POLITECNICO DI TORINO

Master's Degree in ICT for Smart Societies

Masters's Degree Thesis

Interpretable Machine Learning-Based Algorithms for Cardiac Anomaly Detection

Supervisors	Candidate
Prof. MONICA VISINTIN	HAMED MIRZAKHANI
Dr. ISAAC SHIRI	

October 2024

Summary

Cardiovascular diseases (CVD) are the foremost cause of mortality worldwide, responsible for approximately 17.9 million deaths annually, as reported by the World Health Organization(WHO). In order to effectively manage cardiac disease, which leads to decreased morbidity, the essential step is early detection of the disease. The main aim of this study is to look in more detail at the performance of supervised classification machine learning algorithms to detect cardiac anomalies from rest/stress myocardial perfusion imaging (MPI) in single-photon emission computed tomography (SPECT). A total of 266 patients who performed a 2-day stress-rest protocol MPI SPECT were suspected to have a cardiac abnormality. Altogether, 401 features were extracted from Rest-Stress MPI SPECT images. These features included different sets, consisting of Rest-Stress, Delta, and combined-radiomics (a combination of all sets of features). To have training and testing parts, the data was randomly divided into subsets of 75% and 25%. To evaluate the performance of classifiers, combinations of three scaling techniques, four feature selections, nine classification algorithms, and two search strategies were used. For the purpose of evaluating the model, different metrics consisting of Specificity (SPE), Sensitivity (SEN), Accuracy (ACC), and Area Under the ROC curve (AUC) were measured. Having been considered, models built of the combination of Rest-Stress feature set performed better than models of Rest and Stress. The metrics results were ACC = 0.83, AUC = 0.86, SPE=0.81, and SEN = 0.81 for the RobustScaler scaling method, the Logistic Regression (LR) classification algorithm with selected features from the Model-based Feature Selection (FS), Random Search for parameter optimization, and outliers were detected with Z-Score strategy. For models with the highest performance, interpretability model was implemented. Shapley values of features are calculated to find the features that have the highest effect on the output result. As an example, the “Zone distance variance” feature from the Gray-Level Zone Size Matrix (GLSZM) values, significantly impacted the final result. The goal is to identify features of high importance rather than focusing on a wide range of features and spend time and energy on the features that are more important so that the results can be achieved faster and easier.

Acknowledgements

I am grateful to all those who have guided and supported me in achieving my life goals. My heartfelt thanks go to my brother **Mohsen** and my dear sister **Fatemeh** for their unwavering support and kindness through all the ups and downs of life. I would also like to sincerely thank Professor **Visintin** and Dr. **ShiriLord** for their generous guidance and patience throughout this journey. Their mentorship has been invaluable in the completion of this work.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

DL
Deep Learning

ML
Machine Learning

CVD
Cardiovascular Diseases

WHO
World Health Organization

MPI
Myocardial Perfusion Imaging

SPECT
Single-Photon Emission Computed Tomography

GLZSM
Gray-Level Zone Size Matrix

SEN
Sensitivity

SPE
Specificity

ROC

Receiver Operating Characteristic

AUC

Area Under Curve

FS

Feature Selection

MRI

Magnetic Resonance Imaging

ECGs

Electrocardiograms

DP

Deep Learning

LR

Logistic Regression

K-NN

K-Nearest Neighbor

SVC

Support Vector Classification

RF

Random Forest

IML

Interpretable Machine Learning

NN

Neural Network

xAI

Explainable AI

LIME

Locally Interpretable Model-agnostic Explanations

ACC

Accuracy

DT

Decision Tree

XGBoost

Extreme Gradient Boosting

LVH

Left Ventricular Hypertrophy

CAD

Cronary Artery Disease

IQR

Interquartile Range

PCA

Principle Component Analysis

TP

True Positive

TN

True Negative

FP

False Positive

FN

False Negative

CM

Confusion Matrix

FPR

False Positive Rate

TPR

True Positive Rate

PR

Precision-Recall

SHAP

Shapley Additive exPlanation

PDP

Partial Dpendent Plot

PFI

Permutation Feature Importance

PAD

Peripheral Artery Disease

Chapter 1

Introduction

1.1 What are Cardiovascular Diseases?

Health disorders in any part of the heart or the blood vessels come under cardiovascular diseases (CVDs) [1]. The list of disorders that are included under the category of CVDs is very long, containing conditions such as heart failure, hypertensive heart disease, rheumatic heart illness, cardiomyopathy, arrhythmia, congenital heart problems, valvular heart disease, symptoms of carditis, aneurysms, peripheral artery disease (PAD), thromboembolic disease, and venous thrombosis [2]. CVDs can also apply to any malfunction or structural problem occurring within the heart valves. Disease of heart valves, particularly, manifests when the valves of the heart are not able to render an appropriate opening or closing; this consequently alters the way blood flows [3].

1.2 Scenario

Cardiovascular Diseases (CVDs) remain the leading cause of death globally [4], accounting for an estimated 17.9 million deaths per year according to the World Health Organization [4]. Unhealthy eating, overweight and obesity, a sedentary lifestyle, and chronic diseases such as diabetes are remarkable causes of heart disorders [4]. To effectively manage cardiac disease, which leads to a reduction in mortality and also ultimately lowers the morbidity linked to these disorders, the essential step is early detection of the disease [4]. Often, the basic symptoms are similar to those of other diseases, making it difficult for medical professionals to provide an accurate diagnosis for different cardiovascular diseases [4]. The identification of cardiovascular diseases is greatly aided by medical imaging. In order to progress biomedical research and enable early detection of illnesses in

treatment, imaging is crucial, which it can be done in non-invasive way [5]. Key non-invasive imaging tools for CVDs assessments include Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), and Single Photon Emission Computed Tomography (SPECT) [6]. CT scans generate high-resolution, cross-sectional anatomical images by processing multiple X-ray measurements from different angles, which are valuable for diagnosing cardiovascular diseases [7]. MRI provides detailed three-dimensional images without ionizing radiation, making it ideal for assessing soft tissues [8]. Imaging techniques to detect abnormalities, such as CT and MRI, are popular in medical diagnostics. CT scan is a medical imaging technique that forms detailed images of the interior part of the body [9]. Then, a computer processes several measures of X-rays taken from different angles to form images that can be stacked to create a three-dimensional model of the scanned area [7]. CT scans provide high-resolution imaging and a flexible imaging method used to diagnose heart diseases [9]. MRI is another imaging technique that maintains a high ratio of signal to noise and offers a three-dimensional detailed image of the body [8]. MRI is free from ionizing radiation, which produces fine images of the soft tissues, organs, and interior structures [10]. Nuclear medicine techniques, such as PET and SPECT, are crucial for detecting early physiological and metabolic changes, often before structural abnormalities become evident [6]. These techniques produce three-dimensional images, enabling detailed analysis of blood flow and metabolic activity analysis [6].

In both modalities, gamma rays emitted from radioactive tracers inside the body are captured to provide insight into the physiological and metabolic activities [11]. PET and SPECT are useful for research in health at the molecular level [12]. Also, one of the best approaches to discovering diseases and monitoring treatment procedures [13]. One of the above imaging modalities may be chosen depending on the exact clinical scenario in hand, information needed, patient characteristics, and available resources. Particularly, PET and SPECT imaging can detect physiologic metabolic changes at the cellular level, allowing it to discover diseases in their early stages well before changes on a structural level occur [12]. Diseases are most often diagnosed by CT and MRI based on structural abnormalities, which might occur later in the disease process [14]. SPECT imaging faces challenges such as the need for expert interpretation, time-consuming analysis, and variability between observers [15]. Radiomics, with its mathematical approach, addresses these issues by standardizing image interpretation, reducing variability, and minimizing the need for extensive expertise [16]. In healthcare, radiomics is an approach that uses data-characterization algorithms to collect considerable amounts of features from medical images [17, 18]. Integrating machine learning into radiomics enhances diagnostic accuracy, but ensuring model interpretability is crucial for clinical trust and effective patient care [19].

In the past few years, there has been a significant increase in the use of many

Artificial Intelligence (AI) (Machine Learning (ML) and Deep Learning (DL)) in different fields such as text detection and recognition [20]. In ML, models are trained using various probabilistic and statistical techniques on data to identify and learn different patterns, which are subsequently applied to new data to produce desired predictions [21]. Nowadays, ML appears to be utilized more often and has become increasingly important in health [20]. With ML algorithms, it is now potentially possible to recognize diseases with high precision and accuracy [22].

This study uses SPECT imaging data and machine learning models to enhance diagnostic accuracy, addressing limitations in traditional imaging methods and facilitating earlier detection and monitoring of diseases. In this study, we aimed to implement different ML algorithms, and find the best-performing model and make it more interpretable for clinical implementation. Upon identifying the optimal model, the next step is to comprehend its operation.

1.3 Interpretability

Although ML models perform remarkably well, their lack of clarity is a major obstacle to their wider use in clinical practice [23]. This lack of transparency is a challenge in medicine, where understanding the decision-making process is crucial for gaining the trust of healthcare professionals and patients, ensuring regulatory compliance, and facilitating accurate diagnosis [24]. Instead of only predicting and focusing on the final results, the interpretation methods provide an interface that offers additional details. The goal of Interpretable ML (IML) is to fill the gap that exists between the requirement for transparency and the performance of complex models [25]. Some models are inherently interpretable and have self-explain abilities like Decision Trees (DTs), Linear Regression, and K-Nearest Neighbor (K-NN), while many of the most advanced models like the Random Forest (RF) model or Neural Networks (NN) and Deep Learning (DL) models are not easily understandable for humans, which are referred as “black-box” models [26]. Interpretability tries to understand how these models work and make them much more understandable for the end user [26].

1.4 Explainable AI (xAI) methods

Explainable Artificial Intelligence (xAI) initiatives have two main objectives. The first one is to develop methods of machine learning that remain highly performing in learning, with models allowing their decision-making process and output to be understood [26]. The second objective has to do with communicating a user-centric approach to make artificial intelligence understandable for humans [26]. That’s why

xAI tries to provide more confidence in learned models and effective collaboration with artificial and human agents[27].

Generally, xAI methods can be divided into two main types: post-hoc and ante-hoc methods[28]. Post-hoc is used to provide explanations for the model after it has been trained [26]. Then, the ante-hoc xAI refers to the incorporation of interpretability into the ML model development process from the very beginning [26]. This means that the model is designed to be interpretable from the start, rather than attempting to make an opaque model interpretable after it has been trained [29].

Model-specific and model-agnostic are two sub-classes of post-hoc methods [28]. Model-specific methods are designed for specific model categories using specified characteristics of the models to explain the behavior of the model [26]. Another interpretability strategies are Model-agnostic techniques. These models are flexible and can be applied to all ML models, due to their internal structures [26]. The common and popular techniques are Shapley values, Locally Interpretable Model-agnostic Explanations (LIME) [30], and Permutation Feature Importance (PFI) [31]. Shapley values can be useful in explaining the output of any ML model [31]. These values help understand each feature's effect on model output and identify the most important features [31]. This approach allows one to focus on the most important features.

1.5 Research Questions

This study plans to address the following questions about the context of using AI on healthcare data:

1. **How successfully machine learning detects anomalies:** How well do machine learning models identify anomalies in cardiac diseases?
   - To investigate this, several machine learning models were deployed and trained. The main objective is to assess their output results regarding accuracy, specificity, sensitivity, and other metrics.

2. **Performance comparison between different designed classifiers:** Which classifier or combination of classification algorithms, feature selectors, and scaling methods has better results?
   - The goal is to find the classifier with the best performance on data, and which features have been selected and have had the greatest impact on the way the model works and output.

3. **Effectiveness of interpretability approaches on the final results and features:** Why this stage is important, while machine learning models are implemented on data, and results are obtained?

- This step of the study provides more insightful and accurate explanations of different models' decisions and identifies how each feature impacts the final result. Given that sometimes specialists have to work on a large number of features, and this work certainly requires time and high accuracy, interpretability output makes it possible to focus on the features that have the most impact on the results.

1.6 Thesis Overview

This study investigates implementing different ML algorithms on extracted features of MPI SPECT images to identify the classifier with the best performance, and then with a Shapley Additive eXplanations (SHAP) method, finds the features with the highest effect of final prediction.

1. **Chapter 1, Introduction:** The first chapter presents an overview of heart diseases and diagnosis approaches. It also talks about the use of machine learning models in the medical domain, and how can increase transparency with interpretable ML.

2. **Chapter 2, Literature Review:** A summary of previous studies in cardiac anomaly detection. In addition, in this chapter explanations are provided about the implemented machine learning algorithms and interpretability methods and the obtained results.

3. **Chapter 3, Data Preprocessing:** Chapter 3, explains the data, as well as the used method for feature extraction from MPI SPECT images and the software to do this task. Also, the algorithms used in the pipeline for feature selection and outlier detection and definitions of the measured metrics, explanations, and formulas are provided.

4. **Chapter 4, Model Optimization:** In chapter 4, creating a pipeline and also the main search methods are explained. Furthermore, the algorithms to optimize parameters and hyperparameters are explained. This chapter also presents an interpretability explanation and how Shapley values are calculated, and a summary of how SHAP plots can help to understand the rule of each feature in the final prediction.

5. **Chapter 5, Results of ML Models:** In this chapter, the output results of implementing the designed classifier on the test dataset are illustrated for all combinations of ML algorithms, scaling methods, and feature selection strategies. In addition, the results of the top classifiers of each combination and ROC curves are shown in tables and figures. The comparison between results can be found in this chapter.

6. **Chapter 6, Result of Interpretability Model:** Chapter 6 presents the output SHAP plots of implementing the interpretability model on selected ML models with the highest value of metrics and explains each plot.

7. **Chapter 7, Conclusion and Future Work:** The final chapter summarizes the study findings and proposes new topics to investigate more in future works.

Chapter 2

Literature Review

2.1 Literature Review

This study includes two main sections. In the first section, different ML algorithms are implemented, and their results are compared. Then, the model with the highest output metric values is chosen. The second part focuses on implementing a Shapley values explainer on the selected model and the features that are selected in the first step for each combination of the models. ML classification algorithms are commonly used for prediction, and a variety of research has been conducted on cardiac disease and anomaly detection [32, 33, 34]. The output results, such as Accuracy (Acc), Sensitivity (SEN), Specificity (SPE), Area Under the Curve (AUC), and other metrics of ML models in different fields, are commonly good, but a lack of transparency of ML models is evident [35]. Different interpretability methods are used to shed light on the different ML models.

2.2 ML algorithms in the health

The healthcare domain is rich in data due to various monitoring and data collection tools, and ML has enormous significance in analyzing data that is impossible for humans to process effectively [36]. ML is the process of learning patterns and useful information from data [36]. The trained ML model could be used to predict the result for new cases, which allows medical professionals to identify new patients more effectively [37].

Identifying heart diseases is one of the most important tasks, in the medicine [38]. To save lives, a quick, accurate, and efficient diagnosis tool is required [38]. In common methods, multiple tests must be performed on patients, and they need to be checked by specialists [38]. Because of this, researchers are interested in predicting the risk of heart disease and detecting abnormalities by creating various prediction

and detection models using ML techniques [20]. Different classification techniques are used for abnormality detection, and a brief definition of each technique will be provided. All implemented classification algorithms in this study are imported from scikit-learn [39] library.

2.2.1 Summary of used algorithms

Logistic Regression

Logistic Regression (LR) is a supervised ML method used for binary classification tasks [39]. LR can be employed to predict the likelihood of developing a specific illness using observable patient characteristics [39, 40]. It predicts the probability that a given input belongs to a particular class [39, 40]. LR uses a logistic function, basically a sigmoid function, to model binary output [39, 40].

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

x is the input

LR does not require a linear relationship between inputs and target variables [39, 40]. Kannan. et al. [41], used different ML algorithms to predict Cardiovascular disease (CVD), and LR achieved the best accuracy of 86.51% compared to other models.

K-Nearest Neighbor

K-Nearest Neighbor (K-NN) is a simple supervised ML algorithm used for classification and regression problems [39, 42]. It works based on Euclidean Distance and the concept of neighborhood. In essence, to classify a new data point, K-NN examines k neighbors, in the training set and calculates the distances between the test point and these neighbors [39, 42]. Then it sorts the distance and assigns test points to the class that has the majority between all classes. The value of 'k' is defined by the user, and it is a positive integer value [39, 42]. The value of accuracy and other metrics can be changed by changing the value of 'k'. K-NN has always been used in different classification problems to measure defined model performance. Shah et al. [43] implemented various ML algorithms on the existing dataset of the Cleveland database of UCI repository to predict heart disease. Among all ML algorithms, K-NN had the highest performance with an accuracy of 90.78% in predicting heart anomalies.

Decision Tree

Decision Tree (DT) is a supervised ML method that works for both classification and regression tasks[39, 44]. With DT, tree-like structures are created. Created tree with this method consists of a root node (main node), interior node (handle different attributes), and leaf node (output or class label) [39, 44]. DT divides the data into two or more sets based on the most important indicators [39, 45]. At first, the entropy of each feature is calculated and data division is done based on maximum gain or minimum entropy [44]. As this algorithm analyzes data in a tree-shape structure, it is possible to have better results in different metrics and also accuracy values in comparison to other models [39, 45]. Emil, et al. [46] focus on tree-based algorithms to predict heart disease in patients. Based on published results, DT could predict patients with heart disease with 81% accuracy, and with 82% accuracy of patients without heart problems.

Random Forest

One well-known ensemble classification method used in ML across various applications, especially in healthcare, is Random Forest (RF) [39, 47]. It is used in classification and regression tasks. This method fits multiple DT classifiers in parallel using a technique called "parallel ensembling" [39, 47]. As a result, it reduces the overfitting issue and improves accuracy [39, 47]. Siddiqui, et al. [48], used different ML algorithms to construct and explore models to predict coronary illness based on different attributes, and between all algorithms, RF achieved 95.8% accuracy

Adaptive Boosting

Adaptive Boosting (AdaBoost) is an ensemble learning technique that employs an iterative procedure to make better classifiers by learning from their mistakes and errors [39, 49]. It has a base learner that trains in each iteration. The procedure of updating weights is according to the performance of earlier iterations [39, 49]. Instead of parallel ensembling like RF, it uses "Sequential Ensembling". This approach creates a robust classifier with a high accuracy and enhances the classifier efficiency [39, 49]. This method is sensitive to noisy data and outliers, but it remains a preferred choice for boosting the performance of DTs [50].

Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an ensemble learning model, and it builds a final model from a series of individual models, usually DTs [39, 51]. Gradient boosting employs gradient descent to minimize the loss function, similar to what

happens in NN [39, 51]. XGBoost is a specific type of gradient boosting that prioritizes precise approximations when selecting the optimal model. XGBoost can handle large datasets and is also fast to interpret [39, 51].

Support Vector Classification

The main idea of support vector classification (SVC) is to look for the best decision boundary, or hyperplane, in feature space that separates different classes [39, 52]. Some basic concepts that should be considered are: hyperplane, in an n-dimensional space, a hyperplane is the set of points x that solve the equation $x.w=0$, where w is a vector. The second concept is margin, which is the distance from the hyperplane to the nearest data point (support vectors) of either class [39, 52]. These nearest points are termed support vectors. The aim of the algorithms is to maximize margin. It means that the hyperplane should be very far away from the support vectors of both classes [39, 52].

2.2.2 Interpretability in Cardiac Anomaly Detection

While models with higher accuracy are generally more desirable, the created ML classifier can be interpretable. Oftentimes, models with more complexity have higher accuracy, while they are not easily explainable [26]. The interpretability of models varies widely. A more interpretable model simplifies understanding predictions for end users [24]. Interpretability methods can be classified as model-agnostic and model-specific [28]. Methods that are flexible enough to be used with any ML model, regardless of its underlying structure, are known as model-agnostic [28]. Shapley Additive exPlanations (SHAP) [53] and LIME [53] are some methods that examine how each feature affects the output value. Ibrahim et al. [54], after implementing XGBoost, utilized Shapley values to identify features that have the highest contribution to the classification output and tried to demonstrate IML in CVD prediction. In another study [55] on cardiovascular diseases and detecting anomalies in left ventricular hypertrophy (LVH), various supervised ML methods were implemented, which RFs showed the best results. They implemented SHAP approach for model interpretability, and the results show a considerable impact of specific features on each patient separately as well as the interactions between each feature pair [55].

Chapter 3

Data Processing

3.1 Data Collection

The dataset utilized in this study is a proprietary dataset, specifically developed for the purposes of this research. Due to its proprietary nature, it is not publicly available. The data consists of 266 patients suspicious of coronary artery disease (CAD) under the stress-rest protocol single-photon emission computed tomography (SPECT) myocardial perfusion imaging (MPI). Conventional MPI SPECT evaluates the presence, and degree of myocardial ischemia or stroke. SPECT is acquired in rest and after physical activity or pharmacological stress [56].

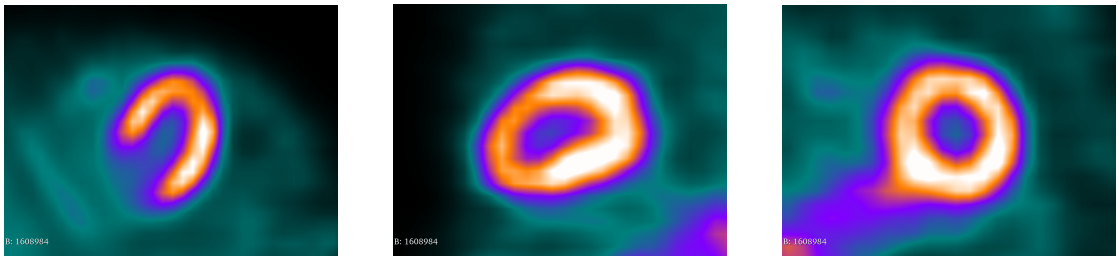

Figure 1: Normal, MPI SPECT images without any defect.

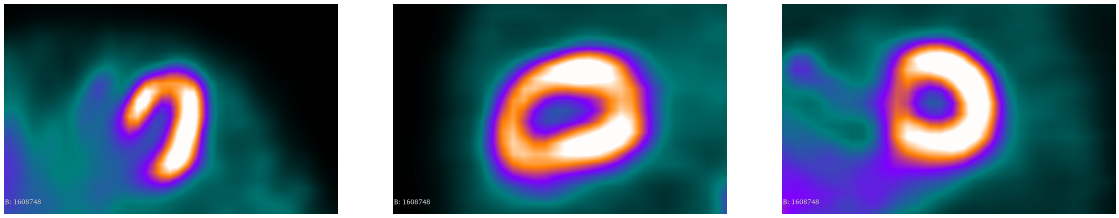

Figure 2: Abnormal, MPI SPECT images containing the defect.

3.2 Image Segmentation

A nuclear medical physicist with over five years of experience, segmented the left ventricle myocardium, using the 3D-slicer software package. An experienced nuclear medicine physician approved the segmentations and edited them if required. The image of myocardium from different aspects was evaluated and a 3D shape was created Figure 3.

