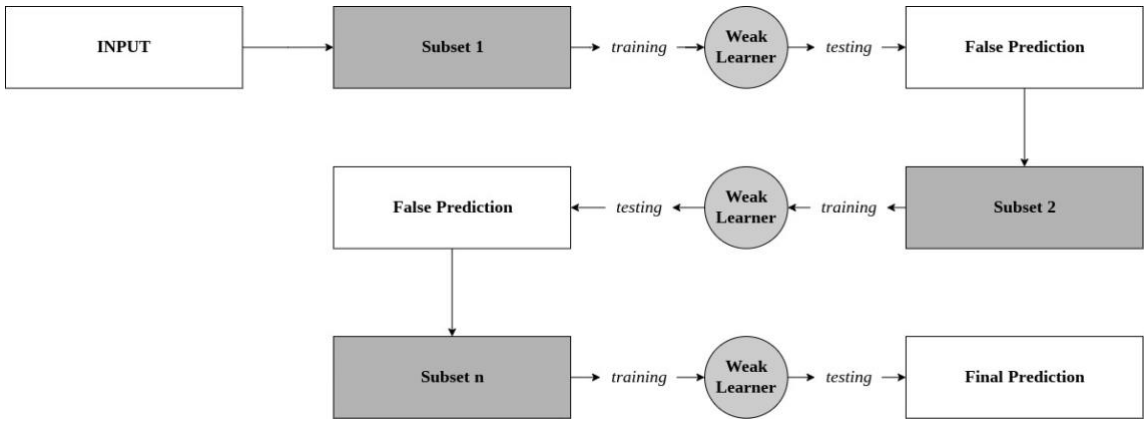

Figure 2 - Diagram of the Sequential Boosting Strategy

Note. In boosting, input data trains a sequence of weak learners, each focused on correcting the errors of the previous ones. More emphasis is placed on misclassified data points as the process continues. The final prediction is a weighted combination of all learners' outputs.

Gradient Boosting. Stanford University Professor Friedman first presented Gradient Boosting in 1999 (Friedman, 1999). Gradient Boosting aims for a strong model by using multiple weak learners like AdaBoost, but there is a difference between these models. GB uses loss function to minimize the errors. The loss function measures the residuals which are the differences between predicted and actual labels in the training data. At the beginning of each iteration, a weak learner is added to the ensemble and fitted to the residuals. In the optimization process, the parameters of this weak learner are adjusted to reduce the overall loss. Every weak learner contributes to the decrease in the prediction error throughout each repetition of this procedure (Zhang et al., 2019).

Extreme Gradient Boosting. The first paper was proposed by Tianqi Chen and Carlos Guestrin in 2016 (Chen & Guestrin, 2016). XGB aims to create a stronger and more accurate model by training multiple weak learners sequentially. The focus is to minimize the model's errors at each iteration. It applies random feature selection to increase diversity and helps to avoid overfitting by applying shrinkage (learning rate) and regularization. It stops the training when no improvement is shown or reaches the maximum number of rounds given.

The final output for both models will be the predicted probabilities of two classes. In binary classification, to convert the probabilities into class labels, the threshold is used as a common approach. For example, if the threshold is 0.6 and the predicted probability is 0.7, the model will be assigned to a positive class.

Additionally, stacking can be applied as more advanced method. Stacking, also known as stacked generalization, is a method that brings together predictions from several models by training another model on their outputs. While bagging and boosting use the same type of model for each base model, stacking uses completely different models. Having explored bagging and boosting methods, ENN and the NN that form its foundation will now be examined.

Deep Learning: Neural Networks

In 2006, deep learning (DL) emerged as a novel area within the field of machine learning. The main principle of DL is nonlinear processing. In this process, deep neural networks apply a

nonlinear transformation to the previous output layer and take it as input in the next layer, so it's creating a hierarchical method. This hierarchical method is used among the layers to measure importance of the data and to understand the relationship within the data (Sarker, 2021, pp. 1-5).

DL applies multi-layer neural networks to understand the data better. Then, it uses backpropagation to correct mistakes in its predictions to improve over time (LeCun et al., 2015, p. 436). Thanks to the hierarchy and backpropagation, DL models can find complex patterns with high generalization such as image classification and face recognition (Ye, J. et al., 2017, pp. 2545-2546). In binary classification tasks, typically, neural networks utilize the binary cross-entropy loss function to measure the difference between predicted probabilities and actual binary labels. Through backpropagation, the loss function guides the model to modify the weights and bias parameters in hidden layers. The model helps the predicted probabilities more closely match the actual labels by repeatedly adjusting these parameters throughout training to minimize the loss function.

Following, by using the sigmoid activation function, the outputs are converted into probabilities that are between 0 to 1. These probabilities show the possibility of being in the positive class. While this process goes on, neurons in hidden layers use nonlinear transformations for the input data to slowly extract important features and patterns. This process makes neural networks differentiate between the two classes effectively.

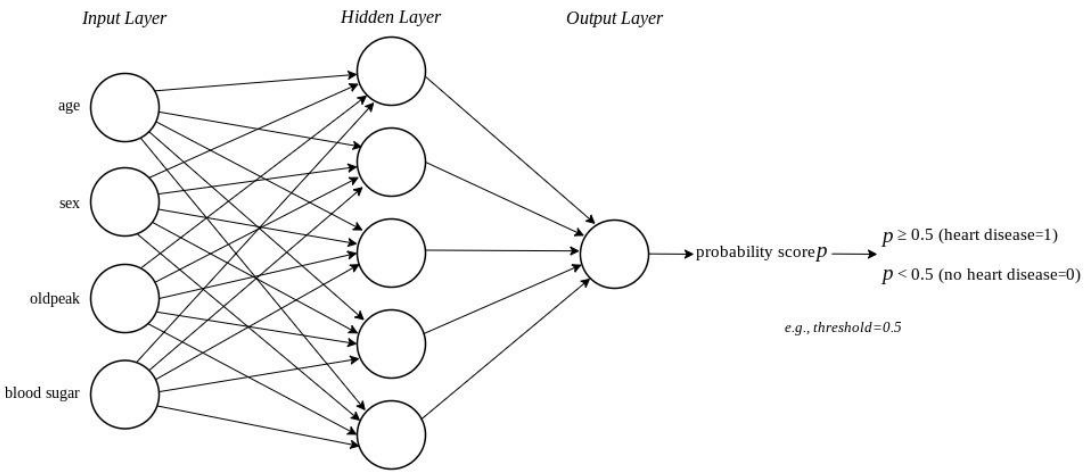

Figure 3 – Workflow of Single Neural Networks Architecture in the Heart Disease Domain

Note. The diagram shows a NN model designed to predict the probability of heart disease based on various input characteristics such as 'age', 'sex', 'oldpeak' and 'blood sugar'. The network consists of three layers: the input layer, which contains neurons corresponding to each feature; the hidden layer, which processes the inputs to identify patterns; and the output layer, which produces a probability score stating the presence of heart disease. If the score is 0.5 or higher, the model predicts the presence of heart disease; otherwise, it predicts the absence of heart disease.

Ensemble Neural Networks

The idea behind ensemble neural networks is to train several neural networks independently on the same dataset but with different initializations and architectures. Each neural network within the ensemble can have different aspects of the data or learn different patterns due to its unique setup (Zhou et al., 2002, pp. 241–242). Through aggregating the predictions of these models, ENN might mitigate individual model biases and errors, leading to improved generalization performance and improved overall accuracy.

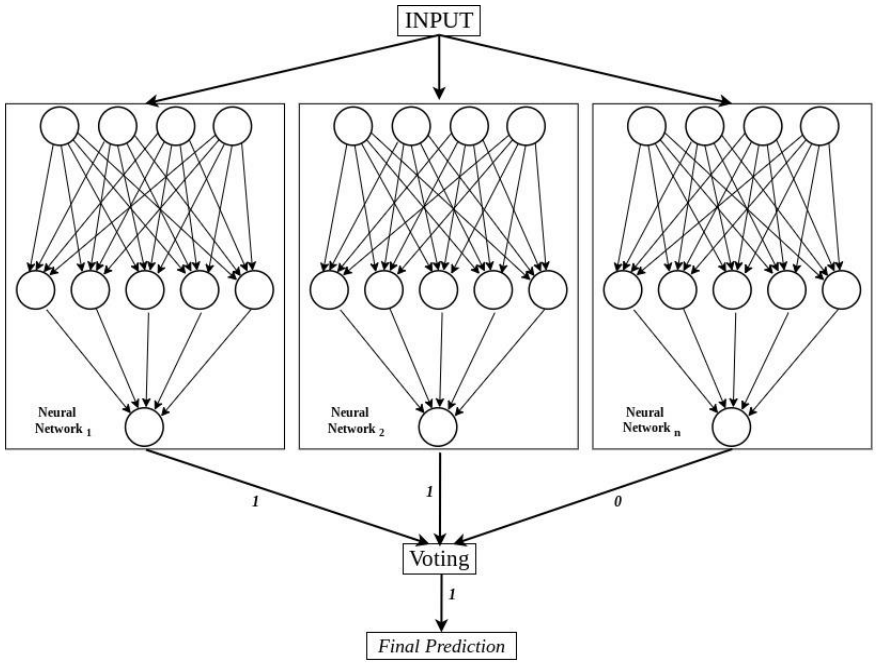

Figure 4 – Diagram of Ensemble Neural Networks

Note. The diagram represents an ensemble model using a voting mechanism with multiple NNs. The model consists of three NNs (the total number of models should be odd to avoid ties and ensure a majority vote), each receiving the same set of input features. After processing the inputs, each NN produces its own prediction. The final prediction is decided through a voting process, as in ensemble learning, where the predictions of the individual NNs are aggregated. If the majority of the NNs predict the same outcome, that outcome becomes the final prediction. This ensemble approach might help to increase the reliability and robustness of the model's predictions by reducing the likelihood of errors from any single NN.