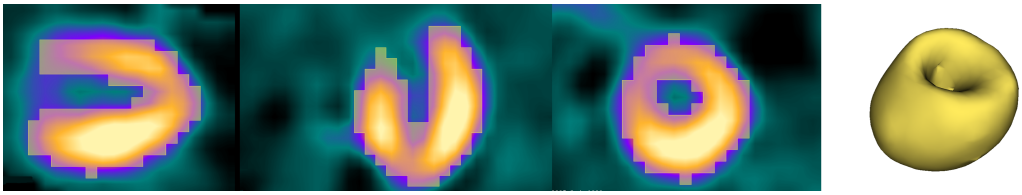

Figure 3: Segmentation

3.3 Feature Extraction

Methods that can extract features are various. For example, AI-based radiomics are either using DL strategies which may automatically learn from features that are represented in data, or engineered hard-coded features, which frequently require specialist domain and knowledge [57].

A total of 401 radiomics features were extracted from rest and stress MPI SPECT images using the Visualized & Standardized Environment for Radiomics Analysis (ViSERA) [58] software. There were 203 patients have abnormal results both in stress and rest protocol MPI SPECT. Three different data were created by combining the extracted radiomics features from MPI SPECT. Data of all patients in rest, stress, and rest-stress MPI SPECT. All the procedures are evaluated on each file separately.

3.4 Data Preprocessing

The first step of creating a ML model is data pre-processing. This phase is crucial as it enhances data quality and refers to all the actions such as data cleaning, imputation of missing values, data normalization, and feature selection performed on the data before applying ML algorithms. In the real world, data has many issues; it can be noisy, contain errors, duplicate values, or miss some elements of information that can affect ML performance. Preprocessing is time-consuming and constitutes the most complex aspect of working with data.

3.4.1 Data Cleaning

In this step, data errors and anomalies are detected and fixed. Data cleaning manages missing values, outliers, and duplicates. First, the number of missing values in each column and row is counted, and then based on the defined strategy, an action is done on the data. In this study, a threshold is defined, and if the number of outliers passes the threshold, the whole column or row is eliminated. To handle duplicate data problems, every dataset is reviewed and then the first value is retained, and other values are deleted by the defined procedure. A model's output can be negatively impacted by instances where all values or a significant portion of a column or row, are zero. In these cases, the columns and rows are removed according to a predefined threshold.

Outlier Handling Strategy

Data points that vary a lot from the dataset, and those that are uncommon in the neighborhood, are considered outliers. By identifying and discovering outliers in the dataset, users can learn more about abnormal patterns, and make a decision to solve the problem by removing errors from the data [59]. Various methods for outlier detection have been proposed. These methods can be classified based on how they describe normal data or outliers in categories like distance-based in high-dimensional data [60], ensemble techniques [61], statistical [62], and density-based [63]. Methods based on statistics suppose that normal data follow a particular distribution, and data points with significant deviations from this distribution are called outliers [64]. In this study, two statistical criteria are considered: Interqurtile Range (IQR) [65] and Z-Score [66].

IQR is a measure of statistical dispersion or distribution of data points in a dataset [65]. Although it provides a description of the distribution of the data, it is mostly unaffected by the non-parametric and non-normality of the data. The dataset is divided into quartiles to calculate the IQR value [65]. Quartiles are denoted by Q_1 (lower quartile), Q_2 (median of the whole data), and Q_3 (upper quartile).

$$Q_1 = \{q \in \mathbb{R} : P(x \leq q) = \frac{1}{4}\}$$

$$Q_2 = \{q \in \mathbb{R} : P(x \leq q) = \frac{1}{2}\}$$

$$Q_3 = \{q \in \mathbb{R} : P(x \leq q) = \frac{3}{4}\}$$

While x is the random variable.

$$\text{IQR} = Q_3 - Q_1$$

All observations below $(Q_1 - 1.5 \times \text{IQR})$ or above $(Q_3 + 1.5 \times \text{IQR})$ are detected as outliers.

The Z-Score method works properly on data that has a normal distribution. Z-Score calculates how far a point is from the mean of the dataset [66]. When a data point deviates significantly from the mean value, it means that the value of the Z-Score of that point is high, and it may be an outlier [66]. To calculate the Z-Score for each data point, first, for each numeric feature in the dataset, mean (µ) and standard deviation (σ) are calculated, and based on these values, the Z-Score is calculated [66].

$$Z = \frac{X - \mu}{\sigma}$$

A threshold is defined to qualify a data point as an outlier. Default and common thresholds are 2, 2.5, and 3. In this thesis, the threshold is set to 3.

$$\text{Outliers} = \{X_i \mid |Z_i| > \text{threshold}\}$$

Without requiring any further steps, Z-Scores can be calculated using the *SciPy* [67] library in Python. After detecting anomalies in the dataset, a decision can be taken to eliminate or handle them by using a specific method. In this study, outliers are eliminated after recognition.

Techniques to handle Missing Values in datasets

To handle missing values in datasets, different techniques can be used. As previously mentioned in this thesis, if the number of missing values in a column or row exceeds the defined threshold, it is possible to eliminate the entire row or column. The procedure of setting a value to instead of missing data is called imputation. To choose a suitable imputation method, it is necessary to understand the dataset, the missing data mechanism, and the missing values [68]. One popular approach is to substitute the missing values with the 'mean' value of the variable, measured on the valid measurements [68].

A particular imputation method is called univariate and uses purely non-missing values from the i-th feature dimension to impute values [68]. SimpleImputer [39], which is available in the scikit-learn [39] library, can be used to impute missing values in a dataset. Missing data can be imputed by a constant value or by a statistical approach like 'mean', 'median', or 'most frequent' values of the columns where missing values are located. If the data is numerical, using the 'mean' can

have a better result [68]. In this study, because all data is numeric, the 'mean' is used to impute data.

3.4.2 Data Scaling

Scaling means modifying data in a way that they fit into a given range. This step is essential since the scale of data affects the performance of many ML algorithms [69]. Models, specifically those based on distance, might perform poorer, especially in high-dimensional datasets when the data distribution is not normal [60]. Data should be scaled to prevent certain features from dominating, only because of their value. The scaling methods used in this study are Standardization (Z-Score Normalization), Min-Max Scaling (Normalization), and Robust Scaling [39]. A class is defined for different types of Scaling methods, and this class will used in a pipeline for training the model. Here are some brief explanations of each method. In this thesis, different strategies are implemented using the scikit-learn [39] library.

MinMax Scaling

Min-Max normalization transforms the value of features in a specified range, usually between 0 and 1 [39]. This method maintains the relative relationships between features, and it is possible to use this method when features are not distributed regularly like Gaussian distribution [39, 70]. For each feature:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

where X is original value, $X_{\min}$ is minimum value of the feature in the dataset, $X_{\max}$ is its maximum value, and X' is the normalized value.

Standardization (Z-score Normalization)

Z-Score normalization is a scaling method that transforms data to have a mean equal to zero and a standard deviation of 1 and is suitable for data that has a distribution like Gaussian or data normally distributed [70, 39].

$$X' = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

Robust Scaling

Robust Scaling method uses the median and the IQR for scaling features. It is suitable when data contains outliers [39, 70]

$$X' = \frac{X - \text{median}(X)}{\text{IQR}(X)}$$

where X is original feature, $median(X)$ is its median value of feature, IQR(X) is its interquartile value of feature and X' is the new scaled feature [39, 70].

3.4.3 Feature Selection

Feature Selection allows one to choose the most important features from a dataset with a high number of features [39]. It helps to avoid overfitting and enhances the model's performance. To implement a feature selection/dimensionality reduction method, it is possible to use scikit-learn modules [39]. It contains several methods, some of which are used in this study, and are described in the next sections.

SelectKBest

SelecKBest [39] is a filter-based technique that selects features without focusing on a particular ML algorithm. The features are selected and scored using statistical measures (e.g., chi2, ANOVA F-value) [39]. ANOVA is one of the most frequently applied methods for analyzing and extracting information using the P-values that are related to it [39]. It is robust because all the sample sets are assumed to be normally distributed with equal variance, and all data points or samples are independent [39, 71]. In this study, the scikit-learn [39] library has been used to implement ANOVA F-value on features[39]. Subsequently, SelectKBest selects K features with the highest scores included in the final feature subset[39].

1. At first it defines a set of features and the target variable y:

$$F = \{f_1, f_2, \ldots, f_n\}$$

   n: total number of features

2. To calculate the relevance of each feature f_i to y, it uses a scoring function S (ANOVA F-value). $s_i = S(f_i, \text{y})$. s_i: Score of feature f_i

3. Sorts scores in descending order. $s_{\pi(1)} \geq s_{\pi(2)} \geq \ldots \geq s_{\pi(n)}$, while $\pi(i)$ is the index of the i-th ranked feature

4. Select top k according to ranking. $F_k = \{f_{\pi(1)}, f_{\pi(2)}, \ldots, f_{\pi(k)}\}$

SelectPercentile

In "SelectPercentile" [39], features are scored using a function (ANOVA F-value), and selected according to their scores about a given percentile: the top features that fall into the highest percent of all feature scores are retained. The steps are the same as SelectKBest. After scoring for n features, the top $p\%$ of features are selected. if $k = \left\lceil \frac{p}{100} \times n \right\rceil$, the features with the highest score of k are selected [39].

Select From Model

"Select from Model" [39] is a feature selection technique that needs a model (mostly a supervised learning estimator) to identify important features and select them. This method exploits the capability of a model to rank features by their importance and pick up the top ones [39].

1. It defines a set of features and the target variable y:

$$F = \{f_1, f_2, \ldots, f_n\}$$

   where n: is the total number of features

2. Using the feature set F and target variable y, it trains a model (M)

3. It extracts feature importances I from model $I = \{I_1, I_2, \ldots, I_n\}$, I_i is importance score for feature f_i

4. It defines a threshold (t) for feature selection. The threshold is defined by the user as a constant number

5. Finally, it selects the feature with an importance score value greater than the threshold

$$F_{\text{selected}} = \{f_i \in F \mid I_i \geq t\}$$

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [39, 72] is a technique to reduce the dimensions of data by transforming original features into new ones, that are uncorrelated and called principal components [39]. It involves several steps: data standardization, variance computation, and eigen decomposition to find eigenvalues and eigenvectors, and then choosing the top components (the ones with the highest value) to transform the data [39].

3.5 Metrics

The main goal of this section is to provide an overview of the metrics, that assess how well the chosen supervised ML algorithms are successful in determining heart anomaly detection. The metrics are based on the number of True positive, True negative, False positive, and False negative cases[73].

1. **True Positive (TP)**: The number of instances that the model correctly predicts as positive results. The value of TP corresponds to the number of cases that are classified as abnormal, being truly abnormal [39].

2. **True Negative (TN)**: The number of instances that the model correctly predicts as negative results. The value of TN corresponds to the number of cases that are classified as normal, being truly normal [39].

3. **False Positive (FP)**: The number of instances that are negative, but the model corresponds by mistake as positive. An indication of cases that are classified as abnormal while the correct value is normal [39].

4. **False Negative (FN)**: The number of instances that are positive, while were mistakenly classified as negative. Abnormal outcomes are classified as normal [39].

In this study, the main metrics are considered as follows:

1. **Accuracy (ACC)**: Ratio between the number of accurately predicted instances and the number of all the instances [39]. To evaluate the performance of a model, focusing on just accuracy may lead to bias, especially when the data is imbalanced [39].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision (PR)**: It indicates the ratio of true positive predictions in comparison to all positive forecasts [39].

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Sensitivity (SEN)**: also called Recall or True Positive Rate (TPR), the ratio of positive instances that are predicted as positive [39].

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

4. **Specificity (SPE)**: Specificity also called True Negative Rate. It gives the ratio of negative instances that are correctly detected by the model [39].

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5. **Receiver Operating Characteristic (ROC)**: The ROC curve is a tool to evaluate the performance of detection classification models. It shows the True Positive Rate (TPR) versus the False Positive Rate across different threshold levels. The best point for the ROC curve is the top-left in the plot, which represents a better classifier and high TPR and lower FPR [39].

6. **Area Under Curve (AUC)**: The AUC is the area under the ROC curve. This value summarizes the model's performance based on different thresholds. In binary classification models, the AUC value 1 corresponds to an excellent classifier [39]. If AUC = 0.5, it means that the classifier is doing random guessing. AUC < 0.5 means that the classifier is not working well. To have the best value for Sensitivity and Specificity, it is necessary to find the optimal threshold, which in this project, is done by Youden Index [74].

$$J = \max_t \{\text{sensitivity}(t) + \text{specificity}(t) - 1\} = \max_t \{q(t) - p(t)\}$$

where t is any threshold value

7. **F1-Score**: The value of F1-score comes from combination of precision and recall. It provides a better understanding of a model's performance. Calculating the balanced average of PR and Recall, makes sure both FP and FN values are taken into account [39].

$$\text{F1-Score} = 2 \cdot \left(\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

In this study, the scikit-learn [39] library has been used to measure the mentioned metrics faster and more accurately [75]. The value of metrics is calculated for both train and test parts of data in a designed separate function. An independent function has been designed to calculate different metrics for the train and test part of the data. In this function, necessary metrics for accuracy, confusion matrix, area under the curve, and mentioned metrics are called. Finally, all the results are stored in a file to use in the plotting section.

Chapter 4

Model Optimization

The thesis goal is to test all the methods described in chapters 2 and 3 and find the best model based on values like Accuracy, AUC, Specificity, and Sensitivity.

4.1 Pipeline

In ML, a pipeline [39] is a system that sequentially applies selected methods/transformations to the data. For example, a pipeline might implement preprocessing, feature selection, training the model, and model evaluation. Hyper-parameter tuning is considerably simpler when using methods like cross-validation inside pipelines. In addition, it is possible to use of different Optimization techniques like grid search and random search in a pipeline.

4.1.1 Create a Pipeline

This study discusses a pipeline constructed in a separate class, with input parameters and values from other classes by several imputation, scaling, transformers, and estimators that are tested on input data.

For the pipeline, a specific class is created at the beginning. Then the pipeline method is called from the scikit-learn [39] library. The goal of this pipeline is to apply a list of transformers and a final estimator. At first, to handle missing values, the imputation class is called which offers a list of possible strategies ('mean', 'median', 'most frequent'), that in this study, 'mean' has been selected [39] . After imputation, the input data should be scaled. For this purpose, a scaling class that contains different scaling strategies is defined and called. Scikit-learn offers some methods of the preprocessing library for scaling data properly, such as Min-Max scaling, Standard scaling, and Robust scaling [39] . The next step involves using transformers . To accomplish this, an independent class is defined that consists of

different feature selection algorithms, and also the number of features that should be selected is determined which in this study is set to 35 features. Sklearn library provides different methods for feature selection [39] . In this thesis, SelectKBest, Select Percentile, Select from Model, and PCA are called and used [39] . Finally, a component of the pipeline is the estimator. Various estimators have been included in a specific class which fit on data in the last step.

4.2 Main Search Methods

Data is divided into training and test parts in percentages of 75% to 25% by stratification. To train the model, a search method has been designed. In this method, two possible search algorithms are used to train various models and optimize parameters: Grid Search and Random Search.
The procedure works such that in each iteration, the imputation strategy is set to 'mean', and for the scaling method, puts one of the defined methods and then takes one of the feature selection methods, and finally takes one of the classification algorithms. This entire procedure is encapsulated within the pipeline. For cross-validation, ShuffleSplit is used, which provides indices for splitting data to test and train parts. The model fits on train data [39]. Finally, this model is implemented on the test dataset, and the resulting model's parameters and hyperparameters are saved into a file [39].

After training the model using the training dataset and evaluating its performance on the test dataset, it is important to know that the test set gives an internal evaluation of the ability of the model for generalization [76]. Although the performance shows how the model works well with previously unseen data from the same dataset, it is possible that it does not guarantee robustness or applicability to completely independent data sources [76]. Another essential part of the model can help in generalizing and preventing overfitting to dataset specifics. An external validation set should be constructed from an entirely different but similar dataset, which does not overlap with the training and/or testing set of the model. In this study, the model is not implemented on any external validation set [76].

4.2.1 Grid Search

Grid search [39, 45] is a search method to find the best parameters for machine learning models. It runs through all the possible combinations of provided hyperparameters [39] . This method takes a list of hyperparameters and iteratively trains all the possible combinations [39] . The best hyperparameters that optimize the performance metrics are the output. Its working procedure is as follows:

1. In the first step it defines a hyperparameter space H with n hyperparameters.

$$H = H_1 \times H_2 \times \cdots \times H_n \quad (1)$$

H_i is the set of possible values for the *i-th* hyperparameter, and $\times$ identifies the cortesian product

2. Making a grid with all possible combinations of hyperparameters is the next step:
$$\left\{(h_1^1, h_2^1, \ldots, h_n^1), (h_1^2, h_2^2, \ldots, h_n^2), \ldots, (h_1^m, h_2^m, \ldots, h_n^m)\right\}$$

   where m is the total number of possible combinations, calculated as:
$$m = |H_1| \times |H_2| \times \cdots \times |H_n|$$

   being $|H_i|$ the cardinality of the set H_i

3. After that, the algorithm uses cross-validation to estimate the performance of the model:
$$\hat{E}(h_1^j, h_2^j, \ldots, h_n^j) = 1/K \sum_{k=1}^{K} E_k(h_1^j, h_2^j, \ldots, h_n^j)$$

   K is the chosen number of foldes in cross validation and E_k represents performance metric for *k-th* fold.

4. Finally, according to the aim of the model, maximizing or minimizing the given performance metric, it selects the combination of hyperparameters:
$$(h_1^*, h_2^*, \ldots, h_n^*) = \arg \min_{(h_1^j, h_2^j, \ldots, h_n^j)} \hat{E}(h_1^j, h_2^j, \ldots, h_n^j)$$

   or
$$(h_1^*, h_2^*, \ldots, h_n^*) = \arg \max_{(h_1^j, h_2^j, \ldots, h_n^j)} \hat{E}(h_1^j, h_2^j, \ldots, h_n^j)$$

4.2.2 Random Search

Another method to optimize hyperparameters is random search [39, 77]. In comparison to grid search, this method is less exhaustive and faster, since it randomly samples among the hyper-parameter combinations [39]. It defines a probability distribution for each hyper-parameter and then it samples from this distribution [39]. The procedure is the same as the grid search. Apart from the random sampling of hyper-parameter combinations from (1).

$$\left\{(h_1^1, h_2^1, \ldots, h_n^1), (h_1^2, h_2^2, \ldots, h_n^2), \ldots, (h_1^k, h_2^k, \ldots, h_n^k)\right\}$$

while h_i^j is sampled from distribution P_i.

4.3 Interpretability

In this study, 401 features from 266 cases are used for training models. Because of using different feature selection algorithms, it was decided to extract more valuable features and store them in separate files based on the feature selector algorithms, so that they can be used in the interpretability stage. In SelectKBest, PCA, and Select from Model algorithms, 35 features were selected, whereas, in Select Percentile algorithm, 7% of features were chosen, which was about 30 features.

Based on cooperative game theory, Shapley value is a concept in which each player who makes even a small contribution to the overall outcome produced by the team, will earn a part of it [53]. If the total payoff can be measured, Shapley values capture the marginal contribution of each player to the final result [78]. In the ML model, different features cooperate together to produce an output result, and Shapley values help to measure the contribution of each feature to the final result [78]. In simple terms, Shapley values are calculated by making carefully considered changes to the input features and observing how these modifications align with the eventual model prediction [78]. Then, the Shapley value of the feature is found as an average marginal contribution to the total model output [78].

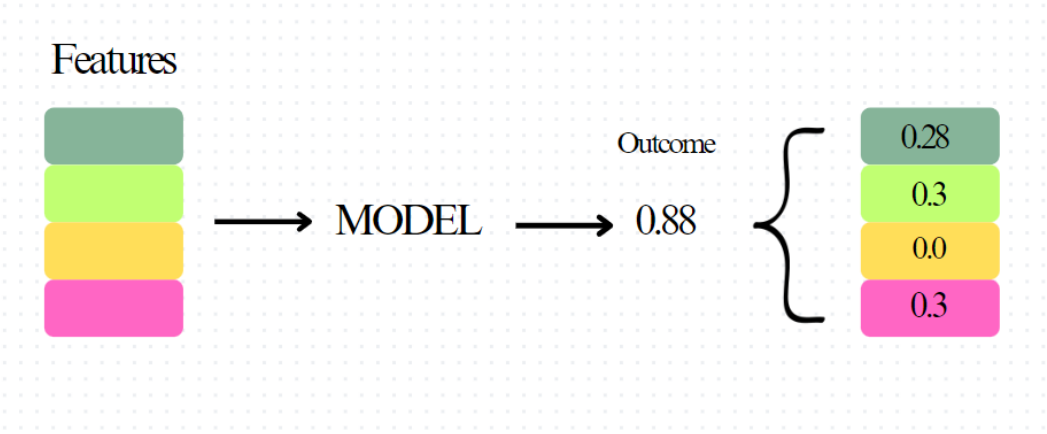

Figure 4: A simple diagram of how individual features can contribute to a model's final prediction score.

As an example of how Shapley values work, Figure 4 displays the contribution of each feature to the output result, and according to participation, scores each feature. Different methods are available to calculate the Shapley values. These methods can be used based on the classifier type. For example, for tree-based classifiers like Random Forest (RF), it is better to use TreeExplainer [79], while for Deep Learning (DL) models, DeepExpaliner [80] should be used. In this study, the Explainer [81] was used to compute the Shapley values. Explainer is a primary explainer interface of the SHAP library, and it accepts any combination of model

and masker that is a function to 'mask' out hidden features of the form [81]. This explainer returns a callable subclass object that implements the specific estimation algorithm that was chosen [82].

4.3.1 Shapley Values Calculation

The Shapley value for feature i in a model that has n features is calculated as follows [54]:

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S))$$

N: Set of all features (or players in a game)
S: Subset of N that does not include feature i (coalition of players)
$|S|$: Number of features in subset S
v: Characteristic function that maps subsets of players to real numbers
$v(S)$: Value of coalition S (describes the total expected sum of payoffs the members of S can obtain by cooperation)
$|N|$: Number of features (players) in N, |N| = n.

4.3.2 Expected Value

It refers to the model's average output value throughout the whole dataset. Expected value acts as a baseline, and provides a reference point from which SHAP values add or subtract to explain a particular prediction [54].

$$\mathbb{E}[f(X)] = \frac{1}{K}\sum_{i=1}^{K} f(x_i)$$

where K is the number of instances in the dataset, X is the dataset containing points x_i and $f(x_i)$ is the model prediction for an input x

4.4 Shapley Additive exPlanation (SHAP) plots

Shapley Additive exPlanation (SHAP) plots are utilized to increase understanding and analysis of predictions of ML models [81]. SHAP plots enhance model interpretability, identify key drivers with the help of summary and bar plots, and show the importance of features for specific predictions with force and waterfall plots [81].

1. **Summary plot**: This plot displays how Shapley values are distributed for each feature. It shows the level of feature importance and also how the model's output is impacted by the feature values. For each feature, summary plots show in a brief and useful approach, the size, direction, and prevalence of an effect [81]. SHAP summary plots do not combine the quantity and prevalence of an effect into a single value and therefore can represent infrequent high-magnitude impacts [83]. To find the most influential features in the model's prediction and also the effect of each feature on the output result, this plot is a suitable choice [83]. The y-axis contains the names of features, and the x-axis represents the Shapley value. As the density of points for a feature increases, shows the distribution of Shapley values per feature [83].

2. **Bar Plot**: It takes the mean absolute value of each feature across all instances of the dataset. The mean absolute value is not the only way to create a global measure of feature importance, it is possible to use any number of transformers [84, 81].

3. **Waterfall Plot**: The goal of this plot is to show the effect of SHAP values for every feature on model output and explain how the output value moves from the expectation results under the background data distribution [85]. The y-axis contains features and the value of the selected data instance. SHAP values are indicated on a scale by the x-axis. The SHAP value of each specific feature value represents a bar in the waterfall plot. In addition, the calculated expected prediction value E(f(x)) and the real prediction value for the instance $f(x_i)$ are shown on the x-axis [81]. The procedure is that bars start from the expected prediction value and move to reach the actual prediction.

4. **Heatmap Plot**: A heatmap plot is designed in a way to reveal the population substructure of a dataset through supervised clustering. This clustering is to cluster data points by their explanation, not by the original value of features [86].

5. **Decision Plot**: For numerous predictions, the decision-making process can be shown by looking at a decision plot. This plot indicates the cumulative effect of all features on the outcome of the model. In this plot, if one predicts, the value of the features will be printed, otherwise, it is not possible to print the values [53].

6. **Beeswarm Plot**: A beeswarm plot can be useful in providing an overview of the relationship between a model's output and the key features of a dataset. Every feature has a dot for each occurrence where an explanation is given [53].

Chapter 5

Results of Implementing Machine Learning Model on Different Datasets

5.1 Output Results of Different Combinations of Classification, Feature Selection, and Scaling Strategies

Various outlier detection and search algorithms have been applied to the data. The results obtained from each combination are given below.

As mentioned in the data cleaning section in Chapter 3, two methods were used for detecting outliers: Z-Score and IQR. Three types of data are available: the first one contains 401 extracted features from the rest protocol where the number of abnormal cases was 203 and normal cases was 63, the second one contains 404 features from the stress protocol, and the last dataset includes 401 features from both rest and stress protocols.

5.1.1 Results on Rest Data, Optimization through Random Search, Outliers Detected by IQR

After the preprocessing stage, the designed model tried different combinations of feature selection, classification, and scaling methods on both the training and test parts of the data. On the first try, the model was implemented on Rest data. At first, the Rest data was examined, and the 'IQR' method was used to identify outliers. Then, the search method sets on the 'random search' to optimize parameters.

ACC, SPE, SEN, and AUC values are measured on both the training and test parts of the dataset, but results are given only for the test dataset. ACC Figure 5, AUC values Figure 6, SPE values are visible in Figure 7, and the results of SEN are illustrated in Figure 8 for selected classifiers with each of three scalers and each of the four feature selectors, optimized with random search on Rest dataset. The number of selected features is 35 for 'SelectKBest', 'PCA', and 'Select from Model'. The number of selected features for 'SelectPercentile' is 30, which is 7% of all features.

This study focuses on choosing the classifiers with the highest value of SPE, SEN, ACC, and AUC metrics. Table 1, represents the classifiers with the highest values of the mentioned metrics. Classifiers with LR, RF, AdaBoost, and K-NN classification algorithms have better performances.

Scaling	**Classification**	**Feature Selector**	**ACC**	**SPE**	**SEN**	**AUC**
StandardScaler	LR	PCA	0.74	0.75	0.7	0.76
MinMax	LR	PCA	0.76	0.69	0.8	0.75
RobustScaler	K-NN	Select k Best	0.7	0.75	0.76	0.75
StandardScaler	AdaBoost	Select from Model	0.77	0.88	0.54	0.75
MinMax	RF	Select k Best	0.7	0.69	0.78	0.72

Table 1: Best classifiers of implementing random search on Rest data, outliers were detected with IQR, and the value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUR: Area Under the Curve

As models are applied to different values of parameters and hyperparameters, the output of each one is different. The output results are stored in a file to recognize the combinations with the highest performance. As mentioned, accuracy is not always reliable, particularly in the presence of imbalanced data, where used data in this study is imbalanced. Therefore, additional metrics are considered. The AUC is especially important and varies across different combinations of parameters and hyperparameters. Youden index for the test part was calculated based on TPR and FPR. Then, all calculated values for the Youden index are sorted incrementally. Then, the optimal threshold is calculated for ROC-AUC curve of each combination.

Plots of ROC-AUC for some combinations with the highest metric values are displayed based on the name of the classification algorithm. Figure 9 shows the ROC curves and the value of AUC for each combination on the test part of the data. Figure 10 displays the ROC curves and AUC value for each combination on the training data.

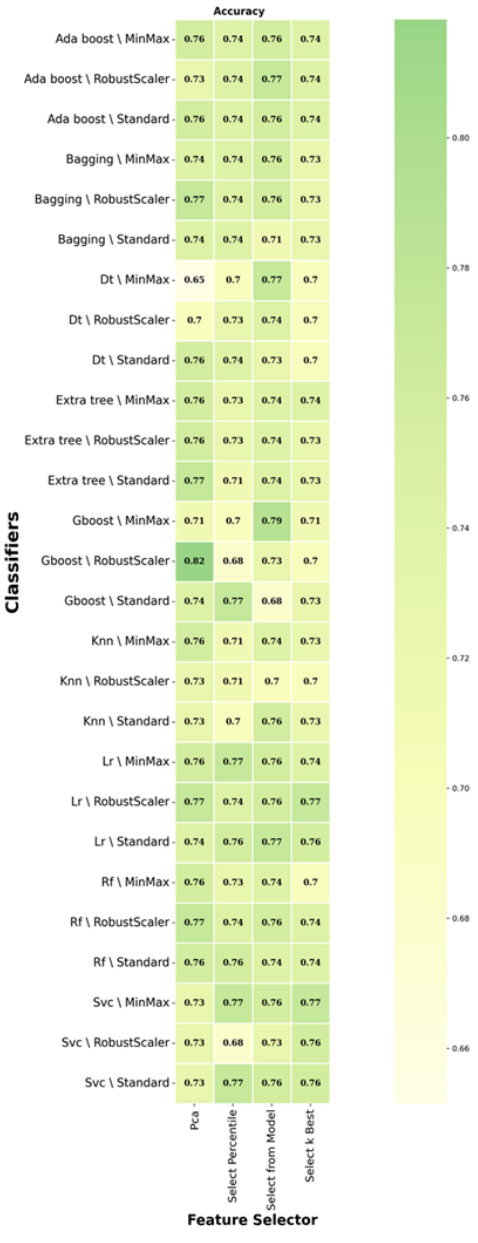

Figure 5: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

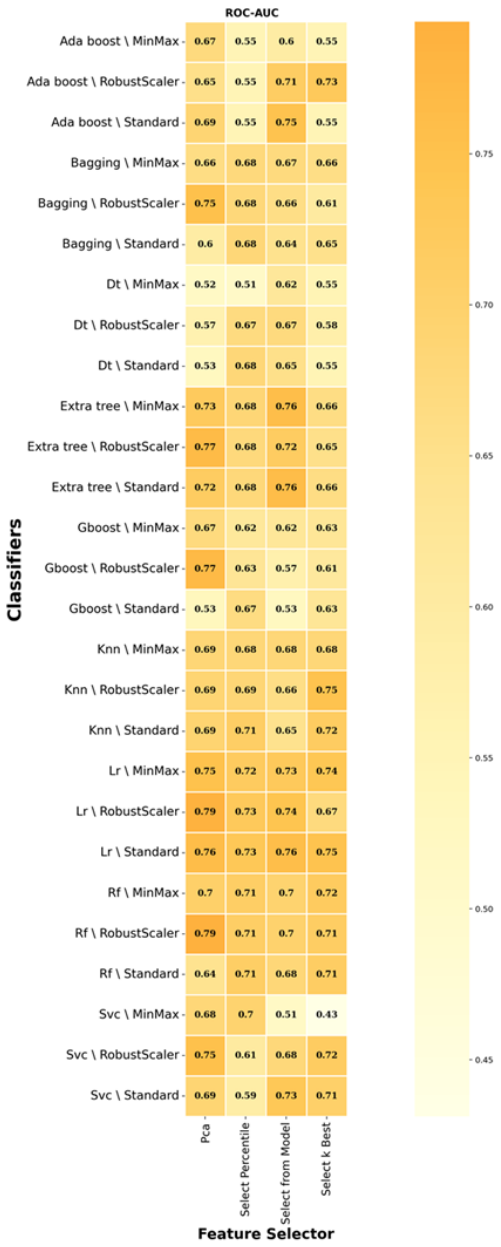

Figure 6: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

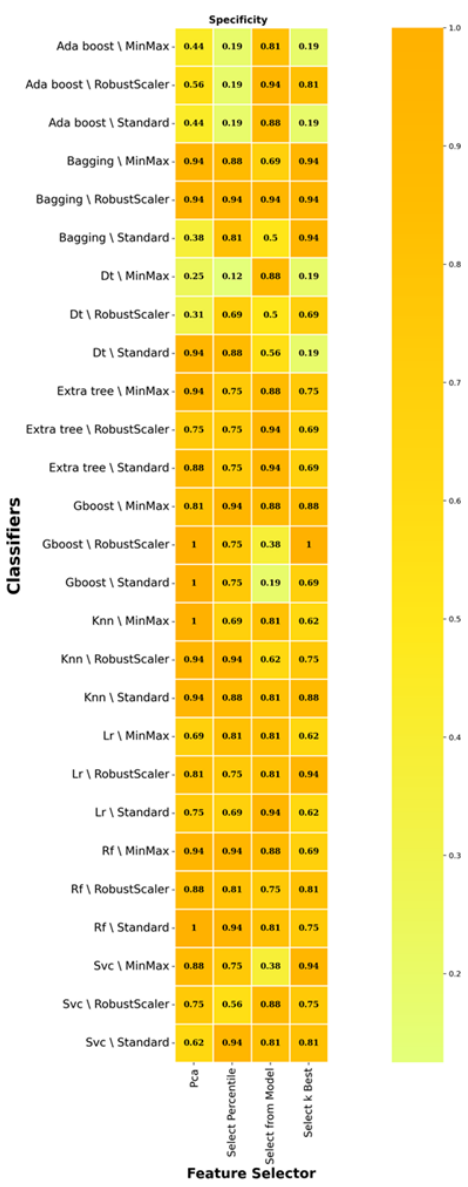

Figure 7: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

Figure 8: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows) and 4 feature selectors (columns), optimized with the random search on Rest data

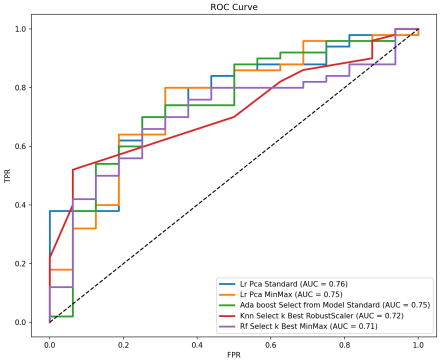

Figure 9: ROC curves of random search for the test data of Rest with the highest AUC values

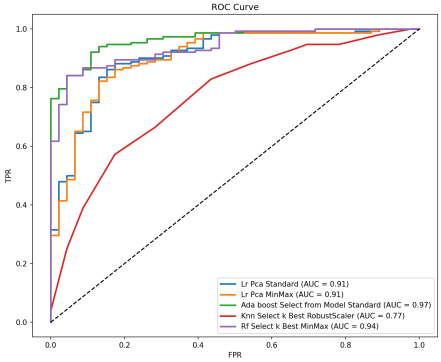

Figure 10: ROC curves of random search for the train data of Rest with the highest AUC values

5.1.2 Results on Rest Data, Optimization through Grid Search, Outliers Detected by IQR

The other setting applied to Rest Data is changing the optimization method. 'Grid search' was replaced with random search.

Grid search, also known as the exhaustive search method, assesses all potential combinations of a defined set of parameters by iteratively exploring various combinations and cross-validating to find the values that provide the best performance on the dataset. The advantage of grid search is that it is exhaustive and finds the best combination. By implementing this model on data, different metrics have been considered. ACC of selected classifiers with each of the three scalers and four feature selectors, optimized with 'grid search' on the 'Rest' dataset is shown in Figure 11. Figure 12 illustrates all the obtained values of AUC for different combinations of scaling, feature selector, and classification algorithms. Two other metrics are SPE and SEN which all values are displayed in Figure 13, and Figure 14.

The aim of finding these metrics is to recognize models with the highest performance and in the next step, to use them to determine the interpretability of features. Table 2 lists of the models with the best results.

Scaling Method	Classification	Feature Selector	ACC	SPE	SEN	AUC
RobustScaler	RF	PCA	0.77	0.87	0.64	0.8
StandardScaler	LR	PCA	0.74	0.75	0.7	0.76
MinMax	LR	PCA	0.76	0.69	0.8	0.75
StandarScaler	SVC	PCA	0.76	0.81	0.74	0.73
MinMax	K-NN	SelectPercentile	0.73	0.69	0.72	0.72

Table 2: Best classifiers of implementing grid search on Rest data, outliers were detected with IQR and value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUC: Area Under the Curve

According to Table 2, models with RF, LR, SVC, and K-NN had better output results, therefore, the ROC plots of the models are illustrated in Figure 15 for test data, and Figure 16 for training part of data.

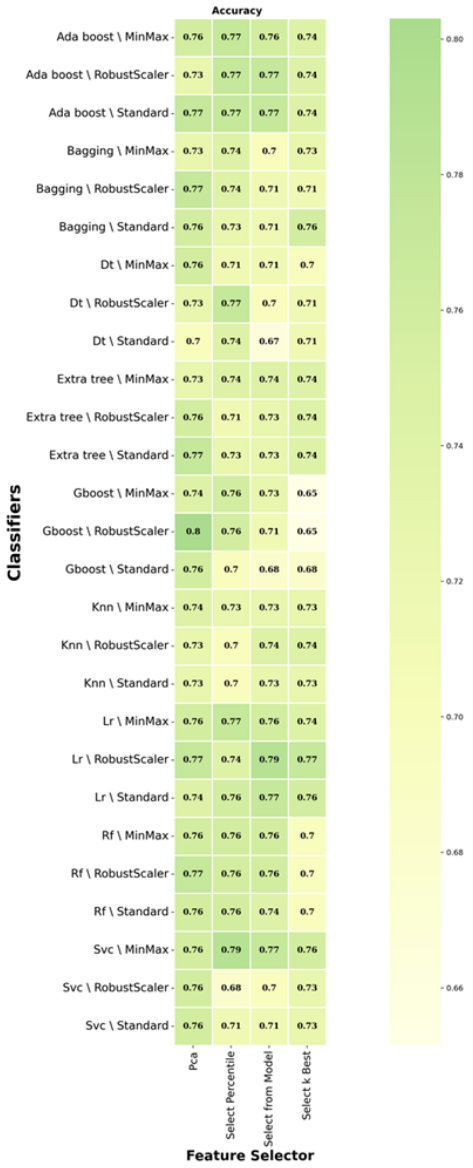

Figure 11: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with grid search on Rest data

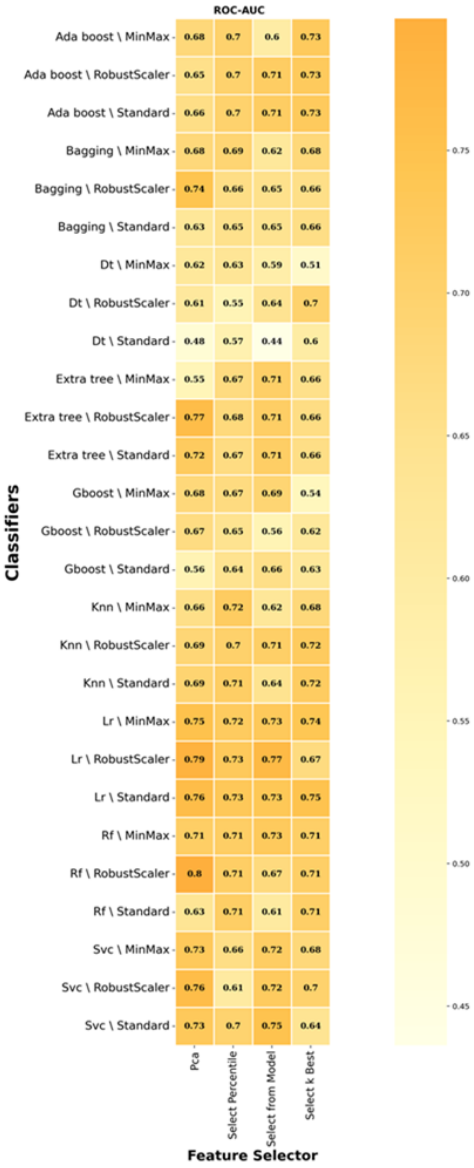

Figure 12: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with grid search on Rest data

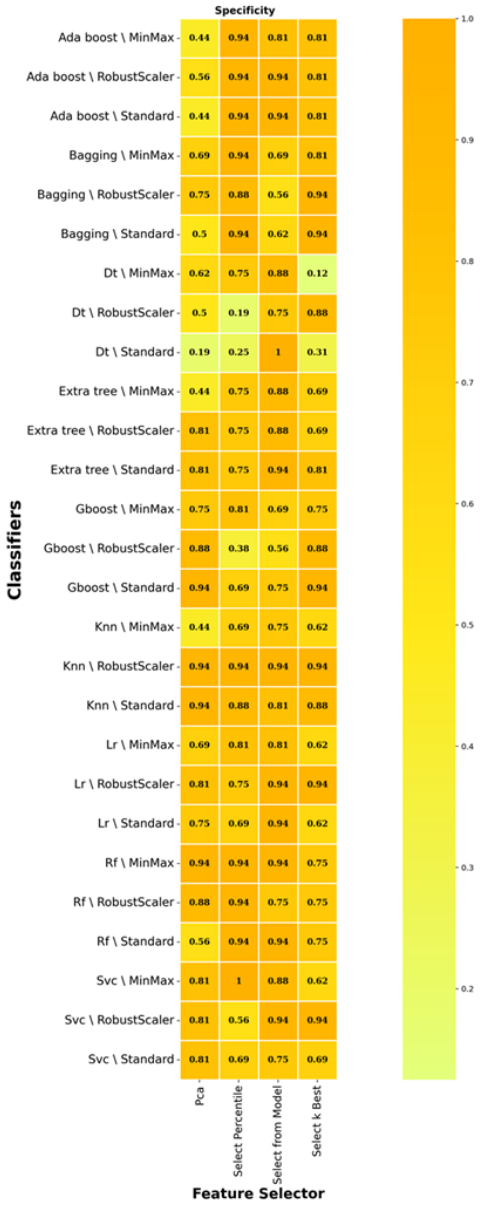

Figure 13: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with grid search on Rest data

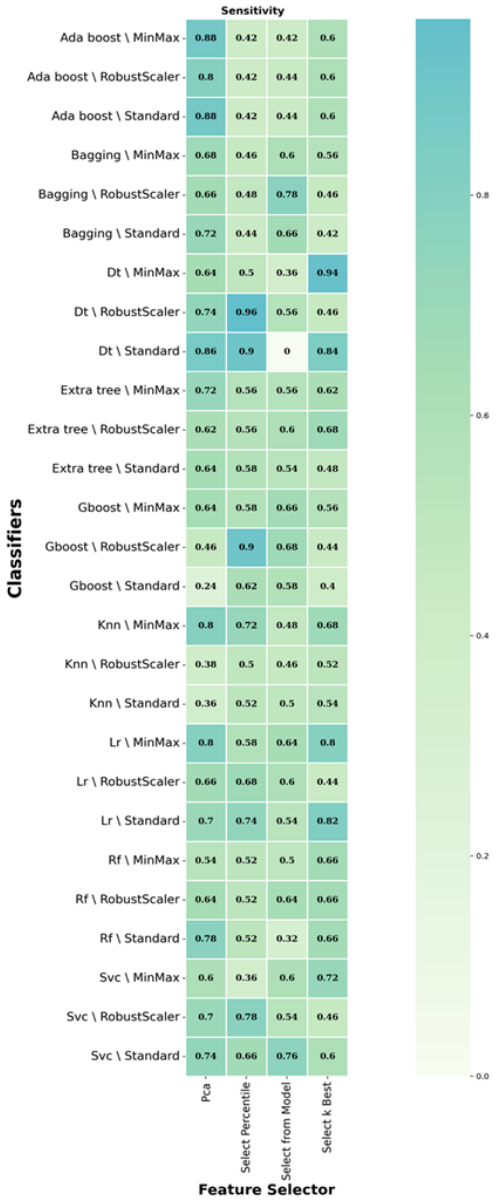

Figure 14: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with grid search on Rest data

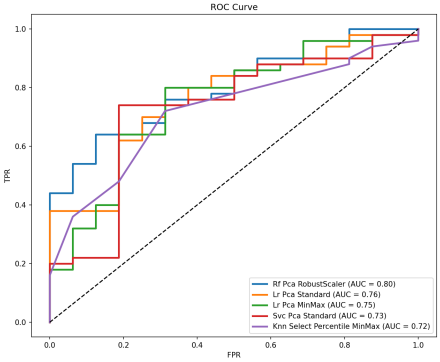

Figure 15: ROC curves of grid search on the test data of Rest with the highest AUC values

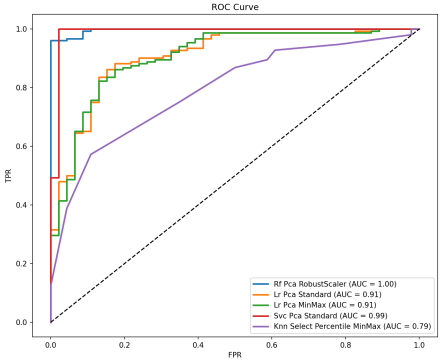

Figure 16: ROC curves of grid search on the train data of Rest with the highest AUC values

5.1.3 Results on Stress Data, Optimization through Random Search, Outliers Detected by IQR

The second data type in this study consists of features extracted from the stress protocol. The outliers are detected with the 'IQR' method and to optimize hyper-parameters and parameters, the 'random search' is used. ACC of selected classifiers with each of the three scalers and four feature selectors, optimized with the random search on the Stress dataset shown in Figure 17. The output results for AUC illustrated in the figure Figure 18. SPE and SEN are two other metrics that are evaluated and can be seen in Figure 19 and Figure 20.