Ensemble Machine Learning Applications in Various Domains

In a comprehensive exploration of ensemble machine learning techniques, several studies have conducted illuminating their effectiveness in various domains. First, Hamza & Larocque (2006) conducted an extensive evaluation using 14 publicly available datasets. Their comparison of RF, bagging, boosting, and single tree methods revealed RF's superiority in terms of overall performance, particularly its robustness against noise. Notably, RF statistically outperformed

other methods, with bagging and boosting showing no significant difference. This study underscores the adaptability and effectiveness of RF in handling complex datasets.

Similarly, Marchese Robinson et al. (2017) applied RF alongside linear modeling techniques to predict biological activity. Through precise evaluation employing cross-validation and metrics such as MCC, Kappa, AUC, and BA, RF consistently outperformed linear models. This finding highlights the advantage of ensemble methods in capturing complex relationships within biological datasets.

Moving beyond biological applications, Ebrahimi et al. (2019) emphasized the importance of early detection in subclinical bovine mastitis. By applying various machine learning models, including GB and ANN, they overcame the limitations of traditional tests. Notably, GB achieved impressive accuracy and sensitivity, demonstrating its potential for timely disease detection.

Finally, Wen et al. (2009) addressed the challenge of detecting liver metastasis using AB and logistic regression models. Employing feature selection techniques, they identified important variables and presented AB's superiority, particularly in handling missing values. This study underlines the importance of tailored model selection and feature engineering in optimizing ensemble methods for specific medical diagnoses.

Collectively, these studies show the effectiveness of ensemble machine learning techniques across diverse domains. While each application presents unique challenges and considerations, the consistent success of ensemble methods underscores their potential as powerful tools for predictive modeling and decision-making in complex real-world scenarios.

Ensemble Learning in Heart Disease

Aggrawal & Pal (2020) used six models and eight ML classifiers were used to predict CVD deaths. In model 2 and 4, RF showed the highest scores for accuracy, precision, recall, F scores and AUC scores. Model 2 uses a correlation matrix to find key features. Age, anemia, hypertension, serum creatinine are some of the eight highly correlated features that are chosen. In model 4, the select from model and linear SVC approaches are employed.

Marchese et al. (2017) improved the predictive ability of ML models for heart disease by combining chi-square with principal component analysis. The study enhances raw data analysis by dimensionality reduction approaches, as it is not possible to employ all features. The best accuracy is obtained by using RF in Cleveland, Hungarian, and CH datasets.

Neural Network Applications in Heart Disease

Yadav et al. (2020) of the paper emphasize the advantages of ML in healthcare, particularly in regards with the detection of heart disease Eight models including LR, Fuzzy KNN and NN are employed and evaluated employing benchmark data from the UCI repository. Accuracy score and findings from confusion matrix analysis showed that Neural Network and Fuzzy KNN performed better than other models.

Bhatt et al. (2023) focuses on using machine learning methods to make accurate CVD diagnosis via the Kaggle dataset. Considering that accurate classification is crucial in directing appropriate therapy, the research suggests a model that uses k-modes clustering with Huang initialization to enhance accuracy. GridSearchCV was used to optimize a range of techniques, including RF, decision tree classifier, NN, and XGB. Accuracy results range from 86.37% to 87.28%. According to the results, NN showed the best cross-validation score 87.28% as well as high precision, recall, F, and AUC scores.

Khemphila & Boonjing (2010) applied LR, DT, and artificial neural networks to predict heart disease. Among these methods, artificial neural networks showed the highest accuracy with %80.2, the lowest error rate with 0.198 and higher specificity compared to LR and DT.

Handling Class Imbalance: Data and Algorithm Level Approaches

In ML, overcoming imbalanced data presents a challenge, but various strategies exist at both the data algorithm levels to address this issue effectively. Data-level approaches involve preprocessing techniques aimed at balancing the distribution of classes, such as oversampling (e.g., SMOTE) and undersampling (e.g., random undersampling) (Ali et al., 2019, p. 1563). In this study, data-level approaches were discussed. Data augmentation is another example of a data-level approach that plays a crucial role, particularly in domains like image classification, although

it is applicable across various data types. It involves introducing artificial examples for the underrepresented class, thus enhancing its presence in the dataset. This technique entails applying transformations to existing data to generate additional samples. Common techniques include rotating images, horizontally or vertically flipping images, or zooming in or out (Wang & Perez, 2017, pp. 2-3).

Algorithm-level methods for handling imbalanced data involve adjusting existing learning algorithms to reduce their bias toward majority classes. A common approach is cost-sensitive learning, where the algorithm adjusts to assign different penalties to various groups of examples. This adjustment helps prioritize less represented classes, aiming to minimize overall classification errors. However, determining suitable values for the cost matrix can be challenging, especially when expert guidance is unavailable for real-world problems (Krawczyk, 2016, p. 222). Additionally, hybrid techniques provide a solution to overcome class imbalance by directly integrating oversampling and undersampling into the learning process within algorithms such as Easy Ensemble and Balanced Random Forest.

Imbalanced Data: Oversampling Techniques

The term imbalanced data is generally used to emphasize the inequity distribution within the target column in the dataset. For example, suppose we have a dataset of 5000 data for heart disease. And in this dataset, the target column states whether a patient has been diagnosed with heart disease. If only 100 instances show patients with heart disease, while the remaining 4900 do not, the dataset is imbalanced. Thus, the machine learning model can struggle to learn from the minority class. As a result, model performance can suffer, especially predicting the minority class.

ROS

ROS is a widely used straightforward oversampling technique. After the ROS finds the minority class out, it randomly chooses instances and replicates them to increase the representation in the dataset. Replicated instances are added to the original dataset. Besides the

simplicity, it may cause over-fitting since the same samples in minority class are repeated (Abdi & Hashemi, 2016, p. 239).

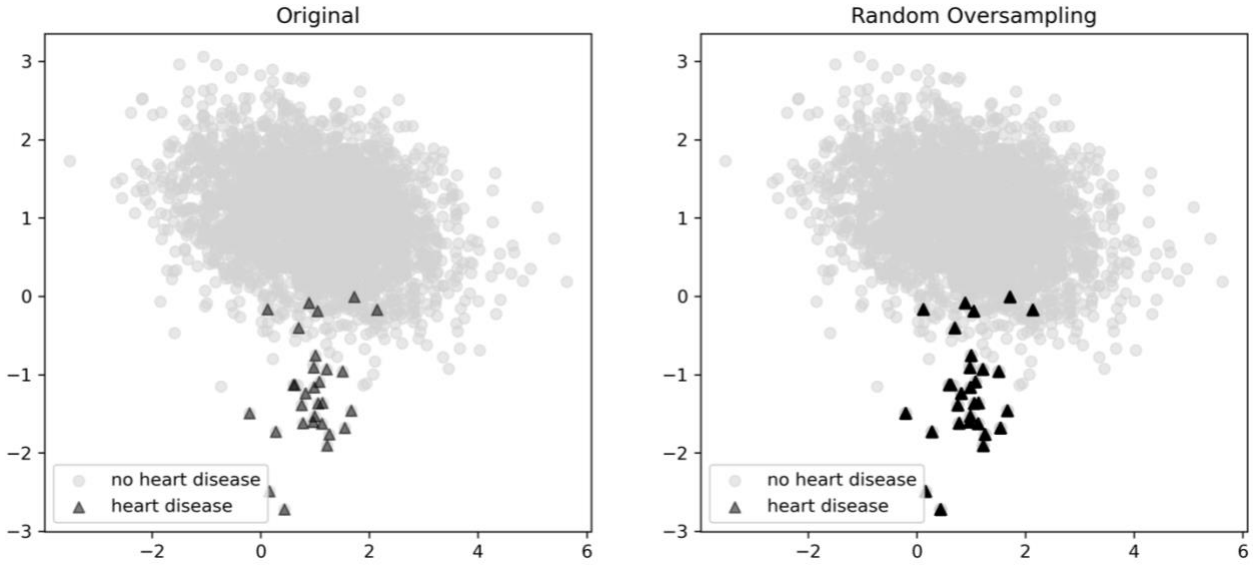

Figure 5 - Class Distribution Before and After Applying Random Oversampling

Note. The figure above (Figure 5) and the following figures (Figures 6 and 7) demonstrate the class distribution of a synthetic dataset with 3,000 samples, and 2 features, highlighting the initial imbalance (left) and the effects of oversampling techniques (center or right). The imbalanced dataset (left) has roughly 2,970 samples in the "no heart disease" class and 30 in the "heart disease" class. The darker color represents the increased presence of examples, indicating the balancing effect.

In Figure 5, the original dataset (left) shows a significant class imbalance, corrected through random oversampling (right) by replicating examples from the minority class.

SMOTE

SMOTE randomly selects minority class instances and uses a distance metric to find its k-nearest neighbors. After randomly selecting one of these neighbors, it creates artificial instances along the line connecting the selected sample and its neighbor (Gosain & Sardana, 2017, p. 80). While SMOTE oversample the minority class, it does not take the majority class into account, which can lead to overlap between classes. Yet, it is widely used by researchers due to its

simplicity and the additional value it offers compared to ROS technique (Maldonado et al., 2019, p. 381).

B-SMOTE

B-SMOTE is an extension of SMOTE that focuses on the instances that are difficult to classify. Because ML models are more likely to misclassify instances that are close to the decision boundary compared to those further away. First, B-SMOTE finds the instances near to the decision-making line and then employs the SMOTE to create synthetic data and add to the original training dataset (Han et al., 2005).

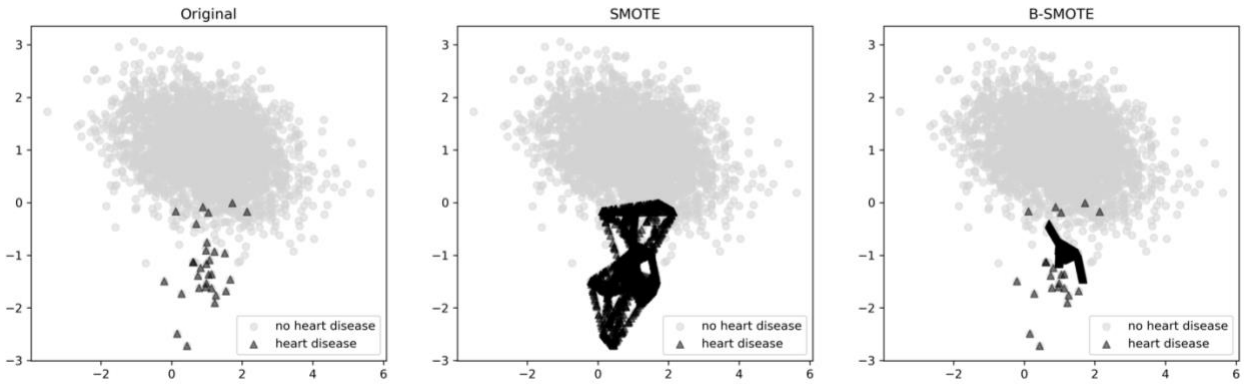

Figure 6 – Class Distribution Before and After Applying SMOTE and B-SMOTE Oversampling

Note. SMOTE (center) addresses the imbalance by generating synthetic samples to augment the minority class. B-SMOTE (right) further extends this approach by focusing on generating synthetic samples near the class boundaries, thereby enhancing class separation.

ADASYN

ADASYN generates synthetic samples like SMOTE to balance the distribution. However, ADASYN advances the SMOTE by adaptively modifying the density of synthetic samples according to the degree of classification difficulty for each instance through k-nearest neighbors. ADASYN first determines the imbalance ratio between the majority and minority classes. After calculating the density of each instance in the minority class, it creates synthetic samples with respect to the imbalance ratio and the predicted density of their neighbors. More synthetic samples are given to instances in denser regions and fewer to those in smaller regions (He et al., 2008). These

features of ADASYN can reduce the bias and dynamically adjust the classification decision boundary to focus more on challenging samples (Ahmed et al., 2022, p. 7).

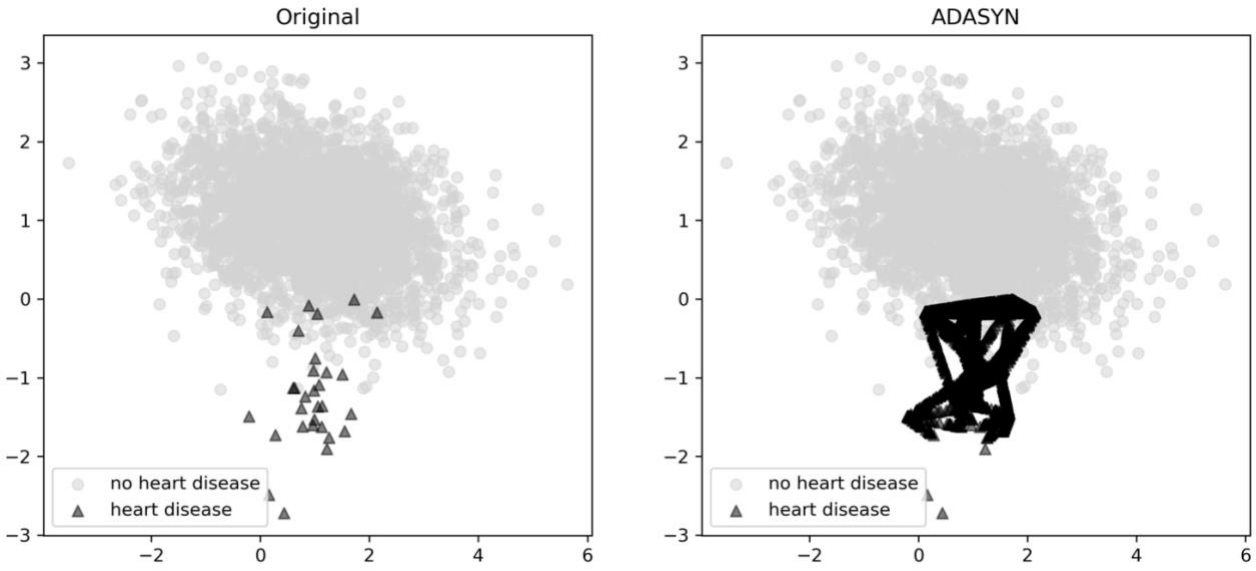

Figure 7 – Class Distribution Before and After Applying ADASYN Oversampling

Note. Applying ADASYN (right), the minority class is balanced by generating synthetic samples that focus on harder-to learn examples.

Related Works: Oversampling

Kovács (2019) compared and evaluated 104 datasets through 85 oversampling methods such as SMOTE, B-SMOTE, SL-SMOTE and ADASYN. The evaluation metrics applied included the geometric mean, F-score, AUC-score, precision, and cross-validation. The results indicated a significant improvement in the performance of classification models with the use of oversampling techniques.

Mohammed et al. (2020) used a dataset from Kaggle competition named Santander Customer Transaction Prediction. They applied undersampling and oversampling techniques with different machine learning models to solve the issue of class imbalance. Using a variety of measures, they assessed the classifiers and found that, on average, oversampling performed better than undersampling across classifiers, leading to higher evaluation metrics scores.

While Viloria et al. (2020) emphasize the importance of deep learning in high-dimensional data, the problems caused by imbalanced data are mentioned. To find a solution to this problem, Random Sub-Sampling (RUS), ROS, and SMOTE techniques were employed. The results reveal that RUS gives almost the same or even less results as the original databases and SMOTE outperforms in terms of AUC score.

3 Methodology

This chapter details the methodology used to assess whether an ENN performs better than previously mentioned ensemble models in predicting CVD after applying oversampling techniques. The methodology is structured into four sections: data collection and selection, data preprocessing, model training and testing, and evaluation metrics.

Data collection and selection describes the datasets and the criteria for their selection. Data preprocessing covers the steps taken to prepare the data, including normalization, handling missing values, and addressing outliers. Model training and testing involved the use of various libraries, splitting the data into training and testing sets, applying sampling techniques to balance the training data, and tuning hyperparameters to enhance model performance. The testing phase evaluated the models' ability to generalize to unseen data. Lastly, evaluation metrics explains the performance criteria, such as accuracy, F-score, AUC and G-mean, used to compare the models comprehensively.

3.1 Data Collection and Selection

Three open-source benchmark datasets related to CVD were used in this study. Diversity in dataset size, features, and class distribution was considered to ensure robust model evaluation and comparison. The imbalance ratio was calculated as the number of instances in the minority class divided by the number of instances in the majority class to provide a better understanding of the class distribution within each dataset. This approach helps the evaluation and comparison of model performance across datasets by providing insights into the imbalance between the classes.

Table 1 – Overview of Datasets Used in the Study

Dataset	Number of instances	Presence	Absence	Number of features	Imbalance Ratio
Heart Disease Health Indicators	253680	23893	229787	21	9.617
Framingham	4240	644	3590	15	5.583
Statlog	270	120	150	13	1.25

Note. This table provides an overview of the datasets used in the study, including the number of instances, the number of features, the presence and absence of the target variable, and the imbalance ratio.

The Heart Disease Health Indicators dataset derived from the 2015 Behavioral Risk Factor Surveillance System (BRFSS) is a cleaned and merged version of the original data and is one of the datasets used in this study. The original data was collected by the Centers for Disease Control and Prevention and is available on Kaggle (Centers for Disease Control and Prevention, 2015). It is notable for its large-scale data representation compared to others, providing extensive insights into diverse indicators related to cardiovascular health. With a high imbalance ratio, this dataset poses challenges typical of scenarios where minority class instances are significantly outnumbered by the majority class. Framingham dataset, also retrieved from Kaggle, offers a moderate-sized sample with a focus on essential cardiovascular risk factors. Its imbalance ratio underscores the prevalence of certain risk factors over others, crucial for predictive modeling in cardiovascular disease research. The Statlog dataset, retrieved from UCI Machine Learning Repository (UCI Machine Learning Repository, n.d.), despite its smaller size, provides concise yet comprehensive features relevant to heart disease diagnosis. Its relatively low imbalance ratio indicates a more balanced distribution between classes, facilitating nuanced analysis and model training in CVD prediction tasks.

Table 2 – Description of Features in the Heart Disease Health Indicators

Feature	Description	Type	Missing values
HighBP	The person has high blood pressure, which means their blood moves through their blood vessels with more force than normal. This can strain the heart and arteries. (0 = No, 1 = Yes)	Categorical	-
HighChol	The person has high levels of cholesterol, a type of fat in the blood that can increase the risk of heart problems. (0 = No, 1 = Yes)	Categorical	-
CholCheck	If the person's cholesterol has been examined during the last five years. (0 = No, 1 = Yes)	Categorical	-
BMI	The calculation of a person's Body Mass Index (BMI) is based on the person's weight and height. It gives an idea of whether someone has a healthy body weight or if they are underweight, overweight, or obese. (12–98)	Numerical	-
Smoker	Whether a person currently smokes cigarettes. (0 = No, 1 = Yes)	Categorical	-
Stroke	If the person has ever experienced a stroke, which is a condition when blood flow to a portion of the brain is cut off. (0 = No, 1 = Yes)	Categorical	-
Diabetes	If the person has diabetes, a condition where the body cannot properly process sugar (multiple levels, including no diabetes, prediabetes, and diabetes). (0 = No, 1 = Yes, 2 = Prediabetes)	Categorical	-
PhysActivity	The person regularly engages in physical activities or exercises. (0 = No, 1 = Yes)	Categorical	-
Fruits	Does the person often consume fruits? (0 = No, 1 = Yes)	Categorical	-
Veggies	Does the person often consume vegetables? (0 = No, 1 = Yes)	Categorical	-
HvyAlcoholConsump	If a person consumes alcohol heavily. (0 = No, 1 = Yes)	Categorical	-
AnyHealthcare	Whether a person has access to healthcare coverage. (0 = No, 1 = Yes)	Categorical	-
NoDocbcCost	If a person has avoided seeing a doctor due to cost. (0 = No, 1 = Yes)	Categorical	-