In some combinations, it is clear that results for some metrics are 1 in both train and test parts, and it shows that the model may have over-fitted or under-fitted. Therefore, these models are not reliable and should not be examined. Models with the best results can be seen in Table 3.

Scaling Method	Classification	Feature Selector	ACC	SPE	SEN	AUC
MinMax	K-NN	Select from Model	0.87	0.94	0.98	0.94
StandardScaler	K-NN	Select K Best	0.87	0.94	0.84	0.92
MinMax	K-NN	PCA	0.84	0.5	0.9	0.73
MinMax	DT	PCA	0.73	0.56	0.82	0.75

Table 3: Best classifiers of implementing random search on Stress data, outliers detected with IQR and value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUC: Area Under the Curve

Models with K-NN classification had better output values. ROC figures of different combinations are displayed in Figure 21 and Figure 22.

Figure 17: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

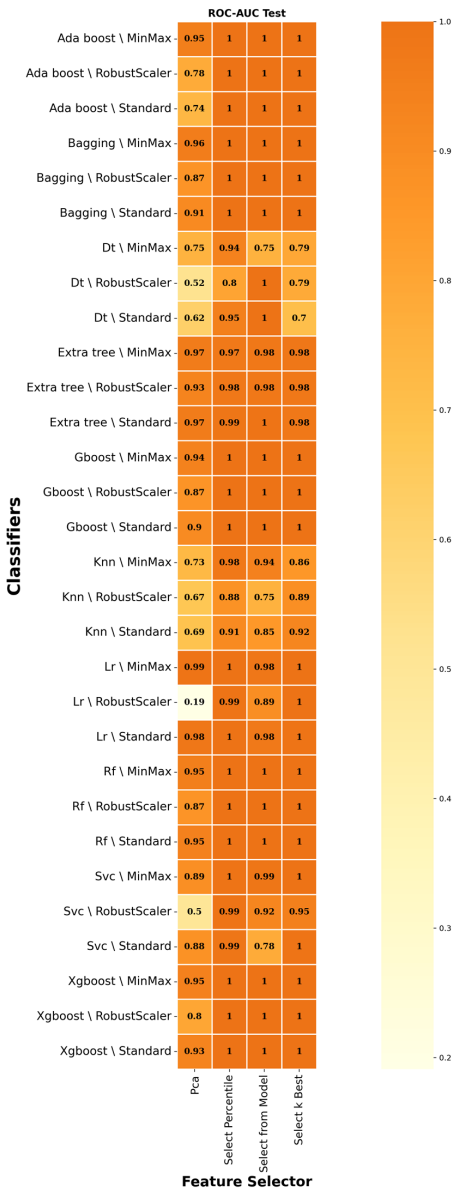

Figure 18: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

Figure 19: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

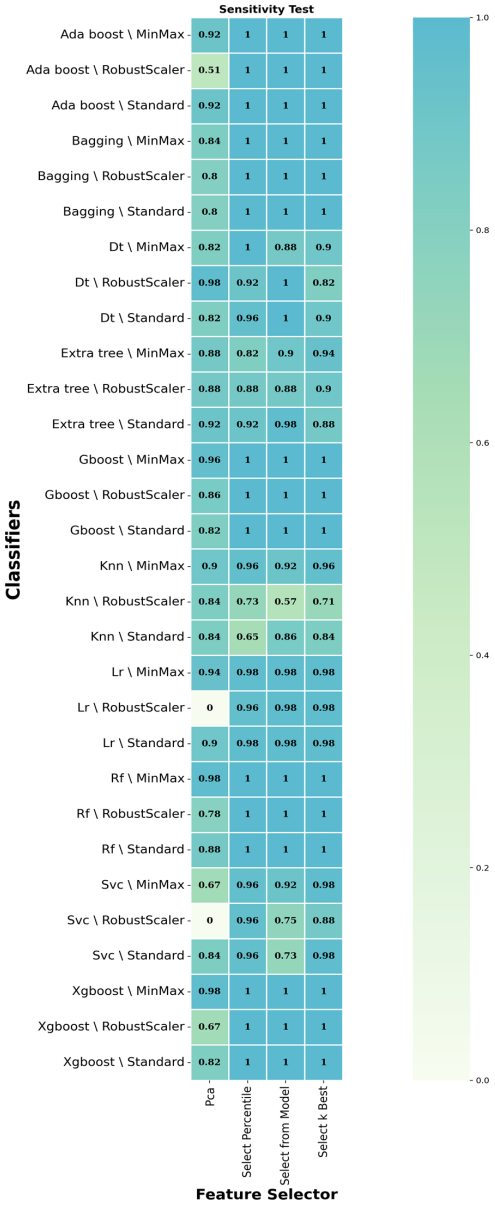

Figure 20: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

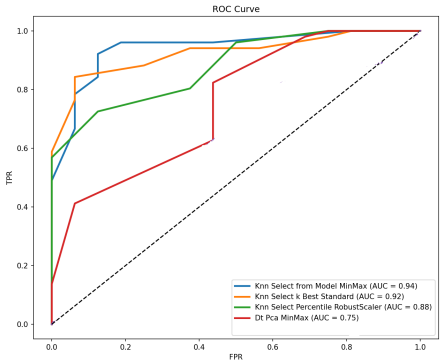

Figure 21: ROC curves of Random search on the test data of Stress with the highest AUC values

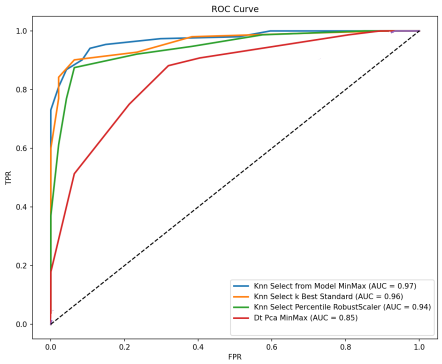

Figure 22: ROC curves of Random search on the train data of Stress with the highest AUC values

5.1.4 Results on Stress Data, Optimization through Random Search, Outliers Detected by Z-Score

Another method that has been tested on obtained data of stress protocol was 'Z-Score' to identify the outliers. To optimize the parameters and hyperparameters, the random search algorithm was used. According to the results obtained for the metrics, it can be concluded that the implementation of the 'Z-Score' has improved the results of the desired metrics in this part of data. All outputs of ACC, AUC, SPE, and SEN are shown as heat maps in figures Figure 23, Figure 24, Figure 25, and Figure 26. The values clearly show that the K-NN, SVC, and LR algorithms produce better results.

The best combinations of classification, feature selector, and scaling strategies are visible in Table 4.

Scaling Method	Classification	Feature Selector	ACC	SPE	SEN	AUC
StandardScaler	SVC	Select k Best	0.83	0.92	0.95	0.95
StandardScaler	LR	Select from Model	0.83	0.92	0.88	0.95
MinMax	K-NN	Select K Best	0.85	0.92	0.76	0.90
MinMax	K-NN	Select from Model	0.81	0.85	0.88	0.89
StandardScaler	K-NN	Select from Model	0.78	0.85	0.68	0.78

Table 4: Best classifiers of implementing random search on Stress data, outliers detected with Z-Score and value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUC: Area Under the Curve

The AUC plot of test and training data for combinations of classification, feature selection, and scaling method with the highest value of AUC are displayed in Figure 27, and Figure 28

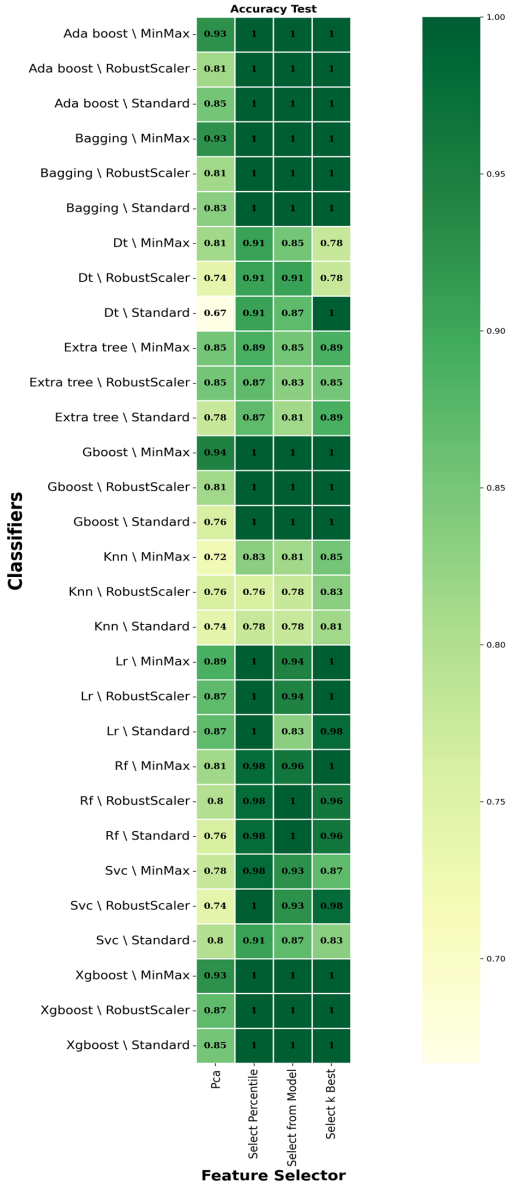

Figure 23: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

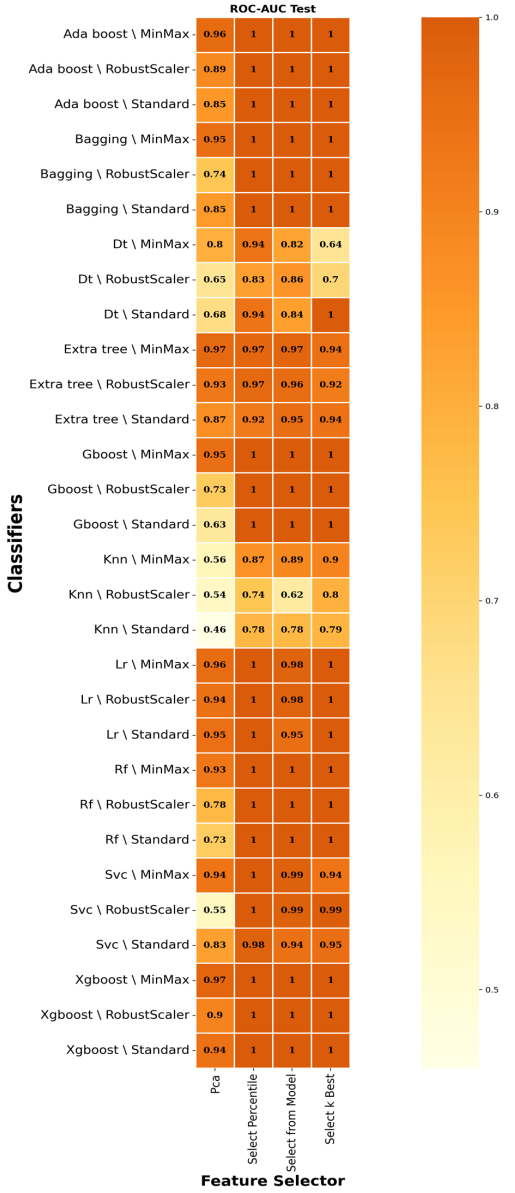

Figure 24: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

Figure 25: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

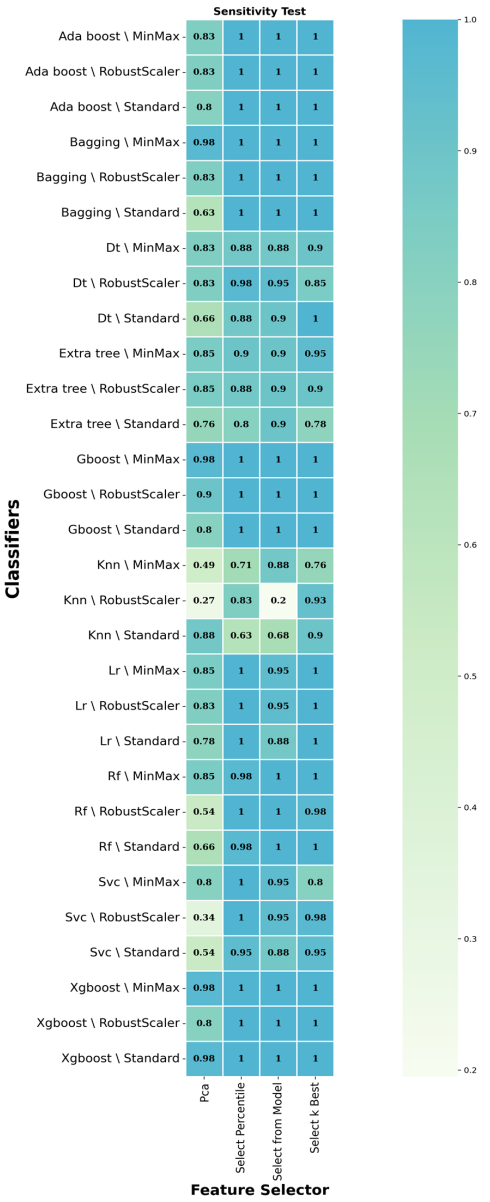

Figure 26: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Stress data

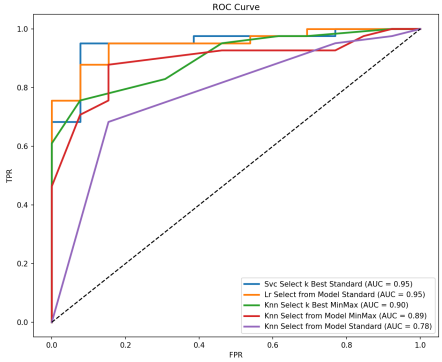

Figure 27: ROC curves of Random search on test data of Stress with the highest AUC values

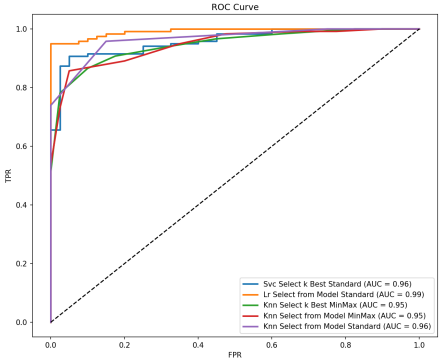

Figure 28: ROC curves of Random search on train data of Stress with the highest AUC values

5.1.5 Results on Rest Data, Optimization through Random Search, Outliers Detected by Z-Score

Another tested combination used the 'Z-score' to identify outliers in the data obtained from the Rest protocol. To optimize parameters and hyper-parameters, a 'random search' was used. 'Z-score' uses a threshold to identify outliers, and the common value for that is 2, 2.5, and 3, which in this study has been set to 3. As the Z-score has a limited threshold, a number of cases are eliminated. The number of patients reduces from 266 to 229. All combinations based on three scaling methods, four feature selector algorithms, and nine classification algorithms are applied to training and test data. The result of classifiers with the highest metric values is in table Table 5.

Scaling Method	Classification	Feature Selector	ACC	SPE	SEN	AUC
MinMax	RF	Select from Model	0.81	0.69	0.89	0.85
MinMax	LR	PCA	0.79	0.77	0.82	0.84
StandardScaler	RF	Select from Model	0.78	0.85	0.76	0.79
StandardScaler	K-NN	Select from Model	0.76	0.69	0.8	0.73
RobustScaler	LR	SelectPercentile	0.79	0.69	0.78	0.74

Table 5: Best classifiers of implementing random search on Rest data, outliers detected with Z-Score and value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUC: Area Under the Curve

The results of all combinations for different metrics are plotted in heatmaps. The ACC results are visible in Figure 29. The results of AUC, SPE, and SEN are illustrated in Figure 30, Figure 31, and Figure 32.

Based on results in Table 5, the classifiers with LR, K-NN, and RF classification algorithms perform better. Therefore, the ROC curves of some combinations with the highest value of AUC are illustrated in Figure 33, and Figure 34.

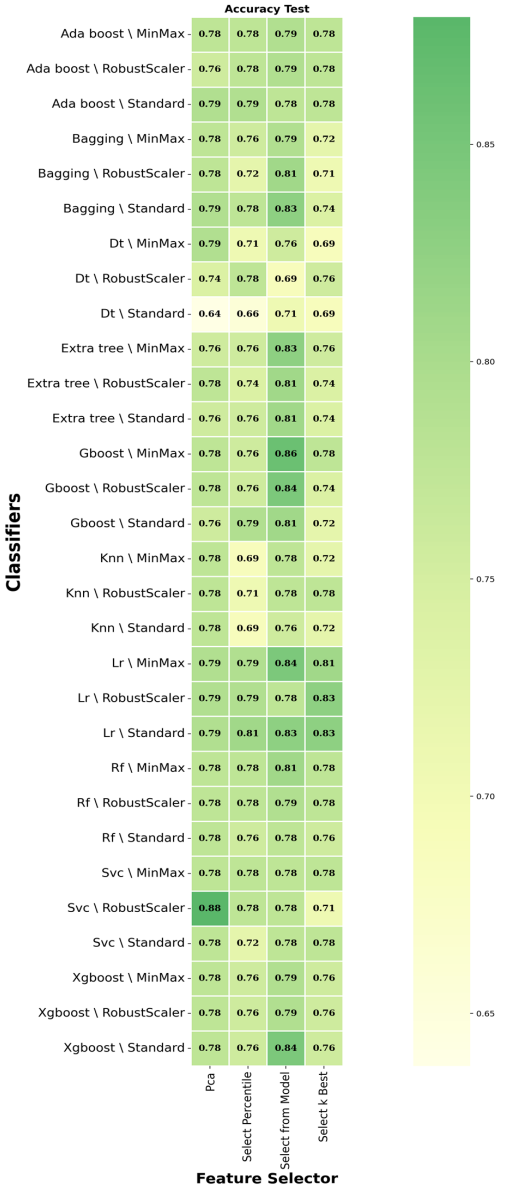

Figure 29: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

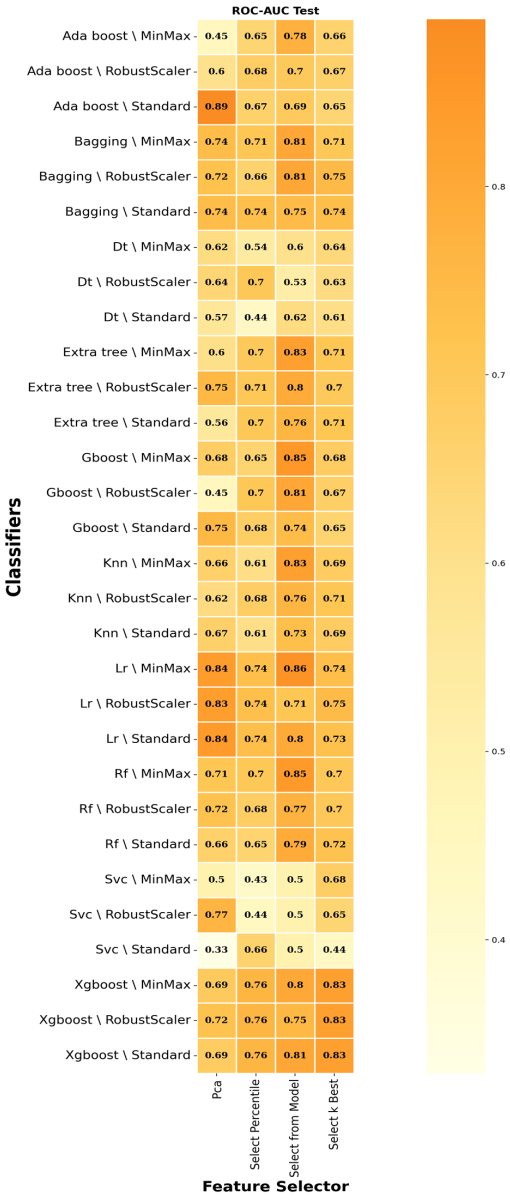

Figure 30: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

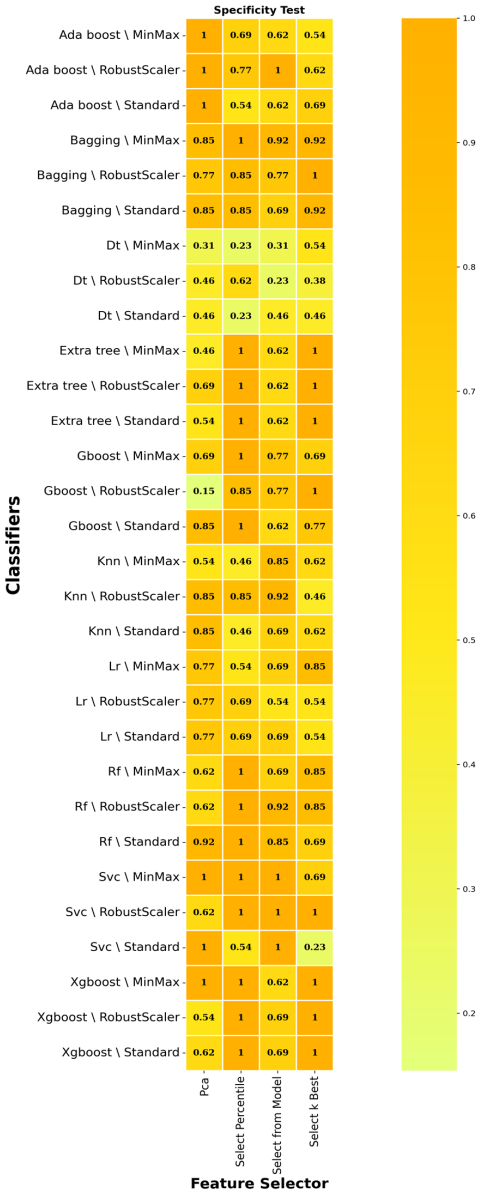

Figure 31: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

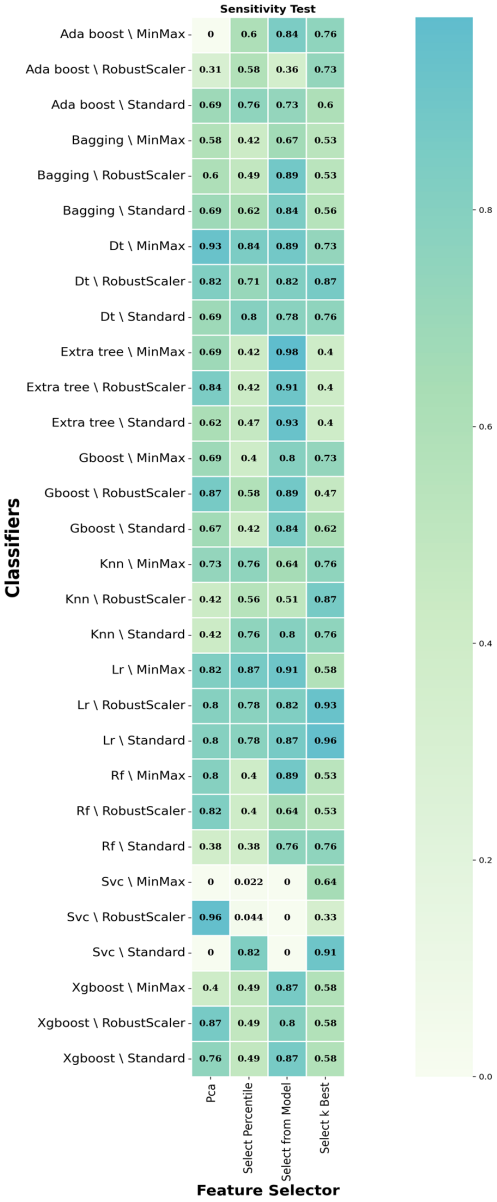

Figure 32: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest data

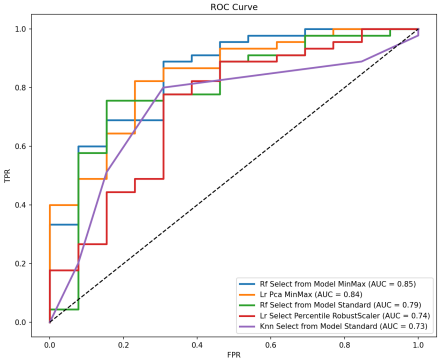

Figure 33: ROC curves of Random search on test data of Rest with the highest AUC values

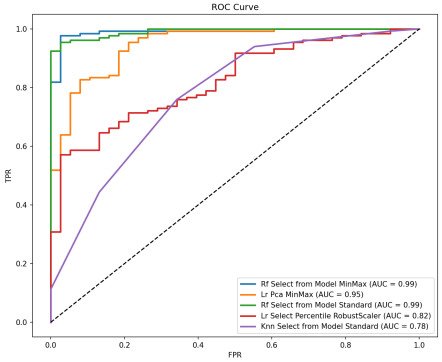

Figure 34: ROC curves of Random search on train data of Rest with the highest AUC values

5.1.6 Results on Combined Data, optimization through Random Search, Outliers Detected by Z-Score

In this stage, Rest and Stress features are combined, and a combined file is created. To detect outliers, the 'Z-Score' method is implemented on the data, and for parameter and hyper-parameter optimization, the 'random search' was used.