GenHlth	A self-reported measure of general health status. (usually on a scale from 1 to 5)	Categorical	-
MenHlth	How many days in the past month the person felt mentally unwell, which could include stress, depression, or anxiety. (0-30 days)	Numerical	-
PhysHlth	How many days in the past month the person felt physically unwell, which might include pain or illness. (0-30)	Numerical	-
DiffWalk	Whether a person has trouble moving around or using stairs. (0 = No, 1 = Yes)	Categorical	-
Sex	Biological sex (0 = Female, 1 = Male)	Categorical	-
Age	Categorizes the person's age into groups, starting from 18-24 years up to 80 and older. (1 = 18-24, 2 = 25-29, ..., 13 = 80+)	Categorical	-
Education	The highest level of education the person has achieved, from never having attended school to having graduated from college. (1 = Never went to school; 6 = Graduated from college)	Categorical	-
Income	Income level, grouped into categories from less than $10,000 to $75,000 or more (1 = Less than $10,000, 8 = $75,000 or more)	Categorical	-
HeartDiseaseor Attack (target)	Indicates whether the respondent has ever been told they have coronary heart disease or a heart attack. (0 = No, 1 = Yes)	Categorical	-

Each of the datasets used in this study contributes unique characteristics that are crucial for analyzing heart disease risk. The Heart Disease Health Indicators dataset includes a range of categorical and numerical features such as high blood pressure, cholesterol levels, smoking status, and lifestyle factors like physical activity and diet. It also incorporates demographic variables such as age, sex, education, and income, offering a comprehensive view of cardiovascular health indicators without missing values.

Table 3 – Description of Features in the Framingham Dataset

Feature	Description	Type	Missing values
male	The person's gender (0 = Female, 1 = Male)	Categorical	-
age	Age of the person in years. (32-70)	Numerical	-
education	The highest education level of the person. (1 = Some high school, 2 = High school/GED, 3 = Some college, 4 = College)	Categorical	105
currentSmoker	Indicates if the person currently smokes. (0 = No, 1 = Yes)	Categorical	-
cigsPerDay	The quantity of cigarettes smoked daily. (0-70)	Numerical	29
BPmeds	Use of medication for high blood pressure. (0 = No, 1 = Yes)	Categorical	53
prevalentStroke	The person has had a stroke. (0 = No, 1 = Yes)	Categorical	-
prevalentHyp	The person has been diagnosed with high blood pressure. (0 = No, 1 = Yes)	Categorical	-
diabetes	Indicates if the person has diabetes. (0 = No, 1 = Yes)	Categorical	-
totChol	The total cholesterol in the blood with levels below 200 mg/dL considered healthy, 200-239 mg/dL as borderline high, and 240 mg/dL or higher as high, indicating increasing risk of heart disease and stroke. (107 – 696 mg/dL)	Numerical	50
sysBP	The pressure in the arteries when the heart beats (contracts), with levels ranging from 83 to 295 mm Hg; normal levels are less than 120 mm Hg, elevated levels are 120-129 mm Hg, and high levels are 130 mm Hg or above, increasing the risk of heart disease. (mm Hg 83 - 295)	Numerical	-
diaBP	The pressure within the arteries when the heart is in a resting state (between beats), ranging from 48 to 150 mm Hg; normal levels are less than 80 mm Hg, elevated levels are 80-89 mm Hg, and high levels are 90 mm Hg or above, indicating increased cardiovascular risk.	Numerical	-
BMI	Body Mass Index (kg/m2 15.5 - 56.8)	Numerical	19
heartRate	The number of heart beats per minute (bpm); for adults, a normal resting heart rate usually ranges from 60 to 100 bpm. Levels above or	Numerical	1

	below this range can indicate various health issues. (from 44 to 143 bpm)		
glucose	The level of glucose (sugar) present in the blood, normal glucose levels in fasting typically range from 70 to 99 mg/dL. Levels between 100–125 mg/dL indicate prediabetes, and 126 mg/dL or higher on two separate tests indicate diabetes. (from 40 to 394 mg/dL)	Numerical	388
TenYearCHD (target)	Risk of coronary heart disease over ten years. (0 = No, 1 = Yes)	Categorical	–

The Framingham dataset is another critical source that focuses on clinical and lifestyle attributes, including age, blood pressure, cholesterol levels, and glucose levels, to predict the ten-year risk of coronary heart disease. Unlike the Heart Disease Health Indicators and Statlog datasets, some features in Framingham, such as education, cigarette usage, and BMI, have missing values, which need careful handling in the preprocessing stage.

Table 4 – Description of Features in the Statlog Dataset

Feature	Description	Type	Missing values
age	The person's age in years. (29–77)	Numerical	–
sex	Gender indicator (0 = Female, 1 = Male)	Categorical	–
cp	Categorizes the type of chest pain felt by individual: 0 = Typical angina (chest pain related to heart issues), 1 = Atypical angina (chest pain not clearly associated to heart problems), 2 = Non-anginal pain (pain not associated to heart), 3 = Asymptomatic (no chest pain).	Categorical	–
trestbps	The blood pressure (in mm Hg) when the individual is at rest, typically taken on admission to the hospital, normal levels are less than 120/80 mm Hg. High resting blood pressure indicates increased cardiovascular risk. (94–200)	Numerical	–
chol	The total cholesterol level in the blood, ideal levels are less than 200 mg/dL, borderline high	Numerical	–

	is 200–239 mg/dL, and high is 240 mg/dL or above. (126 – 564)		
fbs	Whether the person's fasting blood glucose level is higher than 120 mg/dL. (0 = No, 1 = Yes)	Categorical	–
restecg	Heart's electrical activity at rest: 0 = Normal, 1 = ST-T wave abnormality (possible heart issues like past heart attacks), 2 = Left ventricular hypertrophy (thickening of the heart's primary pumping chamber, often from high blood pressure).	Categorical	–
thalach	Maximum heart rate achieved. (bpm 71 – 202)	Numerical	–
exang	Whether the individual experiences chest pain during physical activity. (0 = No, 1 = Yes)	Categorical	–
oldpeak	The overall cholesterol in the blood, ranging from 0 to 6.2 (unit unclear, but typically measured in mmol/L or mg/dL).	Numerical	–
slope	Shows how the ST segment of an ECG changes during intense exercise: 0 = Rises steadily (upsloping), 1 = Stays flat, 2 = Drops (downsloping).	Categorical	–
ca	Counts the number of major blood vessels (out of 3) that show up with color during a fluoroscopy test, which uses X-rays to look at blood vessels in the heart.	Numerical	–
thal	The condition of thalassemia in a patient: 1 = Normal (no issues), 2 = Fixed defect (permanent changes in the heart), 3 = Reversible defect (temporary issues that can improve).	Categorical	–
target	Heart disease (0 = No, 1 = Yes)	Categorical	–

Lastly, the Statlog dataset primarily consists of clinical measures like chest pain type, resting blood pressure and electrocardiographic results, as well as demographic characteristics like age and sex. All features are complete with no missing values, making it straightforward for analysis. Each dataset provides a different perspective on heart disease, allowing for robust model training and testing to determine the most effective predictors and methods for diagnosing cardiovascular conditions.

3.2 Data Preprocessing

Prior to model training, the datasets underwent preprocessing to address missing values, outliers, and correlation analysis using Python libraries such as Pandas and Scikit-learn. Since NN are sensitive to high correlations (unlike tree-based models), correlation analysis was specifically applied only for the ENN. In the Framingham dataset, several variables had missing values. Glucose had the highest proportion, with 388 missing entries (9.15% of the dataset), followed by education with 105 missing values (2.48%), and BPMeds with 53 missing values (1.25%). TotChol and cigsPerDay had fewer missing values, with 50 (1.18%) and 29 (0.68%), respectively. BMI had 19 missing entries (0.45%), and heartRate had just 1 missing value (0.02%).

For imputation, methods implemented through Scikit-learn's SimpleImputer were used: For glucose and heartRate, median imputation was chosen to handle outliers and prevent distortion from extreme values. The education variable, which is categorical, was imputed with the mode to reflect the most common educational level. The cigsPerDay variable was also imputed with the mode, as the most frequent value is 0 (non-smoking), which significantly outweighs other values. Similarly, BPMeds, a binary variable, was imputed with the mode, given that most individuals are not on blood pressure medication (0). For both totChol and heartRate, where the mean, median, and mode are nearly identical, mean imputation was applied to reflect the central tendency of the data and maintain consistency with the overall distribution. Similarly, for BMI, where the mean and median are nearly identical, but the mode is significantly different, mean imputation was also chosen. The close alignment of the mean and median suggests a symmetric distribution, making the mean a more accurate reflection of the central tendency compared to the mode.

These imputation methods were carefully selected to minimize bias and ensure the accuracy of the subsequent analyses. There were outliers in some columns across each dataset, reflecting the natural variability in health metrics such as cholesterol, blood pressure, and glucose levels. Instead of removing these outliers, they were retained to preserve critical information about individuals with extreme health conditions, which are key for accurately predicting heart disease risk. Discarding these outliers could lead to a significant loss of valuable insights,

especially for high-risk groups. Ensemble tree-based methods are inherently robust to outliers and can handle these variations without effecting the performance. For NNs, which are more sensitive to outliers, normalization was applied to minimize their impact. By keeping the outliers, the datasets better capture the full spectrum of health variability, ensuring the models are trained on data that closely resembles real-world scenarios where such extreme values are common and clinically relevant.

3.3 Model Training and Testing

Before training the model, Min-Max normalization was applied exclusively to the ENN using Scikit-learn's MinMaxScaler. This decision was made because Min-Max normalization scales features to a specific range, typically [0, 1], which helps to ensure that all features contribute equally to the model's learning process and reduce the effect of outliers. ENN benefit from this scaling to improve convergence and stability during training. Normalizing features can lead to faster convergence and better performance by preventing certain features from disproportionately influencing the model due to differences in scale. In contrast, tree-based models do not require normalization because they are inherently robust to varying feature scales, as their decision-making process based on the relative sequence of feature values instead of their absolute magnitudes.

Table 5 - Implementation Steps for an Ensemble Neural Network Approach

1:	Begin
2:	Apply MinMaxScaler to normalize the data
3:	Apply oversampling techniques
4:	Set the random seed for reproducibility
5:	Define the neural network architecture with input layer, hidden layers (ReLU), and output layer (Sigmoid)
6:	Specify the number of neural networks to be trained in the ensemble and initialize (e.g., 3)
7:	Train each model using the training data with a defined number of epochs and batch size

8:	Make predictions for both training and test datasets
9:	Apply a threshold to convert probabilities to binary class predictions
10:	Aggregate predictions using majority voting from all models in the ensemble
11:	Calculate performance evaluation metrics
12:	End

Following preprocessing, the dataset was divided into training and testing sets using a stratified split, ensuring that class distributions were maintained in both subsets (with stratify=y). The data was split, with 75% used for training and the remaining 25% used for testing. To address class imbalances and enhance model performance, oversampling techniques were applied to the training data. GridSearchCV was used to fine-tune the hyperparameters. GridSearchCV is a technique that systematically searches a set of specified hyperparameters and uses cross-validation to assess the model's performance for each combination to determine the ideal settings. The testing phase assessed the models' ability to generalize to unseen data. All analyses and model development were conducted using Python within Jupyter Notebook, utilizing libraries such as Scikit-learn for machine learning, Pandas for data manipulation, and TensorFlow for deep learning tasks.

Table 6 - Hyperparameters grid

Model	Hyperparameters	Values
RF	maximum depth	2, 3
	number of estimators	50, 150
AB	maximum depth	2, 3
	number of estimators	50, 150
GB	number of estimators	50, 150
XGB	maximum depth	2, 3
	number of estimators	50, 150
ENN	number of epochs	10, 30

	batch size	16, 32

3.4 Evaluation Metrics

Evaluation metrics are crucial for understanding the performance of models, especially their capacity to distinguish between different classes. The confusion matrix is one of the most popular tools for this purpose. Confusion matrix offers an in-depth summary of a model's performance by comparing its predictions with the true outcomes. It allows us to evaluate various aspects of the model's performance, such as its accuracy, precision, recall, and the overall reliability of its predictions. The confusion matrix provides the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN), giving a comprehensive view of how well the model performs in different categories.

Table 7 - Confusion Matrix for Model Evaluation

	Predicted Positive (Heart Disease)	**Predicted Negative** (No Heart Disease)
Actual Positive (Heart Disease)	True Positive (TP)	False Negative (FN)
Actual Negative (No Heart Disease)	False Positive (FP)	True Negative (TN)

In the matrix, true positives (TP) indicate the count of instances accurately predicted as positive. In contrast, false positives (FP) refer to instances that were mistakenly classified as positive. True negatives (TN) represent the number of instances correctly identified as negative, whereas false negatives (FN) show the instances that were incorrectly predicted as negative.

Each cell in the matrix represents the counts of predicted versus actual classifications, allowing for the calculation of various performance metrics. Precision, recall, and F-score are derived from these values to evaluate the model's accuracy and effectiveness in identifying

positive cases. Analyzing these metrics provides valuable insights into the model's strengths and shortcomings, helping to inform further adjustments and enhancements.

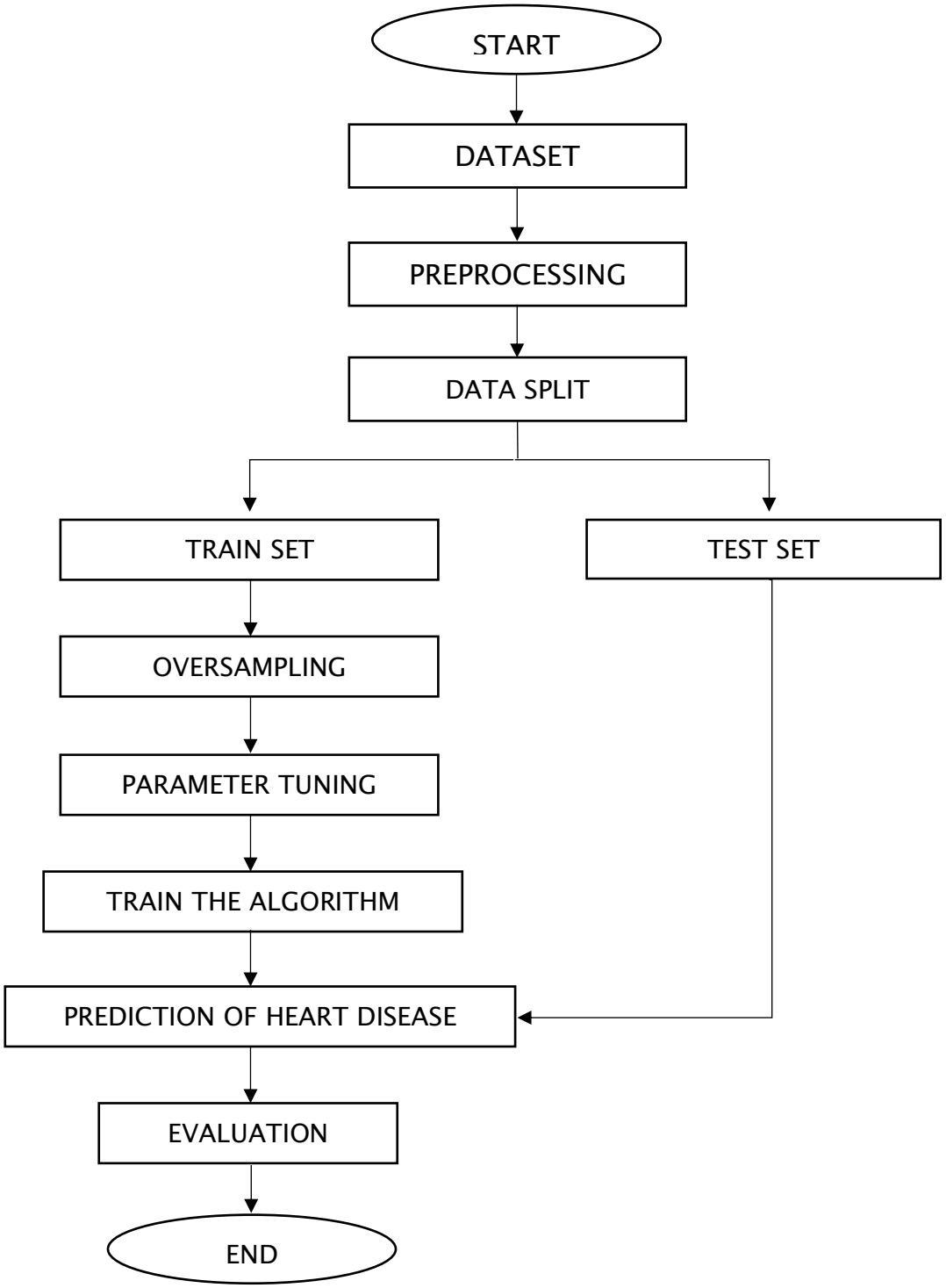

Figure 8 – Flowchart of implementation process

Accuracy

The studies often prioritize improving accuracy. But when it comes to imbalance data, accuracy alone might not represent the performance accurately (Huang & Ling, 2005). Accuracy can be misleading, particularly in the context of heart disease, since the disease frequency is much lower than that of its absence. For example, if 95% of the dataset has samples without heart disease and only 5% with heart disease, a model that predicts all samples as negative will still achieve a high accuracy of 95% even if it does not identify any positive cases.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

F-Score

The F-score balances the Precision and Recall, summarizing the model's accuracy in positive predictions. $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ are combined to calculate the F-Score as follows:

$$F - score = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \quad (2)$$

Precision and Recall are crucial metrics for assessing the effectiveness of a classification model: Precision indicates the proportion of correct positive predictions, calculated as the ratio of true positives (TP) to the total of true positives and false positives (FP). Recall evaluates the model's capacity to capture all actual positive cases, defined as the ratio of true positives to the sum of true positives and false negatives (FN). The F-score integrates both Precision and Recall into a single metric, offering a balanced assessment of the model's performance.

Area Under the ROC Curve

AUC summarizes how well the model ranks true positive rates against false positive rates across various thresholds, providing a single metric for overall discrimination performance.

Geometric Mean

The geometric mean is a performance metric that balances the $Sensitivity = \frac{TP}{TP + FN}$ and $Specificity = \frac{TN}{TN + FP}$. G-mean provides a more accurate assessment of a model's effectiveness when one class dominates the dataset as follows:

$$G - mean = \sqrt{Sensitivity\ x\ Specificity} \tag{3}$$

4 Results and Discussion

This study evaluates the performance of several ensemble models, including an ENN that combines three neural networks, in classifying heart disease health indicators across various oversampling techniques. The models were assessed using accuracy, F-score, G-mean, and AUC metrics. The oversampling methods applied were ROS, SMOTE, B-SMOTE, and ADASYN. Below, the results are presented and discussed in detail.

Separate tables have been created for each evaluation metric, including accuracy, F-score, G-mean, and AUC. After the evaluations for these metrics are complete, the analysis will shift to another dataset. I will study on Heart Disease Health Indicators, Framingham and Statlog datasets respectively.