The output results are shown in different figures. The first one, Figure 35 displays all the output values for the ACC metric. The next measured metric is AUC, all output results illustrated in Figure 36. Two other important metrics are SPE and SEN. The heatmap plot of both are displayed in Figure 37 and Figure 38.

Models with better results have been displayed in Table 6. Models with LR and SVC classification algorithms perform better on data.

Scaling Method	Classification	Feature Selector	ACC	SPE	SEN	AUC
RobustScaler	LR	Select from Model	0.83	0.81	0.81	0.86
RobustScaler	LR	PCA	0.78	0.81	0.77	0.84
RobustScaler	SVC	Select from Model	0.81	0.85	0.77	0.86
StandardScaler	SVC	Select from Model	0.78	0.81	0.83	0.86
StandardScaler	LR	Select from Model	0.81	0.81	0.81	0.85

Table 6: Best classifiers of implementing random search on Rest-Stress data, Outliers detected with Z-Score and value of SEN: Sensitivity, SPE: Specificity, ACC: Accuracy, AUC: Area Under the Curve

ROC plots of some combinations of scaling methods, feature selection strategies, and classification algorithms with the highest value of AUC have been illustrated in Figure 39 and Figure 40.

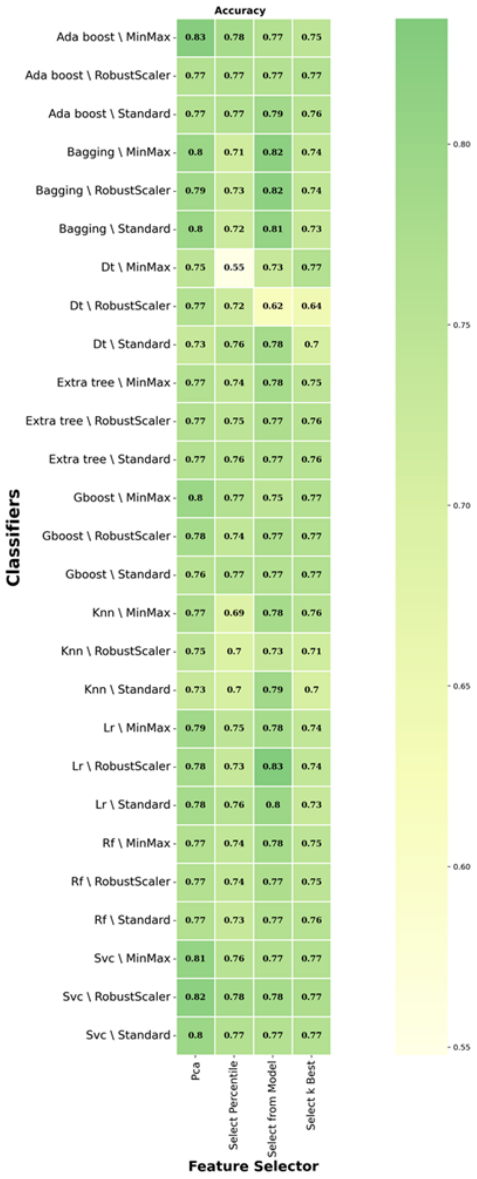

Figure 35: ACC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest-Stress data

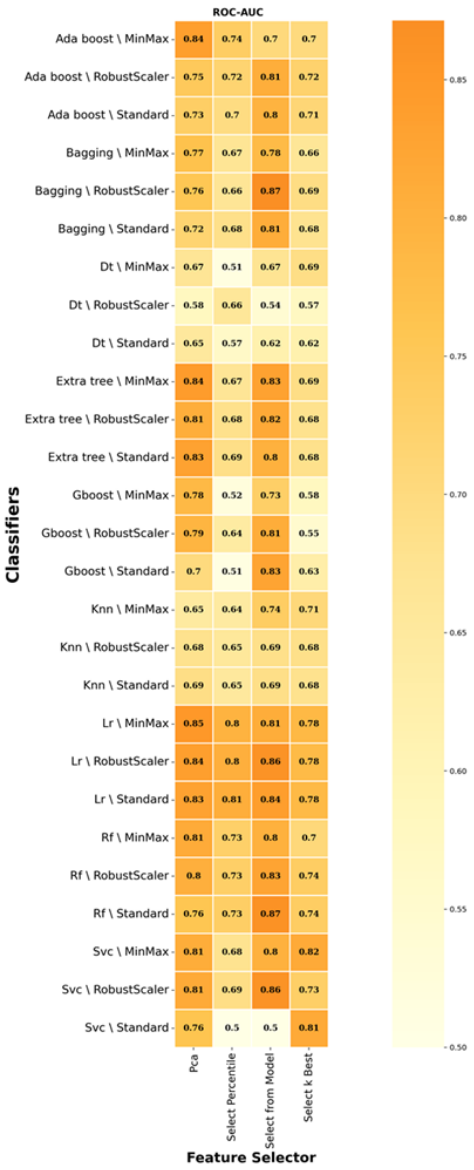

Figure 36: AUC of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest-Stress data

Figure 37: SPE of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest-Stress data

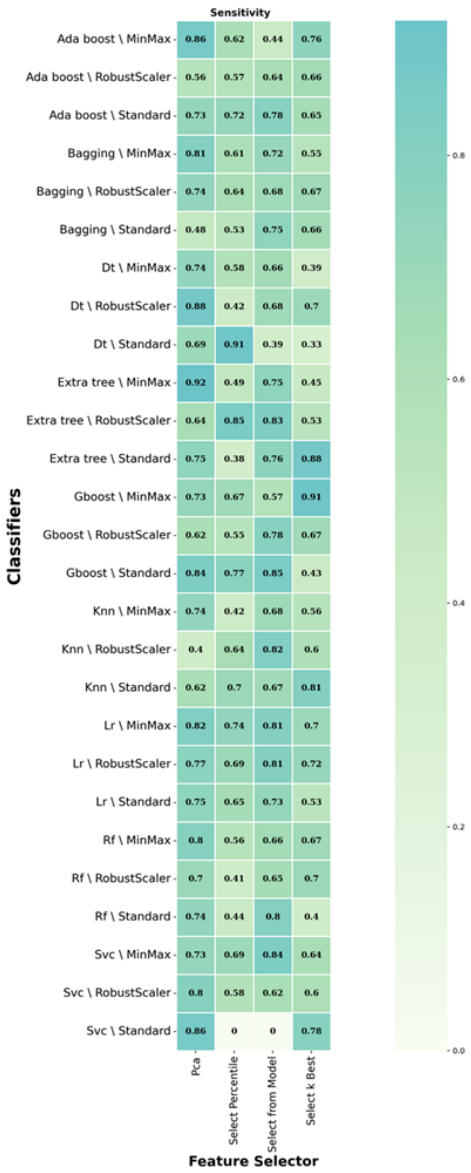

Figure 38: SEN of the selected classifiers with each of the 3 scalers, 9 classification algorithms (rows), and 4 feature selectors (columns), optimized with the random search on Rest-Stress data

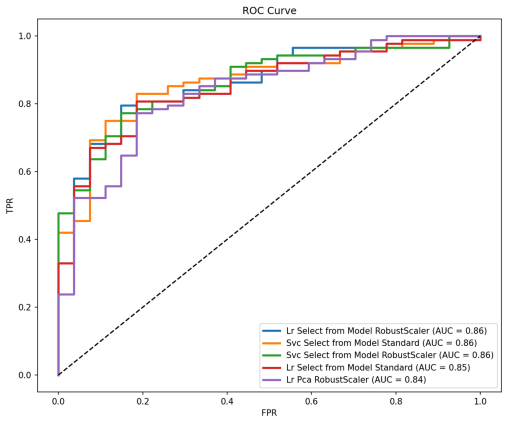

Figure 39: ROC curves of Random search on test data of Rest-Stress with the highest AUC values

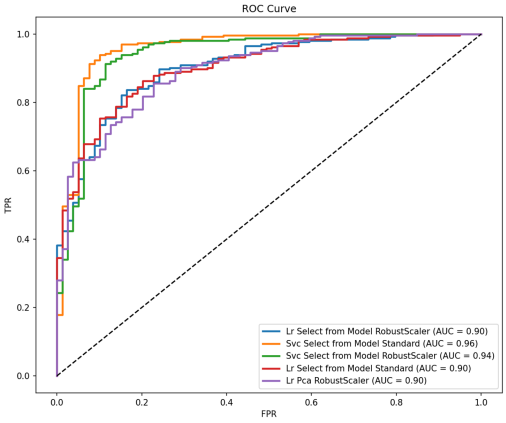

Figure 40: ROC curves of Random search on train data of Rest-Stress with the highest AUC values

5.2 Results Comparison

Here we take a closer look at the results obtained from different models. Among the results obtained on Rest data, where outliers were detected with the 'IQR' method, and parameter and hyper-parameter optimization approach were random search, the classifier with Logistic Regression (LR) classification algorithm, Principal Component Analysis (PCA) as feature selector, and StandardScaler as scaling method, had better results. Accuracy of 0.74, specificity of 0.75, sensitivity of 0.70, and AUC value of 0.76 were the metric values. By implementing grid search as an optimization method on the Rest data, which outlier detection was 'IQR', the best classifier recognized with the Logistic Regression (LR) classification algorithm, PCA feature selector, and StandardScaler as the scaling method. The obtained result for metrics is an accuracy of 0.74, specificity of 0.75, sensitivity of 0.7, and AUC of 0.76. Other data used in this study were extracted features from the Stress protocol. The outliers were detected with 'IQR', and parameters and hyper-parameters were optimized through random search. The best classifier had K-Nearest Neighbor (K-NN) as the classification algorithm, Select from Model feature selector, and MinMax as a scaling method. The results of the metrics are: accuracy of 0.87, specificity of 0.94, sensitivity of 0.98, and AUC of 0.94. Another method to detect outliers which are used in this study was 'Z-score'. Parameters and hyper-parameters were optimized with the random search on Stress data. The best classifier used 'Support Vector Classification (SVC)' as a classification algorithm, 'Select K Best' as a feature selector, and 'StandardScaler' as a scaling method. The best metric results are an accuracy of 0.83, specificity of 0.92, sensitivity of 0.95, and AUC of 0.95. 'Z-score' was implemented on the Rest data to detect outliers, and the optimization method was a random search. The best classifier had the combination of 'LR' as the classification algorithm, 'PCA' as the feature selector, and 'MinMax' as the scaling strategy. The results of metrics are an accuracy of 0.79, specificity of 0.77, sensitivity of 0.82, and AUC of 0.84. By implementing 'Z-score' as an outlier detector on combined data of Rest and Stress, and optimizing the parameters and hyper-parameters by random search, the best classification was Logistic Regression (LR), and the feature selector was 'Select from Model', and the scaling method was 'RobustScaler'. The best values of metrics are an accuracy of 0.83, specificity of 0.81, sensitivity of 0.81, and AUC of 0.86. Finally, the classifiers on combined data are recognized as more acceptable than others, because the greater number of combinations of classification algorithms, feature selectors, and scaling methods had better performance in comparison to other data and classifiers.

Chapter 6

Results of Implementing Interpretability Method on Selected Models

6.1 SHAP plots of the Selected Models for The Combined Dataset

In this section, some analytical plots of the outputs of different combinations are displayed. According to Table 6, the best results are for combined Rest-Stress data with 'Logistic Regression (LR)' classification algorithm and 'Select from Model' as feature selector, and 'RobustScaler' to scale features: the value of metrics is ACC = 0.83, AUC= 0.86, SPE=0.81, SEN = 0.81. As this model was applied to training and test data, selected features with the highest importance and the best parameters or hyper-parameters were stored in separate files, to be used in the interpretability part.

According to Figure 41, Figure 42, Figure 43, Figure 44, and Figure 45, the feature 'Zone distance variance.1' [58] has the highest Shapley value and effect on outcome. In the bar plot Figure 41, on the x-axis, the mean SHAP value represents the average influence of a feature on model predictions. In the y-axis, the features are sorted based on importance. The x-axis in the summary plot Figure 42 represents the Shapley value and the impact of a feature on the model's output. The color shows feature values ranging from low (blue) to high (red). Every dot in the summary plot indicates a single instance in the dataset. On the waterfall plot Figure 43, the starting point is the base value, obtained from calculating average prediction values across the dataset. The value on each feature step represents the Shapley value of that feature's contribution to the final prediction. In the

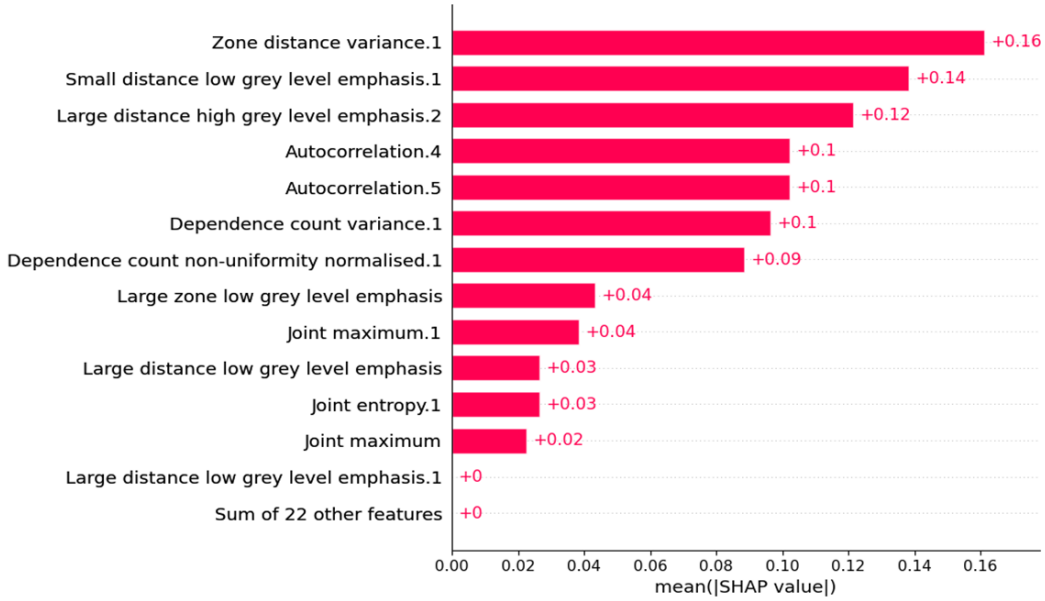

Figure 41: Bar plot of Shapley values of implementing designed interpretability model on Rest-Stress selected features

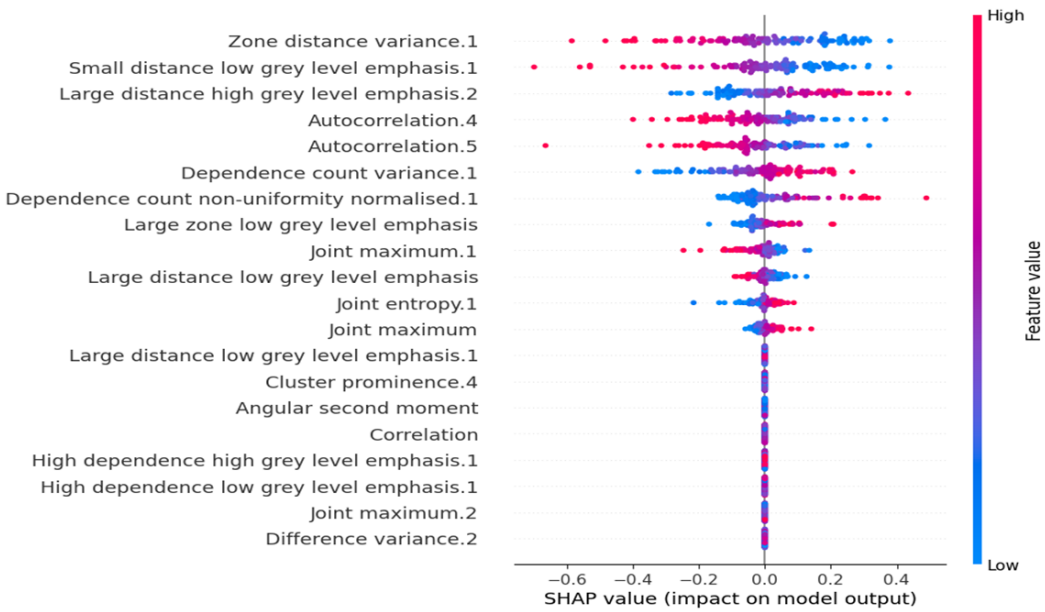

Figure 42: Summary plot of Shapley values of implementing designed interpretability model on Rest-Stress selected features

decision plot Figure 44, along the y-axis, features are sorted by their contribution to the final prediction. The X-axis consists of the cumulative Shapley value, and it

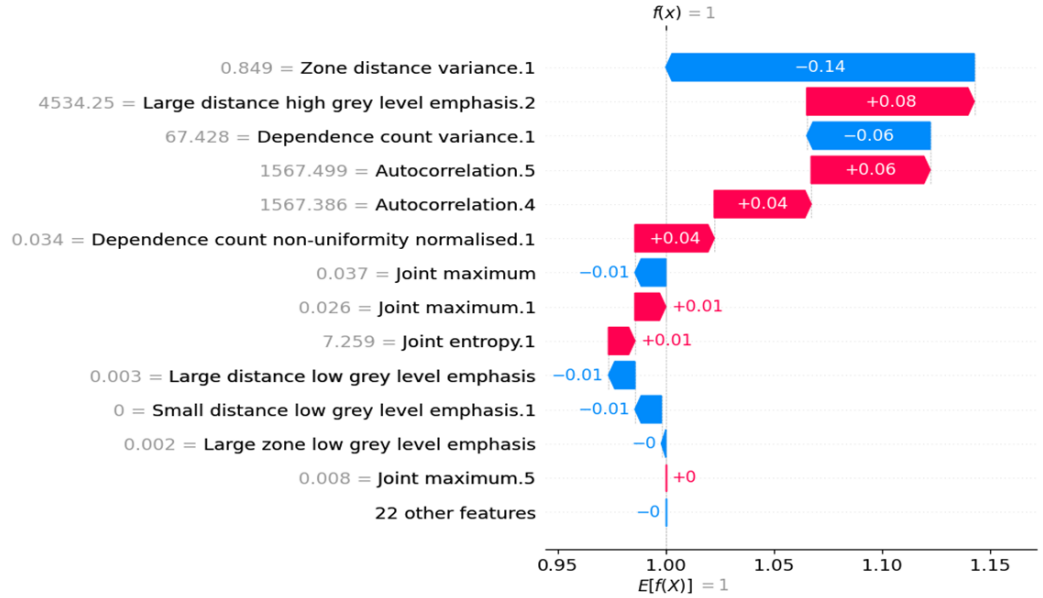

Figure 43: Waterfall plot of Shapley values of implementing designed interpretability model on Rest-Stress selected features

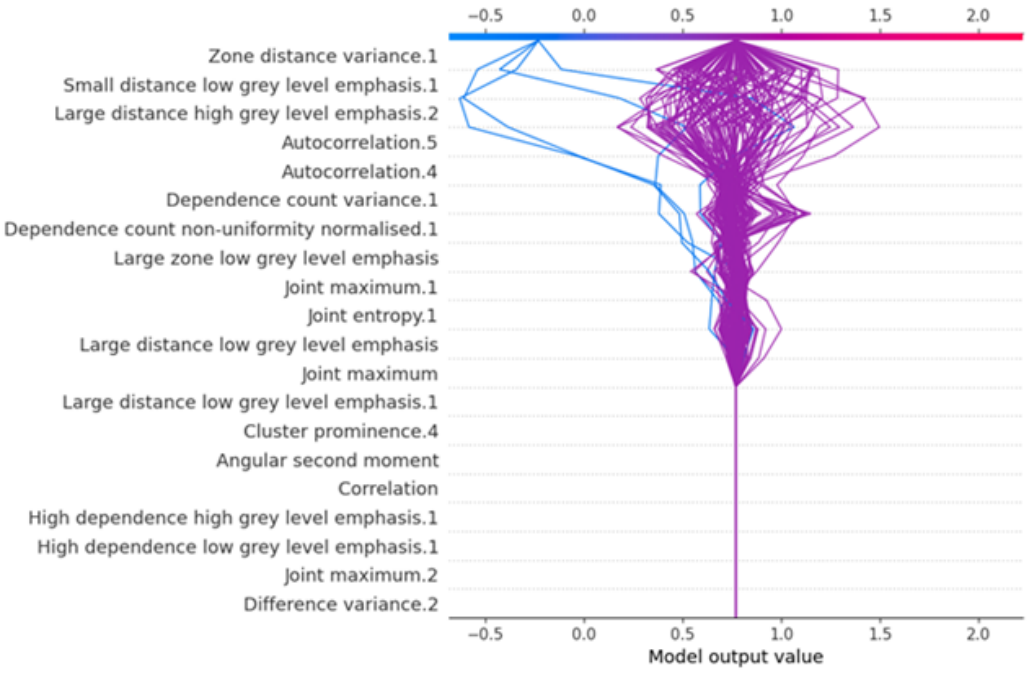

Figure 44: Decision plot of Shapley values of implementing designed interpretability model on Rest-Stress selected features

shows how every single feature adds up to the total prediction value of the model, based on each instance. Every single line indicates an instance of the dataset. The movement of the line horizontally from up to down shows the impact of a feature on the prediction. The middle line in this plot is the expected value of the model output and acts as a reference point. It is the average prediction value of the model without considering feature effects. The heatmap plot Figure 45 is used to display Shapley values of features for several instances. In addition, it shows the interaction and cumulative effect of feature contributions throughout the dataset. The x-axis shows samples in the dataset. The y-axis is to show features from highest value to lowest value. To show the bigness of Shapley values, color density indicates the contribution of each feature for each instance of the dataset. For example, instances of the 'Zone distance variance.1' [58] feature, between 65 to 75 have the highest positive impact while instances between 100 to 105 have the highest negative value.