Table 8 - Heart Disease Health Indicators: Accuracy Results for Models and Oversampling Techniques

Model	**NONE** (Train/Test)	**ROS** (Train/Test)	**SMOTE** (Train/Test)	**B-SMOTE** (Train/Test)	**ADASYN** (Train/Test)
RF	0.905 / 0.905	0.754 / 0.711	0.828 / 0.775	0.841 / 0.781	0.825 / 0.768
AB	0.907 / 0.907	0.769 / 0.753	0.924 / 0.891	0.922 / 0.886	0.923 / 0.889
GB	0.908 / 0.907	0.774 / 0.735	0.929 / 0.892	0.927 / 0.888	0.930 / 0.893
XGB	0.908 / 0.908	0.777 / 0.736	0.941 / 0.905	0.941 / 0.905	0.942 / 0.907
ENN	0.908 / 0.907	0.775 / 0.736	0.865 / 0.831	0.874 / 0.820	0.868 / 0.823

In the accuracy analysis of the Heart Disease Health Indicators dataset (Table 7), the baseline models trained without oversampling (NONE) generally had the highest training and testing accuracies. This demonstrates the models' effectiveness in identifying the majority class in the dataset. For example, the models achieved about 90% accuracy on the test set. However, when oversampling techniques were applied, especially with ROS, the results dropped significantly. This drop in accuracy does not mean that the model performs worse. Instead, it indicates that the model is now learning patterns from both majority and minority classes.

After applying oversampling techniques, ENN outperformed the RF model, achieving AUC scores of 74% with ROS, 83% with SMOTE, and 82% with both B-SMOTE and ADASYN but ENN's results were still lower than those of the other models. Although ENN demonstrated comparable accuracy to the tree-based models when no oversampling was used, it did not show the same level of improvement when oversampling techniques were applied.

Table 9 - Heart Disease Health Indicators: F-score Results for Models and Oversampling Techniques

Model	**NONE** (Train/Test)	**ROS** (Train/Test)	**SMOTE** (Train/Test)	**B-SMOTE** (Train/Test)	**ADASYN** (Train/Test)
RF	0.000 / 0.000	0.765 / 0.343	0.835 / 0.364	0.847 / 0.369	0.833 / 0.362
AB	0.225 / 0.223	0.773 / 0.374	0.922 / 0.368	0.920 / 0.371	0.921 / 0.366
GB	0.190 / 0.179	0.784 / 0.367	0.927 / 0.352	0.926 / 0.361	0.928 / 0.336
XGB	0.191 / 0.182	0.786 / 0.367	0.939 / 0.229	0.938 / 0.238	0.940 / 0.221
ENN	0.115 / 0.109	0.784 / 0.368	0.866 / 0.387	0.878 / 0.386	0.870 / 0.373

While accuracy is often used to evaluate model performance, it can be misleading, especially in the context of imbalanced datasets. To get a clearer picture of how well the models are identifying minority classes, the F-score, which balances precision and recall, is much more revealing.

Without oversampling, F-scores were particularly low across models; for example, RF model had an F-score of 0%, while other models also struggled with low F-scores. This indicates how challenging it was for these models to correctly identify minority samples in a highly imbalanced dataset. Once oversampling was applied, there was a significant improvement in F-scores across all models. ENN, along with AB, GB, and XGB, achieved the highest F-score with ROS (37%). It also reached top F-scores with SMOTE (39%), B-SMOTE (39%), and ADASYN (37%). These results suggest that ENN consistently performed well across different oversampling methods.

Table 10 - Heart Disease Health Indicators: G-mean Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.000 / 0.000	0.766 / 0.419	0.835 / 0.412	0.848 / 0.414	0.833/ 0.413
AB	0.274 / 0.272	0.773 / 0.439	0.922 / 0.369	0.920 / 0.371	0.922 / 0.367
GB	0.256 / 0.244	0.785 / 0.439	0.928 / 0.355	0.926 / 0.362	0.929 / 0.341
XGB	0.258 / 0.248	0.787 / 0.439	0.940 / 0.271	0.939 / 0.278	0.941 / 0.270
ENN	0.199 / 0.191	0.785 / 0.440	0.866 / 0.408	0.878 / 0.413	0.870 / 0.395

Table 9 shows the results of the G-mean metric for Heart Disease Health Indicators dataset, which measures how well the models balance sensitivity and specificity. Without oversampling, the G-mean scores of the models are quite low same as the F-Score; for example, the RF model scored 0%, indicating that it failed to identify minority class samples.

With oversampling, the G-mean scores improved significantly for all models, showing that they became more sensitive to the minority class. ENN achieved the highest test G-mean with ROS (44%), SMOTE (41%) and B-SMOTE (41%). Overall, ENN's performance highlights its reliability in identifying minority samples when combined with oversampling methods.

Table 11 - Heart Disease Health Indicators: AUC Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.500 / 0.500	0.754 / 0.752	0.828 / 0.734	0.841 / 0.735	0.825 / 0.736
AB	0.564 / 0.564	0.769 / 0.767	0.924 / 0.642	0.922 / 0.649	0.923 / 0.643
GB	0.552 / 0.548	0.774 / 0.771	0.929 / 0.631	0.927 / 0.640	0.930 / 0.621
XGB	0.552 / 0.549	0.777 / 0.770	0.941 / 0.566	0.941 / 0.570	0.942 / 0.563
ENN	0.529 / 0.528	0.775 / 0.771	0.865 / 0.712	0.874 / 0.721	0.868 / 0.704

Table 10 presents AUC results, highlighting each model's ability to differentiate between positive and negative cases. Without oversampling, the models generally struggled, with AUC scores close to random guessing. Among them, AB achieved the highest AUC at 56%, while RF had the lowest at 50%. ENN was the next lowest model after RF with a score of 53%.

With oversampling, ENN demonstrated strong discrimination capability, achieving the highest AUC on the test set with ROS (77%), matching AB, GB, and XGB. ENN also performed well with other oversampling methods, achieving test AUC values of 71% with SMOTE, 72% with Borderline-SMOTE, and 70% with ADASYN. RF achieved the best performance in three out of the four oversampling methods, with ENN ranking second best across methods.

Table 12 - Framingham: Accuracy Results for Models and Oversampling Techniques

Model	**NONE** (Train/Test)	**ROS** (Train/Test)	**SMOTE** (Train/Test)	**B-SMOTE** (Train/Test)	**ADASYN** (Train/Test)
RF	0.850 / 0.849	0.677 / 0.658	0.698 / 0.622	0.723 / 0.664	0.699 / 0.604
AB	0.864 / 0.849	0.706 / 0.660	0.830 / 0.789	0.833 / 0.769	0.700 / 0.589
GB	0.878 / 0.851	0.778 / 0.676	0.881 / 0.798	0.882 / 0.709	0.743 / 0.612
XGB	0.894 / 0.853	0.850 / 0.685	0.918 / 0.798	0.738 / 0.598	0.679 / 0.543
ENN	0.852 / 0.853	0.674 / 0.675	0.697 / 0.661	0.718 / 0.673	0.686 / 0.616

For the Framingham dataset, as shown in Table 11, ENN achieved test accuracy of 85%, the same as other models without oversampling. However, ENN showed moderate test accuracy across different oversampling techniques. With ROS, the test accuracy was 68%, while SMOTE resulted in 66%. Using B-SMOTE, ENN reached 67%, and with ADASYN, ENN performed best alongside GB with a score of 62%.

These results suggest that, while ENN is effective in some contexts, it does not benefit from oversampling techniques in the same manner as tree-based models do for this dataset. ENN's performance with SMOTE and B-SMOTE indicates a partial capacity to handle imbalanced data but falls short of the consistent improvements seen in models like XGB and GB.

Table 13 - Framingham: F-score Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.004 / 0.000	0.671 / 0.355	0.711 / 0.335	0.728 / 0.342	0.721 / 0.331
AB	0.243 / 0.177	0.712 / 0.368	0.818 / 0.280	0.826 / 0.256	0.730 / 0.331
GB	0.324 / 0.123	0.787 / 0.346	0.872 / 0.226	0.876 / 0.230	0.765 / 0.315
XGB	0.462 / 0.214	0.854 / 0.303	0.913 / 0.220	0.765 / 0.334	0.716 / 0.316
ENN	0.048 / 0.060	0.665 / 0.374	0.690 / 0.299	0.713 / 0.277	0.705 / 0.315

Without any oversampling, both ENN and RF models had the lowest performance with a test F-score of 0%. However, when ENN was combined with ROS, the F-score increased significantly to 37%, showing that it gave the best results in combination with AB. Using the SMOTE technique, ENN achieved a test F-score of 30%, which was the second-best performance for this method, following the RF model.

Table 14 - Framingham: G-mean Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.046 / 0.000	0.671 / 0.394	0.711 / 0.379	0.728 / 0.375	0.722 / 0.380
AB	0.327 / 0.231	0.712 / 0.410	0.820 / 0.280	0.827 / 0.257	0.733 / 0.385
GB	0.434 / 0.195	0.787 / 0.376	0.874 / 0.228	0.877 / 0.232	0.767 / 0.357
XGB	0.542 / 0.271	0.854 / 0.321	0.915 / 0.223	0.769 / 0.386	0.720 / 0.378
ENN	0.013 / 0.162	0.665 / 0.408	0.690 / 0.323	0.713 / 0.294	0.706 / 0.356

Without oversampling, the ENN method had a low-test G-mean score of just 16%. When ROS was applied, ENN's G-mean jumped to 41%, making it the best G-mean performer on the Framingham dataset, alongside AB. Furthermore, ENN with SMOTE achieved the second highest G-mean score after RF with 32%, but did not stand out strongly with the other methods.

In comparison, RF achieved higher and more consistent G-mean scores across the different oversampling techniques. For instance, RF reached a test G-mean of 37.5% with B-SMOTE and 40% with SMOTE. This suggests that ENN provides some benefit from oversampling (especially with ROS) but does not perform strongly overall and produces scattered results.