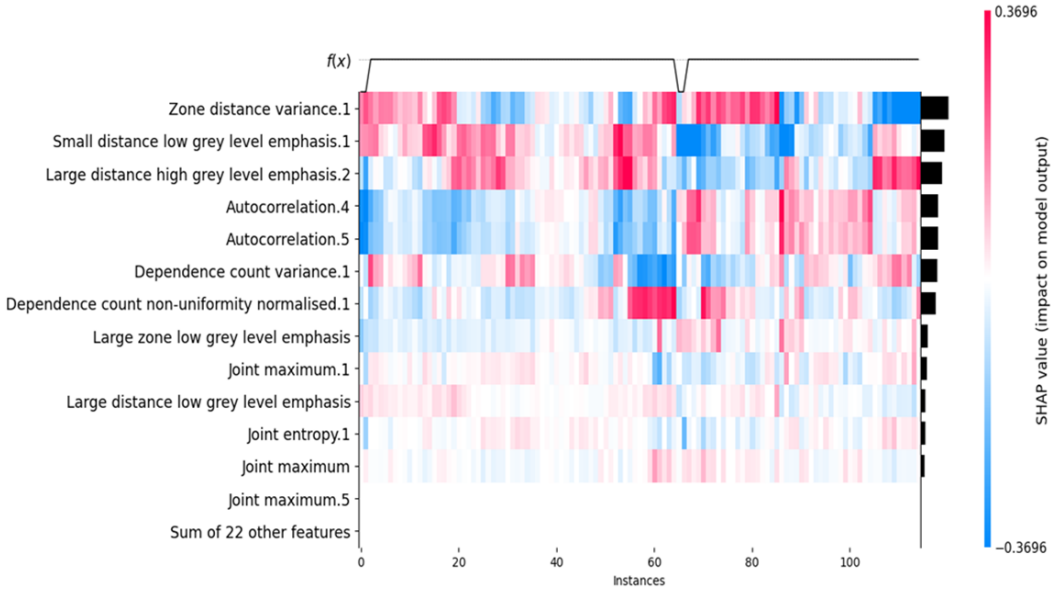

Figure 45: Heatmap plot of Shapley values of implementing designed interpretability model on Rest-Stress selected features

6.2 SHAP Plots of The Selected Model for The Rest Dataset

In Table 1 models with the best results on the 'Rest' data are illustrated, the model with 'StandardScaler' as scaling method, 'Logistic Regression (LR)' as the classifier, and 'PCA' as feature selection method and 'random search' as parameter optimization strategy had metric values as ACC = 0.74, AUC = 0.76, SPE = 0.75, and SEN = 0.7.

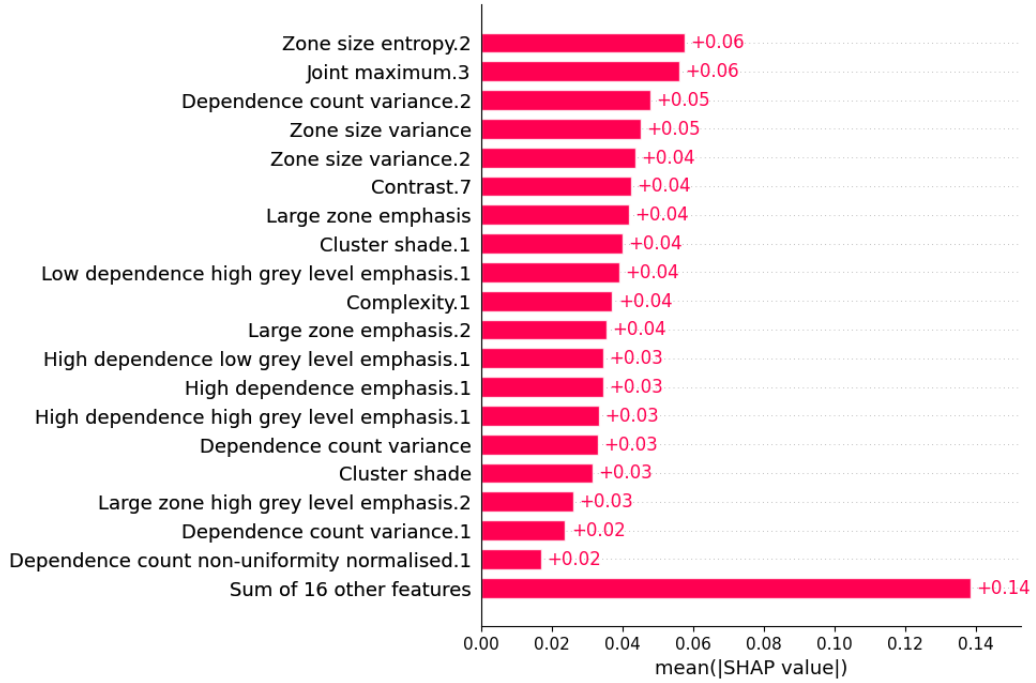

Figure 46: Bar plot of Shapley values of implementing designed interpretability model on Rest selected features

Interpretability model implemented on this model and results have been displayed in Figs, Figure 46, Figure 47, Figure 49, and Figure 50. According to plots, the feature 'Zone size entropy.2' [58], which is a radiomics feature that measures the uncertainty/randomness in the distribution of zone sizes and gray levels, has a higher Shapley value and affects predicted outcomes more than other features. A higher value of this feature shows that texture structure is more complicated and heterogeneous. Different plots also show more balanced interaction and cooperation of the features in the output result.

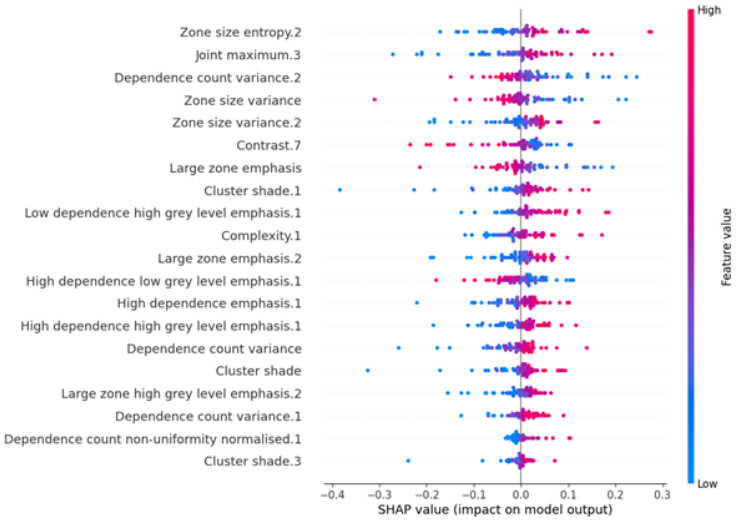

Figure 47: Summary plot of Shapley values of implementing designed interpretability model on Rest selected features

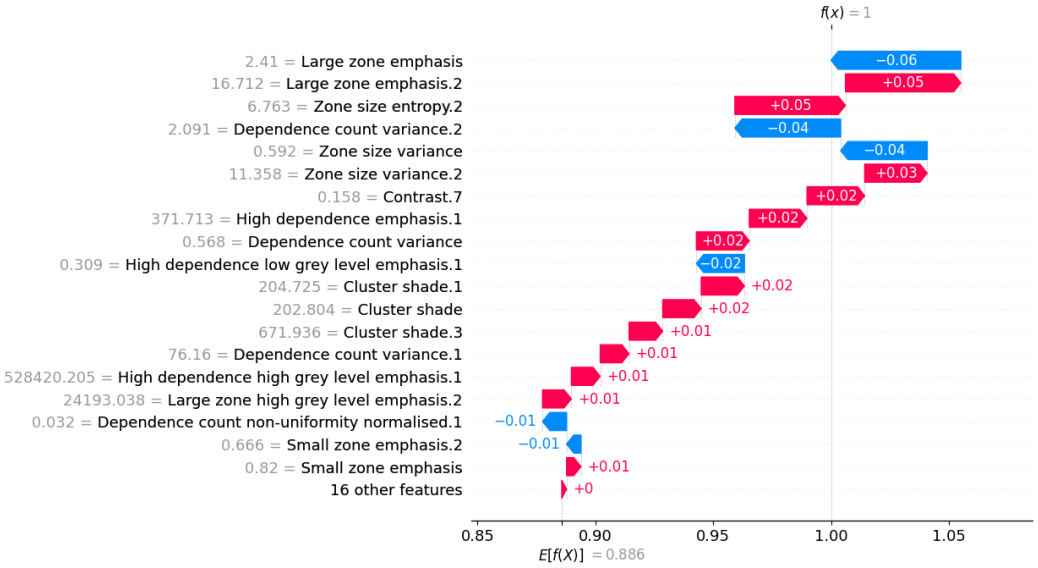

Figure 48: Summary plot of Shapley values of implementing designed interpretability model on Rest selected features

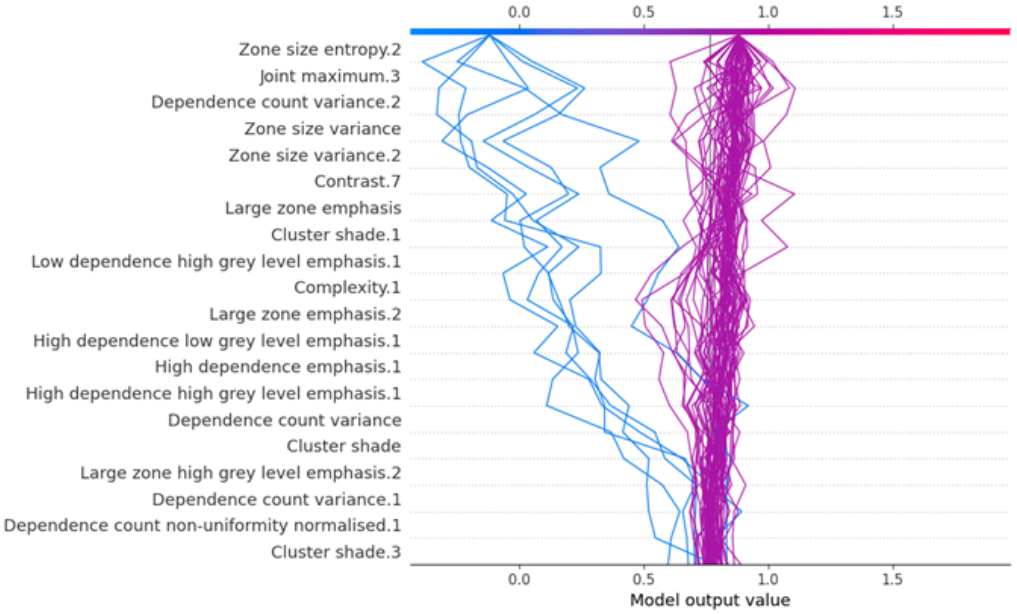

Figure 49: Decision plot of Shapley values of implementing designed interpretability model on Rest selected features

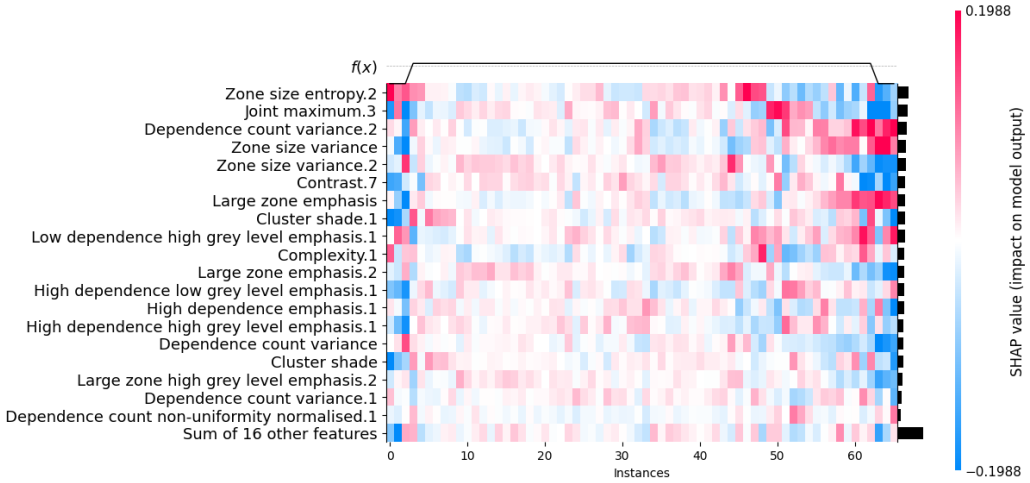

Figure 50: Heatmap plot of Shapley values of implementing designed interpretability model on Rest selected features

6.3 SHAP Plots of the Selected Models for The Stress Dataset

By implementing the model with the 'random search' for parameter optimization and 'IQR' for outlier detection on Stress data, different metrics were measured Table 3, the classification algorithm was 'K-Nearest Neighbor (K-NN)', the feature selection method was 'Select from Model' and for scaling, 'MinMax' was used. ACC = 0.87, SPE = 0.94, SEN = 0.98, and AUC = 0.94 were the metrics value of the best model. SHAP plots have been illustrated in Figure 51, Figure 52, Figure 53, Figure 54, and Figure 55.

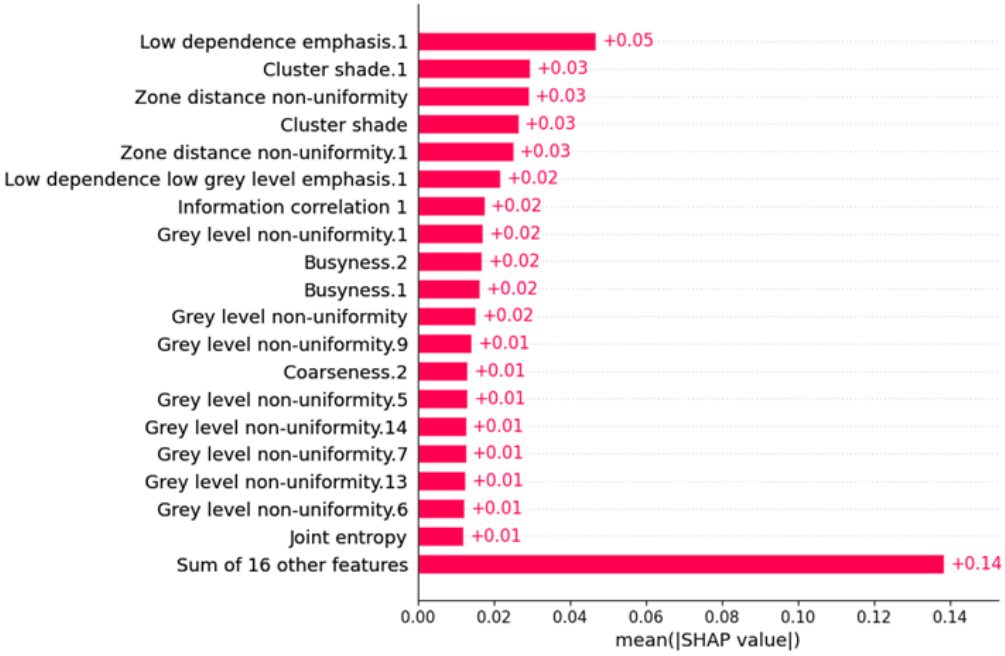

Figure 51: Bar plot of Shapley values of implementing designed interpretability model on Stress selected features

According to the drawn plots, it is obvious that the feature 'Low dependence emphasis.1' [58] has a higher impact than other features on the final result. This feature measures the number of times that pairs of pixel values with a given spatial relationship occur simultaneously. The concentration of this feature is on pairs of pixels with both low grey-level values and independence in spatiality.

However, in the majority of plots, 'Low dependence emphasis.1' has a higher effect on the outcome, in the waterfall [58] plot, 'Cluster shade' [58] highest negative value among other SHAP values.

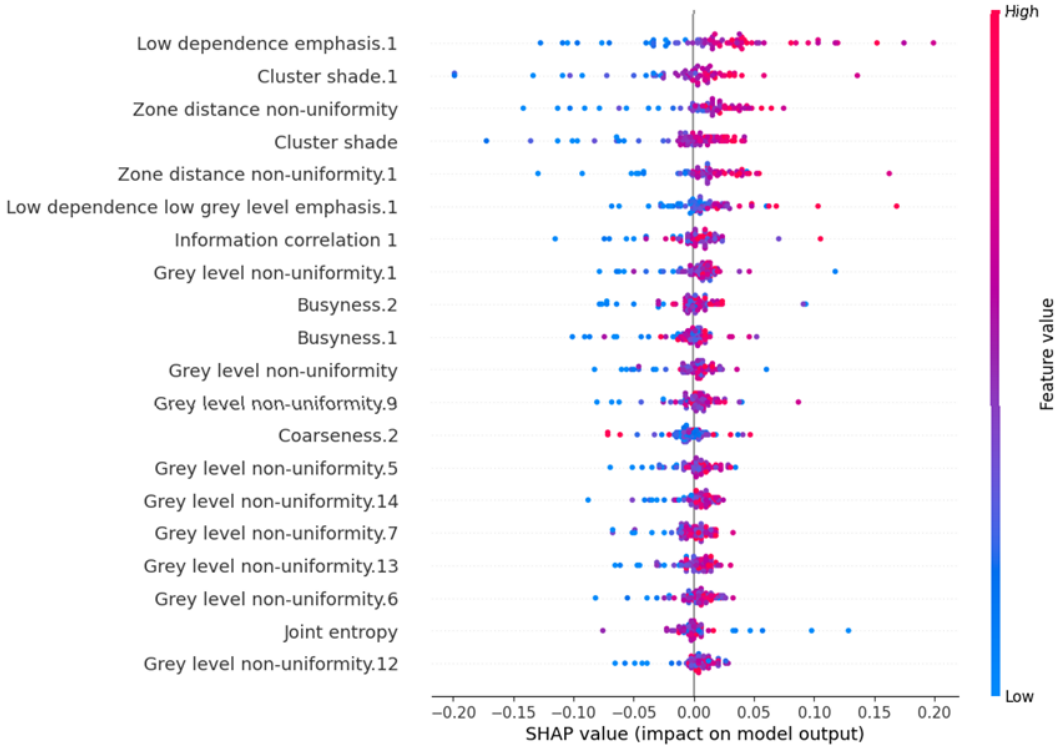

Figure 52: Summary plot of Shapley values of implementing designed interpretability model on Stress selected features

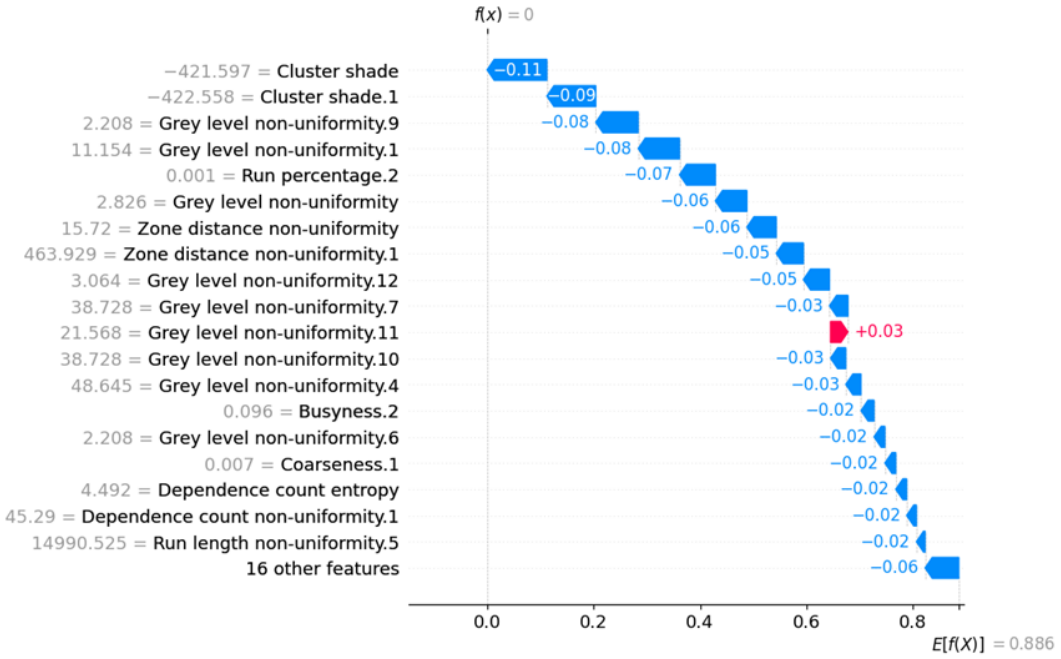

Figure 53: Waterfall plot of Shapley values of implementing designed interpretability model on Stress selected features

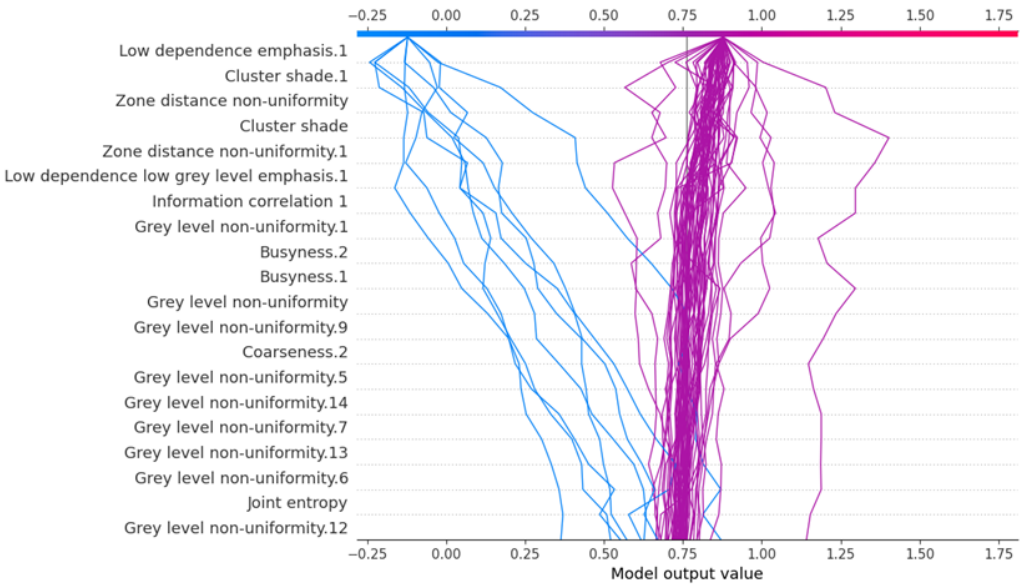

Figure 54: Decision plot of Shapley values of implementing designed interpretability model on Stress selected features

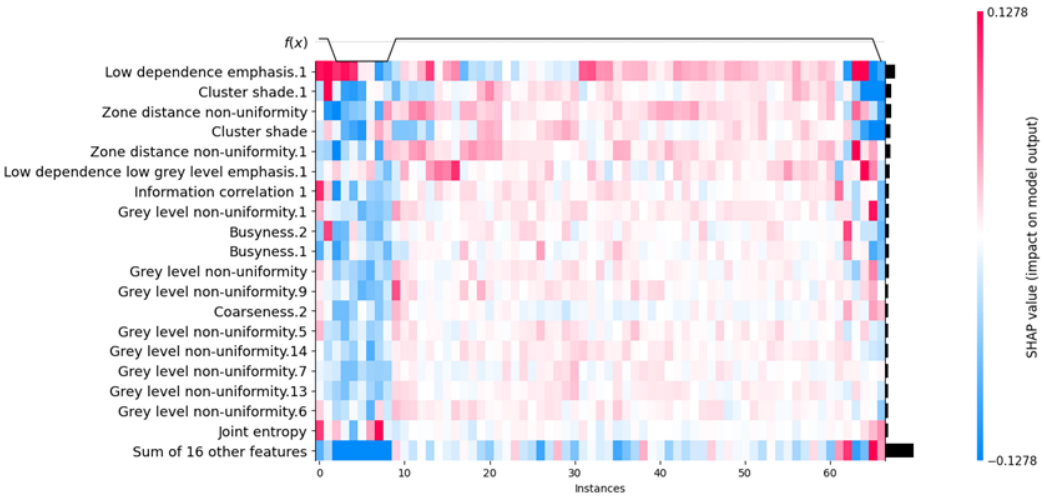

Figure 55: Heatmap plot of Shapley values of implementing designed interpretability model on Stress selected features

6.4 SHAP Plots of the Selected Model for The Rest Dataset

Until now, all parameters were optimized by random search. Another search method used in this study is 'Grid Search'. It is an exhaustive search approach implemented on one combination of classification, feature selection, and scaling strategies. Outliers of data were detected by 'IQR' and this method was implemented on Rest data. The best result based on Table 2, is a model with the 'StandardScaler' method for scaling, Logistic Regression (LR) for the classification algorithm, and for feature selection, 'PCA' used. Output value for different metrics is ACC=0.74, SPE = 0.75, SEN = 0.70, and AUC = 0.76. Different SHAP plots are drawn for this model and visible in figures, Figure 56, Figure 57, Figure 58, Figure 59, and Figure 60.

Figure 56 is the bar plot of implementing interpretability on extracted features. In the y-axis, the names of features are listed, and the x-axis shows the mean SHAP value, which represents the average contribution of each feature to the model's prediction across all data points. The larger value means, the feature, on average, has a stronger impact on the model prediction. "Zone size entropy.2" [58] and "Joint maximum.3" [58] are the most impacted features, each with a mean SHAP value of +0.05, indicating that these features consistently contribute significantly to the model's predictions. The "Sum of 16 other features" has a cumulative mean SHAP value of +0.14, indicating that while these individual features may not be listed separately, together they still have a notable impact on the prediction.

The summary plot Figure 57, gives a brief summary of how each feature affects the model's output for every instance in the dataset. The names of the features are listed in the y-axis, and ranked by importance, with the most important features at the top. The x-axis represents the SHAP values, which show the impact of each feature on the model's output. Positive SHAP values, push the model output higher, and negative SHAP values, push the model output lower. The color of each point indicates the feature value; Red means high feature value, and blue means low feature value. The top features, such as "Zone size entropy.2" [58] and "Joint maximum.3" [58], have a wide spread of SHAP values, indicating they have a significant impact on model predictions across instances. For example, "Zone size entropy.2" [58] has both positive and negative SHAP values. This indicates that depending on its value for an instance, it can either increase or decrease the model's prediction. Red points (high values) on the right-hand side of the plot for a feature show that high values of this feature increase the prediction, while blue points (low values) on the left-hand side show that low values decrease the prediction. The plot also shows how the value of a feature affects its contribution. For example, for "High dependence emphasis.1" [58], higher values (red) tend to

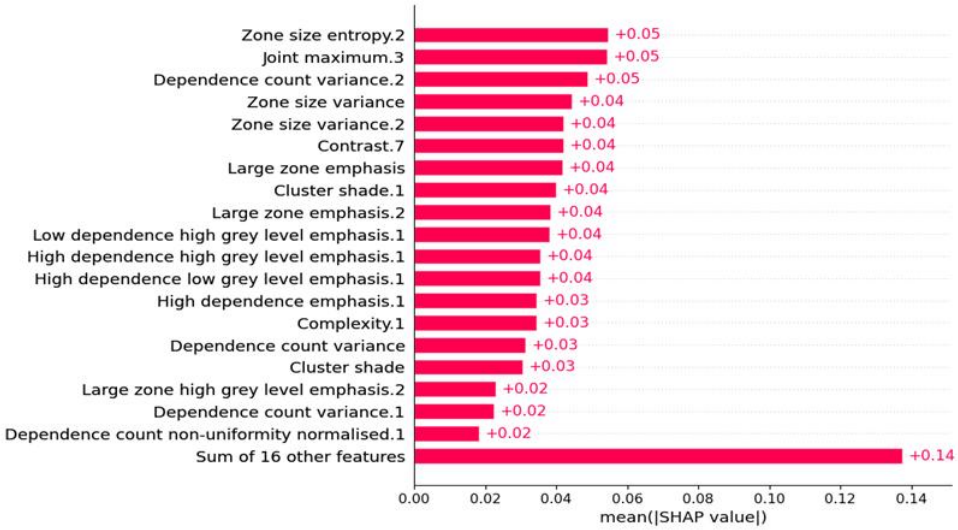

Figure 56: Bar plot of Shapley values of implementing designed interpretability model on Rest selected features

have positive SHAP values, pushing the prediction higher, while lower values (blue) tend to push the prediction lower.

Figure 58 represents SHAP waterfall plot of implementing the designed interpretability model. The waterfall plot calculates the final expected outcome by adding or subtracting the contributions of each individual feature from the base value, which is the expected output of the model. The base value is the average model output along the dataset, which is shown at the left end of the plot (E[f(x)] = 0.886), this plot starts with that value. It also has a final prediction, which is the model's output for the specific instance. This plot uses some bars in two blue and red colors. Red color bars represent features that positively contribute to the final prediction, pushing the value higher than the base value. For instance, "Large zone emphasis.2" [58] adds +0.04 to the model's prediction. Blue color bars illustrate features that negatively contribute to the final prediction. The plot moves from left to right and starts from the base value. Each feature's contribution is added or subtracted, showing the cumulative effect on the prediction. Features with more effective contributions (positive or negative) appear earlier in the flow. Features with smaller contributions appear towards the end.

Figure 59 displays the contribution of various features in the model output as a

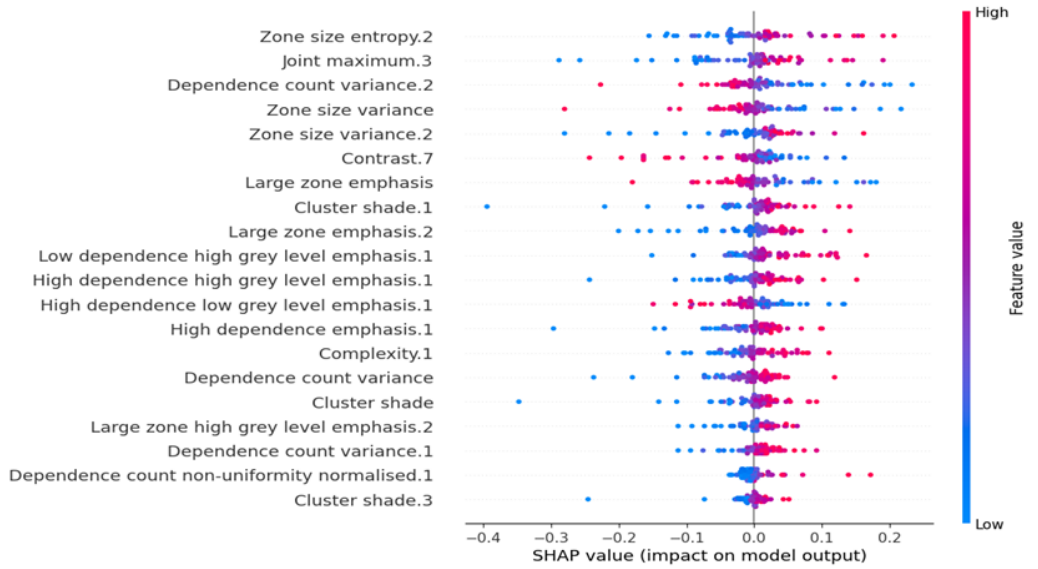

Figure 57: Summary plot of Shapley values of implementing designed interpretability model on Rest selected features

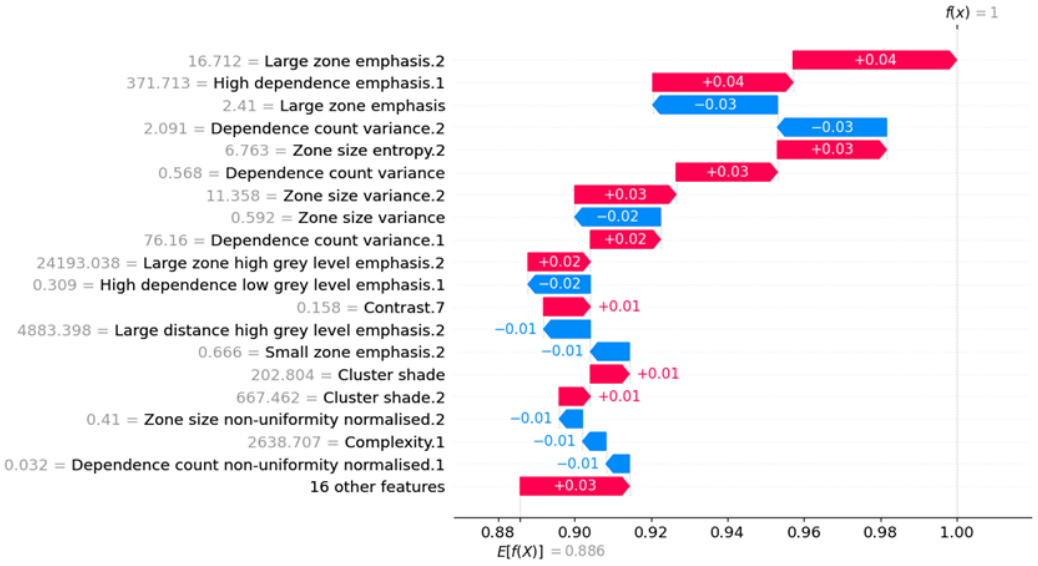

Figure 58: Waterfall plot of Shapley values of implementing designed interpretability model on Rest selected features

decision plot. In the y-axis, features are represented which are used as model input variables. These features are statistical measurements by ViSERA [58] platform. The x-axis displays the Shapley values, ranging approximately from -0.5 to 1.5. These values show the degree to which each feature influences the prediction made

by the model from the base value (shown in blue) to the final estimate (as seen in magenta). Blue lines represent features that reduce the outcome of the model. A line that shifts left (negative Shapley values) indicates that the feature reduces the assumption made by the model. Purple lines indicate features that enhance the outcome of the model. The lines show the contribution of the values of each feature to a specific output. A larger influence on the ultimate model output is indicated by a steeper slope. Features like "Zone size entropy.2" [58], "Joint maximum.3" [58], and "Dependence count variance.2" [58] have significant contributions, either positively or negatively, to the model's decision. Which features push the prediction into higher or lower outcomes are indicated by the purple (positive) and blue (negative) Shapley values. For example, features at the top of the plot typically push the estimation lower, into 0, but features at the bottom get the prediction higher, closer to 1 or higher.

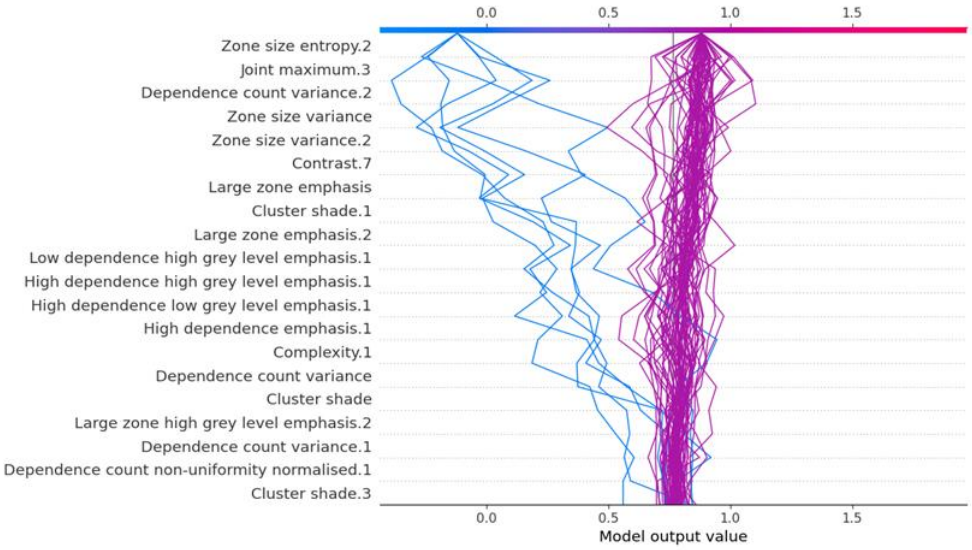

Figure 59: Decision plot of Shapley values of implementing designed interpretability model on Rest selected features

Figure 60 visualizing the impact of different features (y-axis) on the predictions of the model for multiple instances (x-axis). Red or pink color represents positive Shapley values, meaning the contribution of the feature is in a positive way to the model's prediction, while blue color meaning is the feature that pulls the prediction down. The intensity of the color reflects the magnitude of the SHAP value, with

darker shades indicating stronger impacts. As each instance's final prediction is generated by adding the Shapley values, the top black curve shows the expected model output for each instance. With both positive (red) and negative (blue) effects, the features at the top of the chart, like "Zone size entropy.2" [58] and "Joint maximum.3" [58], appear to vary greatly in their impact among various instances. Certain variables appear to have fewer changes in their SHAP values, such as "Dependence count non-uniformity normalised.1" [58] and "Sum of 16 other features" near the bottom. This suggests that they may contribute equally throughout multiple instances or have less of an overall impact.

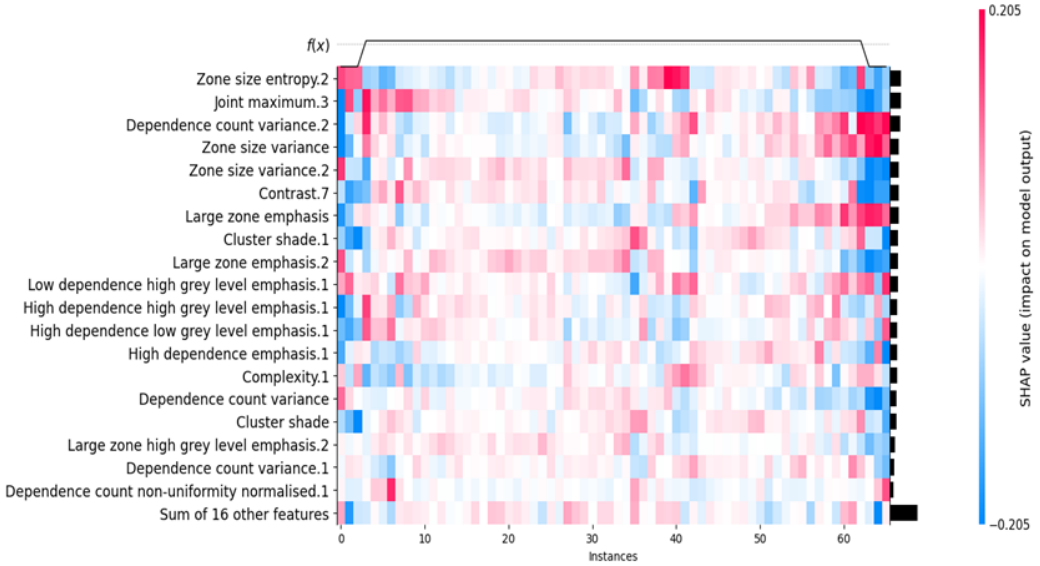

Figure 60: Heatmap plot of Shapley values of implementing designed interpretability model on Rest selected features

Chapter 7

Conclustion and future work

7.1 Future Work

While this thesis has made progress in applying interpretable ML algorithms for detecting cardiac anomalies, there are many areas for future enhancement. The focus of this study was mainly on using a basic SHAP explainer [81] to interpret models with the highest performance. Moving forward it would be beneficial to explore an array of SHAP explainers, like Partition SHAP, Tree SHAP [79], Deep SHAP [80], and Linear SHAP that are designed for types of models and data structures. By focusing on the obtained results of various explainers, a more comprehensive understanding of model behavior and factors that impact cardiac anomaly detection can be attained. To ensure the robustness and transparency of the models, including interpretability techniques to SHAP [81] methods, will play a vital role. One such method is LIME (Local Interpretable Model Explanations) [30] which offers localized approximations to explain predictions. By combining LIME with SHAP can gain insights into how the model makes decisions. It is important to develop and test a framework that links designed models and datasets. Additionally, Permutation Feature Importance (PFI) [31], Partial Dependence Plots (PDP), and Individual Conditional Expectation (ICE) are other interpretability strategies, that offer remarkable opportunities for future research. When the values of a feature are randomly rearranged, PFI helps to understand the impact of every feature on the final outcome, by evaluating the change in model performance. To explore deeper into how the model behaves across feature spaces, PDP and ICE [87, 88, 89] plots visually show the relationship between feature values and the model's prediction.

7.2 Conclusion

This study has attempted to demonstrate that using supervised classification machine learning algorithms may offer the potential to automatically identify cardiac abnormalities from data of Rest-Stress Myocardial Perfusion Imaging (MPI) Single-Photon Emission Computed Tomography (SPECT). Both rest and stress datasets with 266 samples and 401 features from the Rest-Stress MPI SPECT images to find out which combination of feature selection methods, machine learning algorithms, scaling techniques, and parameter optimization methods, when subjected to stringent testing, yield the best model. The best of these methods was that involving a classifier based on Logistic Regression, feature selection using Select from Model, RobustScaler for data scaling, and RandomSearchCV for parameter optimization. It achieved important metrics: ACC = 0.83, AUC = 0.86, SPE = 0.81, SEN = 0.81. We used SHAP (Shapley Additive exPlanations) values to make our best model interpretable to understand the feature role or contribution in the prediction. The interpretability approach emphasizes identifying high-importance features to the task, thus making analysis centered and efficient. Identifying critical features helps bring faster and more precise results for clinicians and researchers toward better detection, treatment, and management of cardiac anomalies. According to this study, interpretable machine learning models can be applied to medical diagnostics to provide more clarification and make health outcomes reliable.

Appendix A

Github

Codes of this project are available at the following link:

https://github.com/AI-in-Cardiovascular-Imaging/Interpretable-ML-Classification.git

Bibliography

[1] Mendis Shanthi, Puska Pekka, and Bo Norrving. *Global Atlas on Cardiovascular Disease Prevention and Control*. Archived (PDF) from the original on 2014-08-17. World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization, 2011, pp. 3–18. ISBN: 978-92-4-156437-3. URL: https://apps.who.int/iris/handle/10665/44701 (cit. on p. 1).

[2] World Health Organization. *Cardiovascular diseases (CVDs) fact sheet*. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Accessed: 2024-10-13. 2023 (cit. on p. 1).

[3] Wassim A. AlJaroudi and Fadi G. Hage. «Cardiovascular disease in the literature: A selection of recent original research papers». In: *Journal of Nuclear Cardiology* 28 (2021), pp. 1823–1826. DOI: 10.1007/s12350-021-02782-9. URL: https://doi.org/10.1007/s12350-021-02782-9 (cit. on p. 1).

[4] World Health Organization. *Cardiovascular Diseases (CVDs)*. https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1. Accessed: 2024-06-29. 2024 (cit. on p. 1).

[5] Pritam Saha, Ankit De, Siddhartha Dutta Roy, et al. «A comprehensive review on deep cardiovascular disease detection approaches: its datasets, image modalities and methods». In: *Multimedia Tools and Applications* (2024). DOI: 10.1007/s11042-024-18953-y. URL: https://doi.org/10.1007/s11042-024-18953-y (cit. on p. 2).

[6] Ming Yang, Reza Arsanjani, and Michael C. Roarke. «Advanced Nuclear Medicine and Molecular Imaging in the Diagnosis of Cardiomyopathy». In: *American Journal of Roentgenology* 215.5 (2020). PMID: 32901569, pp. 1208–1217. DOI: 10.2214/AJR.20.22790. eprint: https://doi.org/10.2214/AJR.20.22790. URL: https://doi.org/10.2214/AJR.20.22790 (cit. on p. 2).

[7] Thorsten M. Buzug. «Computed Tomography». In: *Springer Handbook of Medical Technology*. Springer, 2011, pp. 311–342 (cit. on p. 2).

[8] Alfred Stadler, Wolfgang Schima, Ahmed Ba-Ssalamah, Joachim Kettenbach, and Edith Eisenhuber. «Artifacts in body MR imaging: their appearance and how to eliminate them». In: *European Radiology* 17 (2007), pp. 1242–1255 (cit. on p. 2).

[9] Joseph Schoepf, Peter Zwerner, Giancarlo Savino, Christopher Herzog, J Matthias Kerl, and Philip Costello. «Coronary CT angiography». In: *Radiology* 244.1 (2007), pp. 48–63 (cit. on p. 2).

[10] Giovanni B. Frisoni, Nick C. Fox, Clifford R. Jack Jr, Philip Scheltens, and Paul M. Thompson. «The Clinical Use of Structural MRI in Alzheimer Disease». In: *Nature Reviews Neurology* 6.2 (2010), pp. 67–77 (cit. on p. 2).

[11] Ora Israel et al. «Two Decades of SPECT/CT: The Coming of Age of a Technology: An Updated Review of Literature Evidence». In: *European Journal of Nuclear Medicine and Molecular Imaging* 46 (2019), pp. 1990–2012 (cit. on p. 2).

[12] Marcelo F. Di Carli and Rory Hachamovitch. «New Technology for Noninvasive Evaluation of Coronary Artery Disease». In: *Circulation* 115.11 (2007), pp. 1464–1480. DOI: 10.1161/CIRCULATIONAHA.106.629808. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.106.629808. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.106.629808 (cit. on p. 2).

[13] G. Crișan, N. S. Moldovean-Cioroianu, D. G. Timaru, G. Andrieș, C. Căinap, and V. Chiș. «Radiopharmaceuticals for PET and SPECT Imaging: A Literature Review over the Last Decade». In: *International Journal of Molecular Sciences* 23.9 (Apr. 2022), p. 5023. DOI: 10.3390/ijms23095023 (cit. on p. 2).

[14] Yue Ming et al. «Progress and Future Trends in PET/CT and PET/MRI Molecular Imaging Approaches for Breast Cancer». In: *Frontiers in Oncology* 10 (2020). ISSN: 2234-943X. DOI: 10.3389/fonc.2020.01301. URL: https://www.frontiersin.org/journals/oncology/articles/10.3389/fonc.2020.01301 (cit. on p. 2).

[15] Sharmila Dorbala et al. «Single Photon Emission Computed Tomography (SPECT) Myocardial Perfusion Imaging Guidelines: Instrumentation, Acquisition, Processing, and Interpretation». In: *Journal of Nuclear Cardiology* 25.5 (2018), pp. 1784–1846. ISSN: 1071-3581. DOI: https://doi.org/10.1007/s12350-018-1283-y. URL: https://www.sciencedirect.com/science/article/pii/S1071358123024182 (cit. on p. 2).

[16] Johannes van Timmeren, Davide Cester, Sara Tanadini-Lang, et al. «Radiomics in medical imaging—"how-to" guide and critical reflection». In: *Insights into Imaging* 11.1 (2020), p. 91. DOI: 10.1186/s13244-020-00887-2. URL: https://doi.org/10.1186/s13244-020-00887-2 (cit. on p. 2).

[17] Philippe Lambin, Emmanuel Rios-Velazquez, Rianne Leijenaar, Sergio Carvalho, R. G. van Stiphout, Peter Granton, et al. «Radiomics: extracting more information from medical images using advanced feature analysis». In: *European Journal of Cancer* 48.4 (Mar. 2012), pp. 441–446. DOI: 10.1016/j.ejca.2011.11.036 (cit. on p. 2).

[18] Vivek Kumar et al. «Radiomics: the process and the challenges». In: *Magnetic Resonance Imaging* 30.9 (Nov. 2012), pp. 1234–1248. DOI: 10.1016/j.mri.2012.06.010 (cit. on p. 2).

[19] S. Rizzo, F. Botta, S. Raimondi, et al. «Radiomics: The facts and the challenges of image analysis». In: *European Radiology Experimental* 2.36 (2018). DOI: 10.1186/s41747-018-0068-z. URL: https://doi.org/10.1186/s41747-018-0068-z (cit. on p. 2).

[20] Chiranjib Chakraborty, Manojit Bhattacharya, Soumen Pal, and Sang-Soo Lee. «From machine learning to deep learning: Advances of the recent data-driven paradigm shift in medicine and healthcare». In: *Current Research in Biotechnology* 7 (2024), p. 100164. ISSN: 2590-2628. DOI: https://doi.org/10.1016/j.crbiot.2023.100164. URL: https://www.sciencedirect.com/science/article/pii/S2590262823000461 (cit. on pp. 3, 8).

[21] Shahadat Uddin, Arif Khan, Md Ekramul Hossain, and Mohammad Ali Moni. «Comparing different supervised machine learning algorithms for disease prediction». In: *Journal of Biomedical Informatics* (2022) (cit. on p. 3).

[22] K. Karthick, S.K. Aruna, and R. Manikandan. «Development and evaluation of the bootstrap resampling technique based statistical prediction model for Covid-19 real time data: a data driven approach». In: *Journal of Interdisciplinary Mathematics* (2022), pp. 1–13 (cit. on p. 3).

[23] J. Amann, A. Blasimme, E. Vayena, et al. «Explainability for Artificial Intelligence in Healthcare: A Multidisciplinary Perspective». In: *BMC Medical Informatics and Decision Making* 20.310 (2020). DOI: 10.1186/s12911-020-01332-6. URL: https://doi.org/10.1186/s12911-020-01332-6 (cit. on p. 3).

[24] Sajid Ali, Tamer Abuhmed, and Shaker El-Sappagh. «Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence». In: (2023) (cit. on pp. 3, 10).

[25] C. Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Interpretable*. Lulu, 2019 (cit. on p. 3).

[26] Vikas Hassija, Vinay Chamola, Abhishek Mahapatra, et al. «Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence». In: *Cognitive Computation* 16 (2024), pp. 45–74. DOI: 10.1007/s12559-023-10179-8. URL: https://doi.org/10.1007/s12559-023-10179-8 (cit. on pp. 3, 4, 10).

[27] G. Schwalbe and B. Finzel. «A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts». In: *Data Mining and Knowledge Discovery* (2023). DOI: 10.1007/s10618-022-00867-8. URL: https://doi.org/10.1007/s10618-022-00867-8 (cit. on p. 4).

[28] A. Chaddad, J. Peng, J. Xu, and A. Bouridane. «Survey of Explainable AI Techniques in Healthcare». In: *Sensors* 23.2 (2023), p. 634. DOI: 10.3390/s23020634. URL: https://doi.org/10.3390/s23020634 (cit. on pp. 4, 10).

[29] Emanuele Neri and Gayane Aghakhanyan. «Explainable AI in radiology: a white paper of the Italian Society of Medical and Interventional Radiology». In: *Italian Society of Medical and Interventional Radiology* (2023) (cit. on p. 4).

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. «Why Should I Trust You? Explaining the Predictions of Any Classifier». In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, 2016 (cit. on pp. 4, 68).

[31] Inês Neves, Duarte Folgado, Sara Santos, Marília Barandas, Andrea Campagner, Luca Ronzio, Federico Cabitza, and Hugo Gamboa. «Interpretable heartbeat classification using local model-agnostic explanations on ECGs». In: *Computers in Biology and Medicine* 133 (2021), p. 104393. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2021.104393. URL: https://www.sciencedirect.com/science/article/pii/S0010482521001876 (cit. on pp. 4, 68).

[32] A. Bhowmick, K. D. Mahato, C. Azad, and U. Kumar. «Heart Disease Prediction Using Different Machine Learning Algorithms». In: *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*. Sonbhadra, India, 2022, pp. 60–65. DOI: 10.1109/AIC55036.2022.9848885 (cit. on p. 7).

[33] V. Jain. «Heart Failure Prediction Using Machine Learning Algorithms with Cross Validations». In: *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*. Kollam, India, 2023, pp. 1420–1423. DOI: 10.1109/ICCPCT58313.2023.10245865 (cit. on p. 7).

[34] B. Naseeba, A. P. S. Haranath, S. P. Pamarthi, S. Farook, B. B. Bhanu, and B. N. K. Rao. «Cardiac Anomaly Detection Using Machine Learning». In: *Hybrid Intelligent Systems. HIS 2022*. Ed. by A. Abraham, T. P. Hong, K. Kotecha, K. Ma, P. Manghirmalani Mishra, and N. Gandhi. Vol. 647. Lecture Notes in Networks and Systems. Springer, Cham, 2023. DOI: 10.1007/978-3-031-25399-9_19 (cit. on p. 7).

[35] G. Krishnan, S. Singh, M. Pathania, S. Gosavi, S. Abhishek, A. Parchani, and M. Dhar. «Artificial Intelligence in Clinical Medicine: Catalyzing a Sustainable Global Healthcare Paradigm». In: *Frontiers in Artificial Intelligence* 6 (Aug. 2023), p. 1227091. DOI: 10.3389/frai.2023.1227091 (cit. on p. 7).

[36] Brian Rush, Leo Anthony Celi, and David J. Stone. «Applying machine learning to continuously monitored physiological data». In: *Journal of Clinical Monitoring and Computing* 33 (2019), pp. 887–893. DOI: 10.1007/s10877-018-0219-z. URL: https://doi.org/10.1007/s10877-018-0219-z (cit. on p. 7).

[37] I. Kononenko. «Machine learning for medical diagnosis: History, state of the art and perspective». In: *Artificial Intelligence in Medicine* 23.1 (2001), pp. 89–109 (cit. on p. 7).

[38] Erik Thaulow, Jan Erikssen, Leiv Sandvik, Gunnar Erikssen, Lars Jorgensen, and Peter F. Cohn. «Initial clinical presentation of cardiac disease in asymptomatic men with silent myocardial ischemia and angiographically documented coronary artery disease (the Oslo Ischemia Study)». In: *American Journal of Cardiology* 72.9 (Sept. 1993), pp. 629–633. DOI: 10.1016/0002-9149(93)90875-d (cit. on p. 7).

[39] Fabian Pedregosa et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 8–10, 14–23).

[40] J. Truett, J. Cornfield, and W. Kannel. «A Multivariate Analysis of the Risk of Coronary Heart Disease in Framingham». In: *Journal of Chronic Diseases* 20.7 (July 1967), pp. 511–524. DOI: 10.1016/0021-9681(67)90082-3 (cit. on p. 8).

[41] R. Kannan and V. Vasanthi. «Machine Learning Algorithms with ROC Curve for Predicting and Diagnosing the Heart Disease». In: *Soft Computing and Medical Bioinformatics*. Springer, 2019. DOI: 10.1007/978-981-13-0059-2_8. URL: https://doi.org/10.1007/978-981-13-0059-2_8 (cit. on p. 8).

[42] S. Uddin, I. Haque, H. Lu, et al. «Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction». In: *Scientific Reports* 12 (2022), p. 6256. DOI: 10.1038/s41598-022-10358-x. URL: https://doi.org/10.1038/s41598-022-10358-x (cit. on p. 8).

[43] Devansh Shah, Samir Patel, and Santosh Kumar Bharti. «Heart Disease Prediction using Machine Learning Techniques». In: *SN Computer Science* 1 (Nov. 2020). DOI: 10.1007/s42979-020-00365-y (cit. on p. 8).

[44] YongYang Song and Ying Lu. «Decision tree methods: applications for classification and prediction». In: *Shanghai Archives of Psychiatry* 27.2 (Apr. 2015), pp. 130–135. DOI: 10.11919/j.issn.1002-0829.215044 (cit. on p. 9).

[45] Vili Podgorelec, Peter Kokol, Bruno Stiglic, and Ivan Rozman. «Decision Trees: An Overview and Their Use in Medicine». In: *Journal of medical systems* 26 (Nov. 2002), pp. 445–63. DOI: 10.1023/A:1016409317640 (cit. on pp. 9, 22).

[46] Emil Agbemade. «Predicting Heart Disease using Tree-based Model». Master's thesis. Data Science and Data Mining, 2023 (cit. on p. 9).

[47] Adele Cutler, David R. Cutler, and John R. Stevens. «Random Forests». In: *Ensemble Machine Learning*. Ed. by Cha Zhang and Yunqian Ma. New York, NY: Springer, 2012, pp. 157–175. DOI: 10.1007/978-1-4419-9326-7_5. URL: https://doi.org/10.1007/978-1-4419-9326-7_5 (cit. on p. 9).

[48] S. Xu, Z. Zhang, D. Wang, J. Hu, X. Duan, and T. Zhu. «Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework». In: *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. IEEE. 2017, pp. 228–232 (cit. on p. 9).

[49] Ruihu Wang. «AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review». In: *Physics Procedia* 25 (2012). International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao, pp. 800–807. ISSN: 1875-3892. DOI: https://doi.org/10.1016/j.phpro.2012.03.160. URL: https://www.sciencedirect.com/science/article/pii/S1875389212005767 (cit. on p. 9).

[50] Abdulhamit Subasi. «Chapter 3 - Machine learning techniques». In: *Practical Machine Learning for Data Analysis Using Python*. Ed. by Abdulhamit Subasi. Academic Press, 2020, pp. 91–202. ISBN: 978-0-12-821379-7. DOI: https://doi.org/10.1016/B978-0-12-821379-7.00003-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128213797000035 (cit. on p. 9).

[51] Iqbal H. Sarker. *Machine Learning: Algorithms, Real World Applications and Research Directions*. Mar. 2021 (cit. on pp. 9, 10).

[52] Sourish Ghosh, Anasuya Dasgupta, and Aleena Swetapadma. «A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification». In: *2019 International Conference on Intelligent Sustainable Systems (ICISS)*. 2019, pp. 24–28. DOI: 10.1109/ISS1.2019.8908018 (cit. on p. 10).

[53] Scott M. Lundberg and Su-In Lee. «A Unified Approach to Interpreting Model Predictions». In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 4768–4777 (cit. on pp. 10, 24, 26).

[54] Lujain Ibrahim, Munib Mesinovic, Kai-Wen Yang, and Mohamad A. Eid. «Explainable Prediction of Acute Myocardial Infarction Using Machine Learning and Shapley Values». In: *IEEE Transactions on Biomedical Engineering* (Dec. 2020) (cit. on pp. 10, 25).

[55] Eleni Angelaki and Maria E. Marketou. «Detection of abnormal left ventricular geometry in patients without cardiovascular disease through machine learning: An ECG-based approach». In: *Journal Name* (Jan. 2021) (cit. on p. 10).

[56] Author Unknown. *Left Ventricular Geometry and Function in Cardiovascular Disease*. Accessed: 2024-06-30. 2024. URL: https://emedicine.medscape.com/article/2114292-overview (cit. on p. 11).

[57] Konstantinos Vrettos, Matthaios Triantafyllou, Kostas Marias, Apostolos H. Karantanas, and Michail E. Klontzas. «Artificial intelligence-driven radiomics: developing valuable radiomics signatures with the use of artificial intelligence». In: *BJR|Artificial Intelligence* 1.1 (July 2024), ubae011. ISSN: 2976-8705. DOI: 10.1093/bjrai/ubae011. eprint: https://academic.oup.com/bjrai/article-pdf/1/1/ubae011/58588824/ubae011.pdf. URL: https://doi.org/10.1093/bjrai/ubae011 (cit. on p. 12).

[58] M.R. Salmanpour, I. Shiri, M. Hosseinzadeh, H. Zaidi, S. Ashrafinia, M. Oveisi, and A. Rahmim. «ViSERA: Visualized & Standardized Environment for Radiomics Analysis - A Shareable, Executable, and Reproducible Workflow Generator». In: *2023 IEEE Nuclear Science Symposium, Medical Imaging Conference and International Symposium on Room-Temperature Semiconductor Detectors (NSS MIC RTSD)*. 2023, pp. 1–2. DOI: 10.1109/NSSMICRTSD49126.2023.10338638 (cit. on pp. 12, 53, 56, 57, 60, 63–67).

[59] Jakob Nonnenmacher, Nils-Christoph Holte, and Jorge Marx Gómez. «Tell Me Why - A Systematic Literature Review on Outlier Explanation for Tabular Data». In: *2022 3rd International Conference on Pattern Recognition and Machine Learning*. 2022 (cit. on p. 13).

[60] A. Ghoting, S. Parthasarathy, and M. E. Otey. «Fast Mining of Distance-Based Outliers in High-Dimensional Datasets». In: *Data Mining and Knowledge Discovery* 16.3 (2008), pp. 349–364. DOI: 10.1007/s10618-008-0093-2 (cit. on pp. 13, 15).

[61] M. M. Singh and N. Kane. «Outlier Detection using Ensemble Learning». In: *2022 6th International Conference on Information Technology (InCIT)*. Nonthaburi, Thailand, 2022, pp. 234–239. DOI: 10.1109/InCIT56086.2022.10067524 (cit. on p. 13).

[62] M. Pavlou, G. Ambler, R. Z. Omar, et al. «Outlier Identification and Monitoring of Institutional or Clinician Performance: An Overview of Statistical Methods and Application to National Audit Data». In: *BMC Health Services Research* 23 (2023), p. 23. DOI: 10.1186/s12913-022-08995-z (cit. on p. 13).

[63] H. C. Mandhare and S. R. Idate. «A Comparative Study of Cluster Based Outlier Detection, Distance Based Outlier Detection and Density Based Outlier Detection Techniques». In: *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. Madurai, India, 2017, pp. 931–935. DOI: 10.1109/ICCONS.2017.8250601 (cit. on p. 13).

[64] Xun Zhao, Weiwei Cui, Yanhong Wu, Haidong Zhang, Huamin Qu, and Dongmei Zhang. «Outlier Interpretation on Multi-dimensional Data via Visual Analytics». In: *Eurographics Conference on Visualization (EuroVis)*. 2019 (cit. on p. 13).

[65] Frederik Michel Dekking, Cornelis Kraaikamp, Hen Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics*. Springer Texts in Statistics. London: Springer London, 2005. ISBN: 978-1-85233-896-1. DOI: 10.1007/1-84628-168-7 (cit. on p. 13).

[66] Erwin Kreyszig. *Advanced Engineering Mathematics*. 4th. Wiley, 1979, p. 880. ISBN: 0-471-02140-7 (cit. on pp. 13, 14).

[67] Pauli Virtanen et al. «SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python». In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cit. on p. 14).

[68] T. Aljuaid and S. Sasi. «Proper imputation techniques for missing values in data sets». In: *2016 International Conference on Data Science and Engineering (ICDSE)*. Cochin, India, 2016, pp. 1–5. DOI: 10.1109/ICDSE.2016.7823957 (cit. on pp. 14, 15).

[69] Kelsy Cabello-Solorzano, Isabela Ortigosa de Araujo, Marco Peña, Luís Correia, and Antonio J. Tallón-Ballesteros. «The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis». In: *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)*. Ed. by Pablo García Bringas, Hilde Pérez García, Francisco Javier Martínez de Pisón, Francisco Martínez Álvarez, Alicia Troncoso Lora, Álvaro Herrero, José Luis Calvo Rolle, Héctor Quintián, and Emilio Corchado. Cham: Springer Nature Switzerland, 2023, pp. 344–353. ISBN: 978-3-031-42536-3 (cit. on p. 15).

[70] Ashirbad Pradhan. *Feature Scaling Technique*. Medium. Accessed: 2024-06-30. June 2024. URL: https://medium.com/@ashirbadpradhan8115/feature-scaling-technique-ada4d77aef21 (cit. on pp. 15, 16).

[71] S Santhosh Baboo and S Sasikala. «Multicategory classification using support vector machine for microarray gene expression cancer diagnosis». In: *Global Journal of Computer Science and Technology* 10.15 (2010), pp. 38–44 (cit. on p. 16).

[72] Ian T. Jolliffe and Jorge Cadima. «Principal component analysis: a review and recent developments». In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374 (2016), p. 20150202. DOI: 10.1098/rsta.2015.0202. URL: http://doi.org/10.1098/rsta.2015.0202 (cit. on p. 18).

[73] Bradley J. Erickson and Fabian Kitamura. «Magician's Corner: 9. Performance metrics for machine learning models». In: *Radiology: Artificial Intelligence* 3.3 (2021), e200126. DOI: 10.1148/ryai.2021200126 (cit. on p. 18).

[74] E. F. Schisterman, D. Faraggi, B. Reiser, and J. Hu. «Youden Index and the optimal threshold for markers with mass at zero». In: *Statistics in Medicine* 27.2 (Jan. 2008), pp. 297–315. DOI: 10.1002/sim.2993 (cit. on p. 19).

[75] Lars Buitinck et al. «API design for machine learning software: experiences from the scikit-learn project». In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122 (cit. on p. 20).

[76] Sung Yang Ho, Kimberly Phua, Limsoon Wong, and Wilson Wen Bin Goh. «Extensions of the External Validation for Checking Learned Model Interpretability and Generalizability». In: *Patterns* 1.8 (2020), p. 100129. ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2020.100129. URL: https://www.sciencedirect.com/science/article/pii/S2666389920301707 (cit. on p. 22).

[77] *RandomizedSearchCV — scikit-learn 0.24.1 documentation*. Accessed: 2024-09-17. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html (cit. on p. 23).

[78] Towards Data Science. *The Shapley Value for ML Models*. 2020. URL: https://towardsdatascience.com/the-shapley-value-for-ml-models-f1100bff78d1 (cit. on p. 24).

[79] S. M. Lundberg, G. G. Erion, and S. Lee. «Consistent Individualized Feature Attribution for Tree Ensembles». In: *CoRR* abs/1802.03888 (2018). [online] Available: http://www.arxiv.org/abs/1802.03888 (cit. on pp. 24, 68).

[80] Scott M. Lundberg and Su-In Lee. *SHAP (SHapley Additive exPlanations) Documentation: shap.DeepExplainer*. Accessed: 2024-07-30. 2023. URL: https://shap.readthedocs.io/en/latest/generated/shap.DeepExplainer.html (cit. on pp. 24, 68).

[81] Scott M Lundberg and Su-In Lee. «A Unified Approach to Interpreting Model Predictions». In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf (cit. on pp. 24–26, 68).

[82] SHAP Documentation. *shap.Explainer*. n.d. URL: https://shap-lrjball.readthedocs.io/en/latest/generated/shap.Explainer.html (cit. on p. 25).

[83] Scott M. Lundberg, Gabriel Erion, Hugh Chen, et al. «From local explanations to global understanding with explainable AI for trees». In: *Nature Machine Intelligence* 2 (2020), pp. 56–67. DOI: 10.1038/s42256-019-0138-9. URL: https://doi.org/10.1038/s42256-019-0138-9 (cit. on p. 26).

[84] Scott M. Lundberg and Su-In Lee. *An Introduction to Explainable AI with Shapley Values*. Accessed: 2024-07-02. 2023. URL: https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html (cit. on p. 26).

[85] Scott M. Lundberg and Su-In Lee. *SHAP Waterfall Plot Documentation*. Accessed: 2024-07-02. 2023. URL: https://shap.readthedocs.io/en/latest/generated/shap.plots.waterfall.html (cit. on p. 26).

[86] Scott M. Lundberg and Su-In Lee. *SHAP Heatmap Plot Documentation*. Accessed: 2024-07-02. 2023. URL: https://shap.readthedocs.io/en/latest/generated/shap.plots.heatmap.html (cit. on p. 26).

[87] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. 2nd. New York: Springer, 2009. ISBN: 978-0-387-84857-0 (cit. on p. 68).

[88] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. «Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation». In: *Journal of Computational and Graphical Statistics* 24.1 (2015), pp. 44–65. DOI: 10.1080/10618600.2014.907095 (cit. on p. 68).

[89] Christoph Molnar. *Interpretable Machine Learning*. 2019. URL: https://christophm.github.io/interpretable-ml-book/ (cit. on p. 68).