Table 15 – Framingham: AUC Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.501 / 0.500	0.677 / 0.645	0.698 / 0.626	0.723 / 0.630	0.698 / 0.624
AB	0.568 / 0.544	0.706 / 0.659	0.830 / 0.576	0.833 / 0.561	0.699 / 0.625
GB	0.596 / 0.529	0.778 / 0.632	0.881 / 0.550	0.882 / 0.551	0.742 / 0.605
XGB	0.650 / 0.556	0.850 / 0.590	0.918 / 0.547	0.738 / 0.628	0.677 / 0.609
ENN	0.511 / 0.512	0.674 / 0.661	0.697 / 0.587	0.718 / 0.568	0.686 / 0.605

Among the oversampling techniques, ENN performed best with ROS alongside AB, reaching a test AUC of 66%. This indicates that ROS helped ENN distinguish between classes more effectively. Other methods like SMOTE and Borderline-SMOTE also offered slight AUC improvements for ENN, but none matched the performance of ROS.

Table 16 – Statlog: Accuracy Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.876 / 0.838	0.875 / 0.852	0.892 / 0.838	0.825 / 0.873	0.861 / 0.852
AB	0.945 / 0.823	0.950 / 0.823	0.933 / 0.867	0.933 / 0.808	0.928 / 0.852
GB	0.930 / 0.838	0.941 / 0.838	0.928 / 0.823	0.933 / 0.823	0.923 / 0.823
XGB	0.975 / 0.823	0.964 / 0.823	0.968 / 0.823	0.977 / 0.838	0.961 / 0.823
ENN	0.643 / 0.632	0.727 / 0.705	0.754 / 0.720	0.816 / 0.823	0.790 / 0.764

Table 15 presents the accuracy results for various models applied to the Statlog dataset. Without any oversampling, RF and GB achieved the highest accuracy with a test score of 84%. XGB and AB also performed well, recording test accuracies of 82% while ENN has the lowest accuracy

test result. With oversampling, RF maintained its strong performance with ROS (85%), B-SMOTE (87%) and ADASYN (85%) achieving the highest test accuracies. ENN's accuracy improved significantly with oversampling by reaching the best test accuracy of 82.3% with B-SMOTE but still falls behind the tree-based models.

Table 17 - Statlog: F-score Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.852 / 0.792	0.869 / 0.821	0.888 / 0.800	0.869 / 0.785	0.841 / 0.821
AB	0.938 / 0.785	0.951 / 0.785	0.932 / 0.842	0.933 / 0.771	0.922 / 0.814
GB	0.919 / 0.800	0.941 / 0.807	0.927 / 0.785	0.931 / 0.785	0.914 / 0.785
XGB	0.972 / 0.785	0.963 / 0.785	0.968 / 0.793	0.977 / 0.807	0.958 / 0.785
ENN	0.678 / 0.675	0.776 / 0.736	0.784 / 0.732	0.825 / 0.812	0.796 / 0.764

Without any oversampling, ENN achieves an F-score of 67.5% on the test set, well below XGB's 78.5% and GB's 80%. ROS and SMOTE provided modest improvements for ENN, but still did not fully match the other models. However, ENN and XGB achieved the highest test F-scores with B-SMOTE oversampling, both reaching 81%. The F-score analysis on the Statlog dataset shows that ENN improves significantly with oversampling techniques, but not as much as the other models.

Table 18 - Statlog: G-mean Results for Models and Oversampling Techniques

Model	NONE (Train/Test)	ROS (Train/Test)	SMOTE (Train/Test)	B-SMOTE (Train/Test)	ADASYN (Train/Test)
RF	0.853 / 0.799	0.870 / 0.823	0.889 / 0.803	0.870 / 0.787	0.843 / 0.823
AB	0.938 / 0.787	0.951 / 0.787	0.932 / 0.843	0.934 / 0.773	0.922 / 0.819
GB	0.920 / 0.803	0.941 / 0.808	0.927 / 0.787	0.931 / 0.787	0.915 / 0.787
XGB	0.972 / 0.787	0.964 / 0.787	0.968 / 0.793	0.977 / 0.808	0.958 / 0.787
ENN	0.692 / 0.692	0.789 / 0.753	0.790 / 0.741	0.826 / 0.814	0.799 / 0.770

Without any oversampling, ENN had the lowest test G-mean (69%). ENN had the lowest test G-mean of 69% without any oversampling. When using B-SMOTE, ENN's G-mean improved

to 81%, matching XGB for the highest score achieved with this technique. Although this score is not the highest overall, it shows that ENN makes significant use of techniques that emphasize decision boundaries, as seen in B-SMOTE.

RF achieved the highest G-mean score of 82% with ROS, while AB achieved 84% with SMOTE. Both models achieved 82% with ADASYN. No single model consistently achieved the highest score across all techniques.

Table 19 - Statlog: AUC Results for Models and Oversampling Techniques

Model	**NONE** (Train/Test)	**ROS** (Train/Test)	**SMOTE** (Train/Test)	**B-SMOTE** (Train/Test)	**ADASYN** (Train/Test)
RF	0.868 / 0.823	0.875 / 0.843	0.892 / 0.827	0.875 / 0.814	0.857 / 0.843
AB	0.944 / 0.814	0.950 / 0.814	0.933 / 0.860	0.933 / 0.800	0.927 / 0.840
GB	0.926 / 0.827	0.941 / 0.830	0.928 / 0.814	0.933 / 0.814	0.920 / 0.814
XGB	0.974 / 0.814	0.964 / 0.814	0.968 / 0.817	0.977 / 0.830	0.961 / 0.814
ENN	0.663 / 0.657	0.727 / 0.729	0.754 / 0.735	0.816 / 0.828	0.795 / 0.775

Without any oversampling, RF, GB and AB scored well, each achieving a test AUC of around 81-83%. ENN, however, started with the lowest AUC at only 66%. With B-SMOTE, both ENN and XGB achieved the highest test AUC scores, with 83%.

5 Conclusion

This study evaluated the effectiveness of ENN as an alternative to tree-based ensemble methods for predicting heart disease, especially in the context of imbalanced datasets. The analysis revealed that, while ENN demonstrated strong performance in certain aspects, particularly in balancing sensitivity and specificity, it did not consistently outperform tree-based models.

For the Heart Disease Health Indicators dataset, ENN performed consistently well across multiple metrics. While RF slightly outperformed ENN in some cases, particularly in terms of accuracy and AUC, ENN demonstrated robust performance in handling imbalanced data, as reflected by its competitive F-scores and G-mean values. The improvement in performance with oversampling techniques, especially SMOTE and B-SMOTE, highlights ENN's ability to improve class balance, particularly in terms of sensitivity and specificity. These results suggest that ENN, when paired with effective resampling strategies, can be a valuable method for addressing class imbalance in heart disease datasets. Although it does not always outperform RF in accuracy, ENN's consistent F-scores show that it is a strong competitor in managing imbalanced classification tasks.

On the Framingham dataset, ENN showed improvements in F-score when combined with oversampling techniques, particularly ROS. However, these gains were less consistent compared to tree-based models like RF and GB, which demonstrated more stable and higher F-scores across different methods. While ENN was able to improve its F-score with the use of oversampling, it still did not match the overall performance of tree-based models in handling class imbalance, suggesting that ENN's ability to balance false positives and false negatives is less reliable in this dataset.

The Statlog dataset highlighted ENN's potential in smaller datasets with lower imbalance ratios. Without oversampling, ENN struggled, but when B-SMOTE was applied, there was a notable increase in F-score, making its performance comparable to XGB in some cases. However, despite the improvement, ENN still falls behind RF and GB in terms of F-score, accuracy, and G-mean,

indicating that even with oversampling, ENN could not match the stability and overall performance of tree-based models, especially in smaller datasets. The lower imbalance ratio in Statlog meant that class imbalance was less of an issue, but the small dataset size limited ENN's ability to fully utilize the available data and required oversampling to increase the F-score.

Overall, ENN demonstrated the ability to improve F-score when combined with oversampling techniques such as ROS, SMOTE and B-SMOTE. However, its performance was not as stable as tree-based ensemble models, which consistently achieved high F-scores, accuracy and G-means across datasets and oversampling methods. While ENN showed significant improvements in F-score, particularly with higher imbalance ratios and larger datasets, tree-based models maintained superior performance across all metrics, including F-score, and demonstrated better generalization in handling imbalances. ENN's performance was more variable, and its reliance on oversampling meant that it could not consistently outperform tree-based models, especially in terms of overall accuracy and class discrimination.

Limitations and Future Research

The major limitation of this study is the lack of extensive preprocessing. According to the results, there is a significant difference between training and test performance, indicating overfitting. This discrepancy is attributed to the absence of extensive preprocessing steps such as feature engineering and feature selection. This is a purposeful choice that aims to evaluate the baseline performance of the model in its raw form without the influence of preprocessing techniques.

These steps could have potentially enhanced the model performance and offered a deeper understanding of the models' capabilities. According to the results, the size of the dataset and the imbalance ratio significantly affected the performance of the ENN model, indicating that the results may not generalize well to datasets with different characteristics. This limitation emphasizes the need for caution when applying the findings to datasets with different characteristics.

Another limitation is the limited scope of resampling techniques examined in the study. By examining only a certain number of techniques, the research may have overlooked other methods that could have performed better. This focus limits the conclusions that can be drawn about the effectiveness of different machine learning approaches in handling imbalanced datasets.

Finally, the evaluation metrics used in the study may not capture all aspects of model performance, especially in real-world applications where aspects such as computational efficiency and interpretability are crucial. The common performance metrics applied may fail to reflect these practical aspects and thus provide an incomplete assessment of the models. This highlights the importance of including a wider range of metrics to fully assess the effectiveness of models in practical scenarios.

Future research should prioritize incorporating advanced pre-processing techniques like feature selection, feature engineering and other steps to prevent overfitting. These steps could provide deeper insights into the models' capabilities and lead to improved outcomes. Investigating various methods will help identify superior approaches for handling imbalanced datasets, potentially giving better performance.

Furthermore, developing hybrid approaches that combine multiple resampling techniques and models or creating new hybrid methods could further improve performance on imbalanced datasets. This multi-faceted approach can enhance the adaptability and effectiveness of the models.

Applying the models and techniques in real-world scenarios will provide valuable insights into their practical utility and effectiveness. This application will consider factors like computational cost, ease of implementation, and interpretability, which are often crucial in practical settings. Including a wider range of evaluation metrics, such as precision-recall curves, and cost-sensitive metrics, will offer a more comprehensive evaluation of model performance, ensuring a thorough assessment of their capabilities in various contexts.

References

Abdi, L., & Hashemi, S. (2016). To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques. *IEEE Transactions on Knowledge and Data Engineering*, *28*(1), 239. https://doi.org/10.1109/TKDE.2015.2458858

Aggrawal, R., & Pal, S. (2020). Multi-Machine Learning Binary Classification, Feature Selection and Comparison Technique for Predicting Death Events Related to Heart Disease. *International Journal of Pharmaceutical Research*, *13*(1), 428-438. https://doi.org/10.31838/ijpr/2021.13.01.080

Aher, S. B., & Lobo, L. M. R. J. (2013). Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data. *Knowledge-Based Systems*, *51*, 1-14. https://doi.org/10.1016/j.knosys.2013.04.015

Ahmed, G., Er, M. J., Fareed, M. M. S., Zikria, S., Mahmood, S., He, J., Asad, M., Jilani, S. F., & Aslam, M. (2022). DAD-Net: Classification of Alzheimer's Disease Using ADASYN Oversampling Technique and Optimized Neural Network. *Molecules 2022, 27*(20), 7. https://doi.org/10.3390/MOLECULES27207085

Ali, H., Salleh, M., Saedudin, R., Hussain, K., & Mushtaq, M. (2019, March). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science, 14*(3), 1561. http://doi.org/10.11591/ijeecs.v14.i3.pp1552-1563

Aradi, S. (2022). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems, 23*(2), 740-759. https://doi.org/10.1109/TITS.2020.3024655

Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. (2023). Effective Heart Disease Prediction Using Machine Learning Techniques. *Algorithms 2023, 16*(2), 1-14. https://doi.org/10.3390/A16020088

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST, 25*, 197–227. https://doi.org/10.1007/s11749-016-0481-7

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Centers for Disease Control and Prevention. (2015). *Behavioral Risk Factor Surveillance System survey data*. U.S. Department of Health and Human Services, Centers for Disease Control and Prevention. Retrieved from https://www.kaggle.com/datasets/heart-disease-health-indicators-dataset

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16),* 785–794. https://doi.org/10.1145/2939672.2939785

Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., & Ameen, M. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data, 2*(1), 1–24. https://doi.org/10.1186/s40537-015-0029-9

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems* (Lecture Notes in Computer Science, Vol. 1857, 1–15. https://doi.org/10.1007/3-540-45014-9_1

Dong, X., Yu, Z., Cao, W., & Zhang, X. (2020). A survey on ensemble learning. *Frontiers of Computer Science, 14*(2), 242. https://doi.org/10.1007/s11704-019-8208-z

Ebrahimi, M., Mohammadi-Dehcheshmeh, M., Ebrahimie, E., & Petrovski, K. R. (2019). Comprehensive analysis of machine learning models for prediction of sub-clinical mastitis: Deep learning and gradient-boosted trees outperform other models. *Computers in Biology and Medicine, 114,* 1–9. https://doi.org/10.1016/j.compbiomed.2019.103456

Friedman, J. H. (1999). Greedy function approximation: A gradient boosting machine. 1–39. Retrieved from https://jerryfriedman.su.domains/ftp/trebst.pdf

Gosain, A., & Sardana, S. (2017). Handling class imbalance problem using oversampling techniques: A review. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 80. https://doi.org/10.1109/ICACCI.2017.8125820

Hamza, M., & Larocque, D. (2006). An empirical comparison of ensemble methods based on classification trees. *Journal of Statistical Computation and Simulation*, *75*(8), 629–643. https://doi.org/10.1080/00949650410001729472

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In D. S. Huang, X. P. Zhang, & G. B. Huang (Eds.), *Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644,* 881–882. https://doi.org/10.1007/11538059_91

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks*, 1323–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Heaton, J. (2018). Review of the book *Deep Learning* by I. Goodfellow, Y. Bengio, & A. Courville. *Genetic Programming and Evolvable Machines, 19*(3), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Helm, J. M., Swiergosz, A. M., Haeberle, H. S., Karnuta, J. M., Schaffer, J. L., Krebs, V. E., Spitzer, A. I., & Ramkumar, P. N. (2020). Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions. *Current Reviews in Musculoskeletal Medicine*, *13*(1), 69–76. https://doi.org/10.1007/s12178-020-09600-8

Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1, 278–282. https://doi.org/10.1109/ICDAR.1995.598994

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–303. https://doi.org/10.1109/TKDE.2005.50

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Javaid, M., Haleem, A., Pratap Singh, R., Suman, R., & Rab, S. (2022). Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, *3*, 58–73. https://doi.org/10.1016/J.IJIN.2022.05.002

Kaggle. (n.d.). Framingham Heart Study Dataset. Retrieved from https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset/data

Kaur, H., Pannu, S., & Malhi, A. K. (2019). A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Comput. Surv*, *52*, 79. https://doi.org/10.1145/3343440

Khemphila, A., & Boonjing, V. (2010). Comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients. In *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 193–198. https://doi.org/10.1109/CISIM.2010.5643666

Kovács, G. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, 83, 105662. https://doi.org/10.1016/j.asoc.2019.105662

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 222. https://doi.org/10.1007/S13748-016-0094-0

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436. https://doi.org/10.1038/nature14539

Lee, H., & Cho, C. H. (2019). Digital advertising: present and future prospects. *International Journal of Advertising*, *39*(3), 332–341. https://doi.org/10.1080/02650487.2019.1642015

Maldonado, S., López, J., & Vairetti, C. (2019). An alternative SMOTE oversampling strategy for high-dimensional datasets. *Applied Soft Computing Journal, 76*, 381. https://doi.org/10.1016/j.asoc.2018.12.024

Marchese Robinson, R. L., Palczewska, A., Palczewski, J., & Kidley, N. (2017). Comparison of the Predictive Performance and Interpretability of Random Forest and Linear Models on Benchmark Data Sets. *Journal of Chemical Information and Modeling*, *57*(8), 1773–1792. https://doi.org/10.1021/acs.jcim.6b00753

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *11th International Conference on Information and Communication Systems (ICICS)*, 243–248. https://doi.org/10.1109/ICICS49469.2020.239556

National Heart, Lung, and Blood Institute. (n.d.). Know the Differences: Cardiovascular Disease, Heart Disease, Coronary Heart Disease. Retrieved from https://www.nhlbi.nih.gov/sites/default/files/publications/FactSheetKnowDiffDesign2020V4a.pdf

Ryman-Tubb, N. F., Krause, P., & Garn, W. (2018). How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence, 76*, 130–157. https://doi.org/10.1016/j.engappai.2018.07.008

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249, 1–2. https://doi.org/10.1002/WIDM.1249

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227.

https://doi.org/10.1007/BF00116037

Susan, S., & Kumar, A. (2021). The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. *Engineering Reports*, *3*(4), 1–2. https://doi.org/10.1002/ENG2.12298

Teo, K. K., & Rafiq, T. (2021). Cardiovascular Risk Factors and Prevention: A Perspective From Developing Countries. *Canadian Journal of Cardiology*, *37*(5), 733. https://doi.org/10.1016/J.CJCA.2021.02.009

UCI Machine Learning Repository. (n.d.). Statlog (Heart). https://doi.org/10.24432/C57303

Sarker, I.H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications, and research directions. *SN Computer Science, 2*, 1–5. https://doi.org/10.1007/s42979-021-00815-1

Viloria, A., Lezama, O. B. P., & Mercado-Caruzo, N. (2020). Unbalanced data processing using oversampling: Machine Learning. *Procedia Computer Science*, *175*, 108–113. https://doi.org/10.1016/J.PROCS.2020.07.018

Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *Stanford University, CS231n: Convolutional Neural Networks for Visual Recognition.* Retrieved from http://vision.stanford.edu/teaching/cs231n/reports/2017/pdfs/300.pdf

Wen, J., Zhang, X., Xu, Y., Li, Z., & Liu, L. (2009). Comparison of AdaBoost and logistic regression for detecting colorectal cancer patients with synchronous liver metastasis. In *2009 International Conference on Biomedical and Pharmaceutical Engineering*, 1–6. https://doi.org/10.1109/ICBPE.2009.5384087

World Health Organization. (n.d.). Cardiovascular diseases. *World Health Organization*. Retrieved January 19, 2024, from https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

World Heart Federation. (2023). *World Heart Report 2023: Confronting the world's number one killer.* World Heart Federation. Retrieved from https://world-heart-federation.org/wp-content/uploads/World-Heart-Report-2023.pdf

Wyner, A. J., Olson, M., Bleich, J., & Mease, D. (2017). Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers. *Journal of Machine Learning Research, 18,* 1–33. http://jmlr.org/papers/v18/15-240.html

Yadav, S. S., Jadhav, S. M., Nagrale, S., & Patil, N. (2020). Application of Machine Learning for the Detection of Heart Disease. *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings*, 165–172. https://doi.org/10.1109/ICIMIA48430.2020.9074954

Ye, J., Ni, J., & Yi, Y. (2017). Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security, 12*(11), 2545–2546. https://doi.org/10.1109/TIFS.2017.2710946

Zhang, Z., Zhao, Y., Canes, A., Steinberg, D., & Lyashevska, O. (2019). Predictive analytics with gradient boosting in clinical medicine. *Annals of Translational Medicine, 7*(6), 129. https://doi.org/10.21037/atm.2019.03.29

Zhou, Z.-H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137*(1–2), 239–263. https://doi.org/10.1016/S0004-3702(02)00190-X

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa