DEVELOPMENT OF A SYNCOPE CLASSIFICATION ALGORITHM FROM PHYSIOLOGICAL SIGNALS ACQUIRED IN TILT-TABLE TEST

GAN MING HONG

UNIVERSITI TUNKU ABDUL RAHMAN

DEVELOPMENT OF A SYNCOPE CLASSIFICATION ALGORITHM FROM PHYSIOLOGICAL SIGNALS ACQUIRED IN TILT-TABLE TEST

GAN MING HONG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Biomedical Engineering with Honours

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Gan Ming Hong

ID No. : 1801320

Date : 17/5/2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEVELOPMENT OF A SYNCOPE CLASSIFICATION ALGORITHM FROM PHYSIOLOGICAL SIGNALS ACQUIRED IN TILT-TABLE TEST"** was prepared by **GAN MING HONG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Biomedical Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature :

Supervisor : Dr. Goh Choon Hian

Date : 17/05/2023

Signature :

Co-Supervisor : Dr Kwan Ban Hoe

Date : 19 May 2023

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisors, Dr Goh Choon Hian and Dr Kwan Ban Hoe for invaluable advice, guidance and enormous patience throughout the development of the research.

ABSTRACT

Syncope also known as transient loss of consciousness which caused problem to human daily life. Since machine learning is much more advanced, classification of syncope can be done with machine learning. Head-up tilt table test (HUTT) having a lengthy procedure and might causing patient to feel discomfort during the test. Aim of this study is to design an algorithm which able to classify syncope patient based on their physiological signal. In this study, electrocardiogram (ECG) and blood pressure (BP) signal has been collected from 144 subjects with head-up tilt table test (HUTT) by clinicians. Several features have been extracted from heart rate, systolic and diastolic blood pressure. There are 8 set of feature selection model has built and a total of 24 set of classifiers with 3 different type of classification techniques were developed. Additionally, stratified 5-fold cross-validation was performed to evaluate the performance of proposed model. Features that selected for the classification is mean of systolic and diastolic blood pressure, standard deviation of real variability of diastolic blood pressure, and the mean of systolic blood pressure in low and high frequency ratio. The proposed model yielded the following result: 85.71% sensitivity, 91.43% specificity, 88.18% F1-score and 88.57% accuracy. Future work can be focus on utilise more different type of classifier and carry out external cross validation for achieving a better classification model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

TLOC	transient loss of consciousness
HUTT	head-up tilt table test
ILR	implantable loop recorder
ECG	electrocardiogram
ML	machine learning
AI	artificial intelligence
NTG	nitroglycerine
ICG	impedance cardiography
CO	cardiac output
PCA	principle component analysis
SD	standard deviation
CV	coefficient of variance
ARV	average real variability
RMSRV	root mean square of real variability
SDRV	standard deviation of real variability
HRV	heart rate variability
BPV	blood pressure variability
LF	low frequency
HF	high frequency
HR	heart rate
LVET	left ventricular ejection time
MCAR	missing complete at random
MNAR	missing not at random
SFS	sequential forward selection
SBS	sequential backward selection
RFE	recursive feature elimination
SVM	support vector machine
GA	genetic algorithm
SMOTE	synthetic minority over-sampling technique
SVR	support vector regression
ROC	receiving operating characteristic
UMMC	University of Malaya Medical Centre

CI	cardiac index
RRI	RR-interval
SI	stiffness index
SV	stroke volume
SpO2	oxygen saturation
TPR	total peripheral resistance
TRPI	total peripheral resistance index
dBP	diastolic blood pressure
mBP	mean blood pressure
sBP	systolic blood pressure
NaN	not a number

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Syncope is defined as transient loss of consciousness (TLOC) due to global cerebral hypoperfusion, which is characteristically of rapid onset, brief duration with complete spontaneous recovery (Brignole et al., 2018). It is a common condition, with 18.9 – 39.7 per 1000 patient episodes reported in the general population (Brignole et al., 2018). The Framingham Heart Study reported an overall incidence rate of 6.2 per 1000 person-years with increased incidence with age, and a sharp increase after 70 years (Walsh et al., 2015). An incidence rate of 11.1 per 1000 person-years has been assigned to those aged 70 to 79 years and 18.25 per 1000 person-years for those aged 80 years and above (da Silva, 2014). Approximately 40% of the U.S. population experienced a syncopal episode in their lifetimes, with 30% to 50% admitted to the hospital for further evaluation, and one-third of cases were classified with an unexplained etiology (Runser et al., 2017).

Syncope can be classified into three main types: neurally-mediated or neurocardiogenic or reflex, orthostatic hypotension and cardiac syncope. Neurally-mediated syncope is by far the most common type of syncope. The brief loss of consciousness is attributed to a neurologically induced drop in blood pressure and/or a decrease in heart rate. Orthostatic hypotension is typically characterized by postural-induced hypotension and is most often related to impaired in systemic resistance (Runser et al., 2017) . Associated factors include medication effects, volume depletion, acute haemorrhage, and autonomic dysfunction (Runser et al., 2017). Cardiac syncope could occur as a result of cardiac arrhythmias, structural defects or perfusion issues. Many cases of syncope remain unexplained, and this has been attributed to lack of structured evaluation and diagnostic capabilities (Sutton, 2013).

Diagnostic strategies for syncope may include head-up tilt table test (HUTT) and implantable loop recorder (ILR) (Ungar et al., 2013). HUTT is an orthostatic stress test to assess the susceptibility of the vasovagal response to an orthostatic challenge (Shen et al., 2017). Patients are tilted to 70 degrees for up to 40 minutes (Shen et al., 2017). The American Heart Provocation with low dose of isoproterenol infusion or sublingual nitrates is usually used to improve the sensitivity of the test. The ILR is developed to permit long term cardiac monitoring to capture the electrocardiogram (ECG) during a spontaneous episode in patients without recurrence in a reasonable time frame (Kenny and Krahn, 1999). The ECG is recorded in a continuous loop and stored if the device is activated either automatically using arrhythmia detection algorithms or manually using an external device (Bisignani et al., 2019). Thus, the ILR is able to record the information before, during and after the event, to facilitate accurate diagnosis.

1.2 Importance of the Study

As the number of syncope cases is under an increasing trend, especially for aging patient, it is becoming a troublesome issue. Although the overall mortality rate is relatively low, it rises sharply with increasing age. The annual mortality rate for patients aged 70 to 79 years is 14%, rising to 22% for 80 to 89 years old patients, while reached 43% for patients that above 90 years old (Wong, 2018). This showed that the probability of syncope causing death in elderly is much higher compared to adult. Not just causing mortality among syncope patient, also affecting the quality of life and interfering with the daily activities with potential occupational implications (McCarthy et al., 2020).

Technologies like machine learning (ML) and artificial intelligence (AI) are becoming more advanced, their capabilities in classification and prediction are more mature, and their involvement in healthcare field will bring benefits to doctors and patient. Integration of AI with HUTT will resulting early syncope detection which significantly reduce the morbidity and mortality rate. With aids of ML algorithm in predicting the outcome of HUTT, the lengthy procedure will be able to shorten, significant gain in the efficiency and cost-saving for healthcare services (Hussain et al., 2021).

1.3 Problem Statement

As for current, the diagnosis method on syncope is still using HUTT, which is phenomenological and the corresponding terminology is inconsistent (Brignole, 2007). According to The Italian Protocol, the best methodology of HUTT is 5 minutes of stabilization in supine position, 20 minutes at a tilt angle of 60 degrees and a further 15 minutes after injection of 400μg nitro-glycerine (NTG) sublingual spray (Bartoletti et al., 2000). In order to finish one set of HUTT, at least 40 minutes needed, and some patients might be requested by healthcare provider to carry out second HUTT for conformation on previous test result. Due to the procedure of HUTT, it led to time consuming. Patient has a high chance to experience severe hypotension or bradycardia before the test is terminated, for collecting sufficient information for HUTT (He et al., 2021). Patients need to be tilted to a certain degree which cause them to feel discomfort and unsuitable for physically weak patient.

1.4 Aim and Objectives

The aim of this study is to design an algorithm which able to classify syncope patient based on their physiological signal which can aid healthcare provider in their justification for treatment planning. The objectives of this study are:

1. To conduct a review search and understanding on hemodynamic parameters relevant to syncope.
2. To conduct a review search and understanding on machine learning algorithms applied in syncope classification.
3. To design syncope classification algorithm with accuracy of 85% and above.
4. To evaluate the performance of designed algorithm with state-of-the-art algorithms

1.5 Scope and Limitation of the Study

In order to build the algorithm for syncope classification, raw data is needed to train and valid the algorithm. The used raw data must only be collected under HUTT, strictly not with ILR.

Limitation on this study is that the patient data that used to train and valid the classification algorithm was collected by one medical centre. Outcome of the algorithm might have bias since the algorithm was constructed with patient data that from same medical centre.

Another limitation of the study is impedance cardiography (ICG) is not used in this study. ICG is a non-invasive measure of changes in thoracic impedance generated by fluctuating blood volume during cardiac cycle, allows calculation of stroke volume and cardiac output (CO) (Parry et al., 2009). The fall of blood pressure during vasovagal syncope is mediated initially by decreased CO and reduction of CO may be the primary cause of the hypotension of vasovagal syncope, hence the use of ICG might improve the predictive value of ML algorithm (Wieling et al., 2016).

1.6 Contribution of the Study

In this study, a binary classification model of syncope by using physiological signal that obtained by HUTT was built and achieved accuracy of 85%. With this proposed model in this study, it able to aid in decision making of clinician as classification model provides another reference for clinician to consider, which this model is useful when patient present with unknown syncope status signal.

1.7 Outline of the Report

This report is mainly describing the work has been done to complete this project. Literature that related to this study have been reviewed in Chapter 2 and Chapter 3 is more focus on the workplan and methodology on developing the algorithm. Performance of the algorithms and discussion of the study was mentioned in Chapter 4 while Chapter 5 concluded the overall study.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is mainly included the literature review which related to the topic such as process of classification and the related work on classification of syncope.

2.2 Machine Learning and Classification

Machine Learning (ML) algorithm are organized into a taxonomy which based on the desired outcome of the algorithm while the function that maps input data to the desired output is generated by supervised learning (Osisanwo et al., 2017). ML algorithms are separate into few types or groups which are supervised learning, unsupervised learning and semi-supervised learning. Supervised learning generated a function that maps input to the desired output which classification is one of the standard formulations that is required to learn a function and maps a vector into one of several classes by looking at several input-output example of the function (Nasteski, 2017). Semi-supervised learning used the combination of both unlabelled and labelled example to generate the function (Nasteski, 2017).

Classification is significant to data analytics, ML and pattern recognition which used supervised learning technique to categorizes the obtained data from the prior information (Singh et al., 2016). Classification is not only limited on structured dataset, also applicable on unstructured data (Sen et al., 2020). Classifier algorithm learns and concludes some valid mapping function from the training dataset and predict the outcome or class label with the help of the mapping function (Sen et al., 2020). Binary and multi-label classification are the most common type of classification used. Binary classification output two possible outcome such as positive or negative and yes or no while multi-label classification are suitable on the application that needed more than two possible outcomes such as academic performance of student as excellent or good or poor (Sen et al., 2020).

The process to construct a complete classification algorithm is distributed into four parts. First, the data set is collected and undergo pre-processing. Second, related features are extracted from raw dataset. Next, feature selection was carried out to determine the most suitable feature. The last step of the construction of classification algorithm is using the selected features for model training (Singh et al., 2016).

2.2.1 Data Collection

The purpose of data collection is to obtain a set of data that able to be used in training ML models. Data discovery, data generation and data augmentation are the three methods for data collection (Roh et al., 2019). Data discovery is mean to share or search for new datasets as it is become more significant when more datasets are available on the database. Data augmentation is done by adding external data for enhancing the existing dataset and as a complement of data discovery. Data generation is applied when there is no suitable dataset and generate crowdsourced or synthetic dataset. Crowdsourcing is the standard method for manual data construction (Roh et al., 2019).

2.2.2 Feature Extraction

Feature extraction is a general method to create a transformation of the input space into a low-dimensional subspace that preserve most of related information (Chumerin and Van Hulle, 2006). Feature extraction able to reduce the complexity and simplified the representation of the data by representing each variable in feature space as a linear combination of original input variable (Khalid et al., 2014). Principle Component Analysis (PCA) which introduced by Karl is the most popular and widely used feature extraction method. PCA is a simple and non-parametric approach to extract the most relevant information from a set of noisy data. It also a linear transformation of data that reduce the redundancy and maximize the information by measuring the variance (Khalid et al., 2014).

From Table 2.1, the feature that extracted by Ferdowsi et al. (2022) for syncope classification algorithm are the standard deviation (SD), coefficient of variance (CV), average real variability (ARV), root mean square of real variability (RMSRV) and standard deviation of real variability (SDRV) of the heart rate variability (HRV) and blood pressure variability (BPV). In this study, the low frequency (LF) power, high frequency (HF) power and the ratio of low frequency to high frequency power (LF/HF ratio) were extracted. In the study of Couceiro et al. (2015), the authors extracted the heart rate (HR) and left ventricular ejection time (LVET) from ECG data with feature selection score. Miranda and da Silva (2016) have extracted HRV and the LF, HF, LF/HF for its classification model with the accuracy of 92.2%.

2.2.3 Missing Data Management

One of the common problems in medical research is missing data. There are few types of missing data, where two common missing data are missing completely at random (MCAR) and missing not at random (MNAR). The missing data of MCAR is completely random, where the patient characteristic does not have any relation with the missing data (Donders et al., 2006). When the missing data depends on the actual value of missing data, it is classified as MNAR as it is related unobserved patient characteristic (Scheffer, 2002).

Missing data can be solved by imputation such as deletion method and single imputation method. Deletion method is a traditional missing data technique that discards the cases with missing data. Although this method has the advantage of produces a complete data set, it reduces the total sample size, resulting the significance test lack power (Baraldi and Enders, 2010). Single imputation has included mean imputation, where using the mean value to replace the missing data; regression imputation using a regression equation to compute the predicted scores and replace it (Baraldi and Enders, 2010).

2.2.4 Feature Selection

From features extracted, there has some are irrelevant, misleading or redundant and causing the difficulty in processing the algorithm and reduce the accuracy of the classification (Khalid et al., 2014). Thus, feature selection is the process of selecting the best and suitable features for the classification. According to (Feuilloy et al., 2006), feature selection were categorised into three category which are exhaustive search of a feature subset, heuristic method and randomized search regroups methods. In exhaustive search, all the feature subsets are then evaluated and remain the optimal solution. Heuristic method is used to increase the exploration space by decrease the cost of computation. The popular heuristic methods are Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS). The randomized search regroup method which concepted on random or probabilistic processes and generate different output by changing the input by a random source (Feuilloy et al., 2006). SFS is a bottom-up search, that start with empty set and continue to add best features one at a time, based on the cross-validation score and stop when the predetermined number of features are selected or the performance stop increasing (Vergara and Estévez, 2014). SBS works in a vice versa, which included all the features and eliminate the lowest priority feature.

Recursive Feature Elimination (RFE) is a feature selection method that able to interpret the direction and strength of association between the predictor and output which is suitable used on biomedical data (Sanz et al., 2018). RFE mostly used with support vector machine (SVM) which RFE eliminate the feature by using the SVM weighs as a ranking criterion (Rustam and Kharis, 2020). In the study of (Ferdowsi et al., 2022), RFE was used for the feature selection which reduce the size of the data by decreasing the number of characteristics in data set and choosing the best features for the classification. (Huang et al., 2014) used RFE in their study and obtain the optimum feature subset after removing the features with minimum weight that determined in every iteration.

Genetic Algorithm (GA) is a wrapper-based feature selection technique that search for the besr feature subset by mimicking the natural evolution process of man (Babatunde et al., 2014). In study of (He et al., 2021), GA was selected as the feature selection method as it able to prevent overfitting and reducing the interference of noise of the model. The basic procedure of GA are endocing, population initialization, fitness evaluation, selection, crossover and mutation (He et al., 2021).

2.2.5 Imbalance Data Management

A set of data with a not equivalent ratio or portion of positive and negative data set is concluded as imbalance data which is a challenging problem in binary classification as it caused bias and affecting the performance of classification. In order to solve this issue, Synthetic Minority Over-sampling Technique

(SMOTE) has been introduced by Chawla, Boywer, Hall and Kegelmeyer, which proven SMOTE is more effective in dealing imbalanced data problem (Chawla et al., 2002). The minority class is over-sampled by interpolating the synthetic instances between existing examples in the minority class (Bunkhumpornpat et al., 2012).

2.2.6 Classification Algorithm

Classification algorithm is technique of supervised machine learning which utilizes the previous and present data to gain knowledge with the aid of label to forest cast, and compare the result with actual and expected result to identify error to change the model based on results (Saravanan and Sujatha, 2018).

2.2.6.1 Support Vector Machine

An advanced supervised algorithm has invented with the ability to deal with both regression and classification task that more favourable to classification is called Support Vector Machine (SVM) (Sen et al., 2020). Although SVM is more complex compared to other algorithm, but it provided a higher accuracy without overfitting and suitable for linear and non-linear dataset (Singh et al., 2016). Normally, SVM are revolving around the ‘margin’, which is hyperplane to separate two different class labels of the data. ‘Kernel’ function is the main factor that changing SVM from linear classification to nonlinear classification. There are four core kernel function that determine the linearity of the algorithm which are linear, polynomial, radial basis kernel and sigmoid kernel function (Huang et al., 2014). By changing the parameter such as C and γ, it can reduce the complexity of SVM and improve the efficiency of calculation (Huang et al., 2014).

2.2.6.2 Logistic Regression

The concept of logistic regression is extracting some set of the weighted features from the input and calculation their log value, combining them linearly (Nasteski, 2017). This technique commonly specific the boundary between the classes exists and class probabilities depend on distance from the boundary. The application of this technique is on classification that use single multinomial logistic regression model with single estimator (Osisanwo et al., 2017).

Logistic regression able to carry out good probabilistic interpretation and new data set can be added to the model easily by using online gradient descent method. The advantage of using logistic regression as the classification algorithm is able to handle the interaction effect, non-linear effect and power terms (Singh et al., 2016). However, some researches shows that logistic regression is inefficient and inaccurate by comparing with other advanced machine learning technique (Saravanan and Sujatha, 2018). In order to increase the model's stability for accuracy, large sample is required to train the model and might suffering from multicollinearity (Singh et al., 2016).

2.2.6.3 K-Nearest Neighbour

Another classification algorithm that well known in supervised learning is k-Nearest-Neighbour (kNN). kNN is a non-parametric classification algorithm which assign to an unlabelled sample point and the class of the nearest of a set of previously labelled point (Singh et al., 2016). It will store all the available record or input and predict the class of a new instances that giving attention to similarity measurement from the nearest neighbour. This technique is well suited for multimodal classes as it allows multiple labelling on the input data (Singh et al., 2016). Important factor in this classification technique is the value of 'k', which represent the number of nearest neighbours who's used to predict label for a new record around (Guo et al., 2003). The value of 'k' will affect the accuracy of the model as the decision boundary is highly dependent on 'k'. There is no fixed method to determine the suitable 'k' value, only is to run the algorithm many times with different 'k' values and choose the best performance (Guo et al., 2003).

2.2.6.4 Decision Tree

Decision tree is a classification technique which flow-chart-like structure, where it is made up from root, internal and leaf nodes, where each internal node denotes a test condition on an attribute, leaf node represents a class label and branch indicates the outcome of test condition (Song and Ying, 2015). Decision tress utilizes data mining induction techniques which partitions the data with breadth-first approach or depth-first greedy approach until all data group to a particular class (Jadhav and Channe, 2016). Tree building and tree pruning are two phases that performed during classification. Tree building is done in a top-down direction while tree pruning is performed in a bottom-up approach for improving the classification's accuracy (Jadhav and Channe, 2016). (Song and Ying, 2015) mentioned that decision tree approach is popular in medical research such as used in diagnosis the medical condition by study the pattern of symptoms.

2.2.6.5 Random Forest

Random forest is another type of tree-based classifier which based on random vector sampled from the input vector and each tree casts a unit vote for the most popular class to classify an input vector (Pal, 2005). Random forest classifier is popular in biology and medical field due to the high predictive accuracy. Random forest is a group of un-pruned classification tress that developed from randomly select sample from training data by induction process and majority vote for classification result the prediction of the ensemble (Ali et al., 2012).

2.3 Related Works

The studies that included in Table 2.1 is related the classification syncope by using machine learning. SVM and support vector regression (SVR) is the most used classification algorithm to classify the result. Among 11 studies included, 4 studies have used SVM or SVR, 4 studies used logistic regression, 4 studies used receiver operating characteristic (ROC) analysis and 3 studies used kNN. In the study of (Couceiro et al., 2016), the algorithm able to achieve sensitive of 95.2%, specificity of 95.4% and 95.4 % accuracy, by using ROC analysis as the classification algorithm, HR and LVET as the parameter and compromise with drug application, which can considered it as the best algorithm performance among all the studies as the performance metric didn't have any bias.

From the table, all included studies are collecting ECG signal and eight studies have recording BP during HUTT process. ECG is important in syncope as abnormal ECG indicated the possibility of cardiac syncope (Brignole et al., 2001) and continuous blood pressure monitoring is significant during assessment (Brignole et al., 2018a). Therefore, the parameters that extracted from ECG such as HR and HRV are important factor in determine the accuracy of the algorithm. Features such as heart rate variability (HRV), RR-interval (RRI), diastolic blood pressure (DBP), systolic blood pressure (SBP), mean blood pressure (MBP), heart rate (HR), cardiac output (CO), stroke volume (SV) and total peripheral pressure (TPR) are extracted by all the studies included.

Based on Table 2.1, the range of sensitivity that achieved by ML algorithm is 52.8% to 97.4% while the range of specificity is 56% to 97.3%, accuracy from 67.6% to 95.4%and the range of PPV is 75% to 91.7%. The highest sensitivity is 97.4% which is from the studies of Miranda, C. M. and R. da Silva (2016) while the highest specificity is 97.3% from Mereu, R., et al. (2013). Although Miranda, C. M. and R. da Silva (2016) were able to achieve the highest sensitivity, their reported specificity was low, 83.3%. The sensitivity of the RR/SBP combination for Mereu, R., et al. (2013) study is 52.8% which is considered as low, although the specificity of that combination is the highest. Hence, the performance of these two studies is not good enough to correctly classify syncope patient.

Throughout all the comparison, Coureiro et al. (2016) achieved the best performance among all the studies. Since the performance of Coureiro et al. (2016) reached 95.4% accuracy, it indicated that machine learning has the ability or suitable use for classifying syncope.

Table 2.1: Characteristic of each included studies

Article	Subjects		Age range	HUTT protocol	Type of signals	Feature extraction algorithm	Parameters extracted	Classification algorithm	Performance Metrics			
	No. Subjects (n)	Men (n, %)							Sensitivity (%)	Specificity (%)	Accuracy (%)	PPV (%)
Ciliberti, M. A. P., et al. (2018)	26	11 (42.3)	21 - 58	HUTT (30 mins resting state + 45 mins 60 degree) + (15mins NTG)	ECG, BP	-	HRV, VLF, LF, HF, LF/HF ratio	Univariate analysis, multivariable analysis, logistic regression	87.5	72.2	76.9*	75
Couceiro, R., et al. (2015)	43	23 (53.5)	39 - 80	HUTT (15 mins lying rest + 20mins 70 degree) + (15mins NTG)	ECG, PPG	Feature selection score	HR, LVET	ROC analysis	95.2	95.4	95.4*	90.9

He, Z., et al.(2021)	209	76 (36.4)	22.4 - 61.4	HUTT (5 mins supine + 20mins 70 degree) + (15 mins NTG)	ECG	GA	HR, RRI, SBP, DBP, MBP, LVET, TPR, CO, SV	SVR, LR, KNN, RF	SVR: 86 LR: 82 KNN: 84 RF: 81	SVR: 82 LR: 71 KNN: 81 RF: 79	SVR:84.2* LR: 63.2* KNN: 83.3* RF:80.3*	-
Ferdowsi et al., (2022)	52	-	-	HUTT (10 mins supine+ 20 mins 70 degree) +(GTN)	ECG, BP	-	HR, HRV_ARV, HRV_SDRV, HRV_HFnu, HRV_LFnu, SBP, SBPV_CV, SBPV_SDRV, SBPV_HFnu, SBPV_LFnu, DBP, DBPV_CV,	SVM	88.9	85.7	86.5	84

							DBPV_SDRV, DBPV_HFnu, DBPV_LFnu (in supine and 70 degree)					
Khodor, N., et al.(2016)	57	-	18 - 35	HUTT (15 supine + 45mins 80 degree)	ECG, BP	Relief method, SFS, Probe feature algorithm	RRI, Amps, dPdt_max, PTT	KNN, SVM	KNN: 86.4 SVM: 87.5	KNN: 87.9 SVM: 93.8	KNN: 86.0* SVM: 89.5*	87.5
Khodor, N., et al.(2014)	66	-	18 - 35	HUTT (11 mins supine + 45 mins 80 degree)	ECG, BP	DFA, SampEn	RRI, SS-interval	KSVM	88.5	80.6	84.8*	-
Klemenc, M. and E. Strumbelj (2015)	92	38 (41.3)	16 - 82	HUTT (5 mins stabilization + 45 mins 65 degree + 5 mins	ECG, BP	Linear regression	HRV, BRS, RRI	Logistic regression	-	-	80.6*	-

				final) + (15 mins NTG)								
Mereu, R., et al. (2013)	145	59 (40.7)	7 - 82	HUTT (5 mins supine + 35mins 60 degree)	ECG, BP	-	RRI, SBP, DBP, MBP, RR/SBP, dRR/SBP, dRR/DBP, dRR/MBP, dRR/PP	ROC analysis with classification	RRI: 84.4 SBP:88.9 DBP:87.4 MBP:86.2 RR/SBP:52.8 dRR/SBP:86.2 dRR/DBP:61.2 dRR/MBP:80.6 dRR/PP:82.0	RRI: 74 SBP:67.2 DBP:79.5 MBP:72.7 RR/SBP:97.3 dRR/SBP:89.1 dRR/DBP:93.2 dRR/MBP:86.4 dRR/PP:93.2	RRI:78.6* SBP: 77.9* DBP: 83.4* MBP: 79.3* RR/SBP: 74.5* dRR/SBP: 87.6* dRR/DBP: 67.6* dRR/MBP: 83.4*	-

											dRR/PP: 87.6*	
Miranda, C. M. and R. da Silva (2016)	64	35 (54.7)	14 - 77	HUTT(10 mins supine+20 mins 70 degree) + (15 mins isosorbide)	ECG	-	HRV, LF, HF, LF/HF	ROC analysis	97.4	83.3	92.2*	85.3
Mossello, E., et al. (2018)	372	146 (39.2)	>65	HUTT (5 min supine + 20 min 60 degree) +(NTG)	ECG, BP	-	-	Multinomial logistics regression	82	56	75.9*	-
Zhang, Z. N., et al.(2020)	176	86 (48.9)	5 - 17	HUTT(Duration not specified)	ECG, BP	Multivariate logistic regression	SBP, DBP, HR	Logistic regression	89.3	80.8	90.9*	91.7

Footnote: * Accuracy is back-calculated, PPV: positive predictive value, HUTT: head-up tilt test, ECG: electrocardiogram, BP: blood pressure, HRV: heart rate variability, VLF: very low frequency, LF: low frequency, HF: high frequency, PPG: photoplethysmography, LVET: left ventricular ejection time, SI: stiffness index, PAT: pulse arrival time, RI: reflection index ROC: receiver operating characteristic, GA: genetic algorithm, RRI: R-R interval, SBP: systolic blood pressure, DBP: diastolic blood pressure, MBP: mean blood pressure, TPR: total peripheral resistance CO: cardiac output, SV: stroke volume, SVR: support vector regression, LR: logistic regression, KNN: k-nearest neighbour, RF: random forest, SFS: sequential forward selection, Amps: point on the BP, dPdt_max: point on the dP/dt signal, PTT: pulse transit time, KSVM: kernel support vector machine, DFA: detrended fluctuation analysis, SampEn: sample entropy, BRS: baroreflex sensitivity, PP: pulse blood pressure, SBPV, systolic blood pressure variability, DBPV, diastolic blood pressure variability; CV, coefficient of variance; ARV, average real variability; RMSRV, root mean square of real variability; SDRV, standard deviation of real variability; HFnu, normalized high frequency power; LFnu, normalized low frequency power.

2.4 Summary

Classification is one of supervised learning method that under machine learning. Data collection is to obtain the relevant data which used to train the algorithm. Feature extraction is aimed to reduce the complexity of the data by transforming the data into a low-dimensional data while feature selection is using a specific method to select the optimum feature that able to bring best output for the classification algorithm. SVM, logistic regression and kNN are the popular supervised learning which able to classify the data.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter will discuss the planned workplan and the methodology used to build the classification algorithm. Gantt chart will be included in this chapter as well and the processes such as data collection, feature extraction, feature selection and building a series of classification model will be mentioned.

3.2 Work Plan

According to Table 3.1, all the task listed were completed on time. The content of task 1 is to research and study the coding technique of classification by using Python, since the future work in Part 1 mentioned continue to study with Python. In order to have better understanding on Python language, self-study on feature selector and classifier were done by exploring their function and parameters.

The upcoming task is constructing the algorithm. A series of feature selector and classifier were developed in 6 weeks. After all the classifier's performance was reviewed by supervisor, amendment and improvement such as missing data and imbalance data management were done for increase the accuracy of classification. The work is then continue with the poster preparation and report writing.

Table 3.1: Gantt Chart of FYP 1

Task no.	Task Description	Progress	Duration (days)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1	Problem Formulation and project planning	100%	14														
2	Literature review (systematic review)	100%	35														
3	Data collection and understanding	100%	14														
4	Construction of classification algorithm	100%	35														
5	Preliminary testing/ Evaluation	100%	21														
6	Report writing & presentation	100%	14														

Footnote: Duration of Gantt Chart start from 13/06/2022 which is Monday of W1 and end at 16/09/2022, Friday of W14

Table 3.2: Gantt Chart of FYP 2

Task no.	Task Description	Progress	Duration (days)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1	Research on Python classification algorithm	100%	21														
2	Construction of feature selection and classification	100%	42														
3	Amendment and improvement on classifier	100%	21														
4	Preparation of FYP poster	100%	7														
5	Report writing & presentation preparation	100%	14														
6	Report submission and presentation	100%	7														

Footnote: Duration of Gantt Chart start from 30/01/2023 which is Monday of W1 and end at 05/05/2023, Friday of W

3.3 Data Extraction and Data Study

All of the data were conducted at University of Malaya Medical Centre (UMMC) by HUTT. Each participated subject's consent was obtained before the test. UTAR Scientific and Ethical Review Committee (U/SERC/218/2020) and UMMC Medical Research Ethics Committee (MREC ID NO: 2020913-9066) has approved the ethical of test. A total 144 subjects were participated for this study, 56 participants were syncope positive, and 88 participants were syncope negative. A continuous non-invasive monitoring machine (Task Force Monitor, CNSystem, Austria) was used for collecting the hemodynamic measurement, which is the continuous physiological signals (beat-to-beat BP, ECG). The condition of surrounding of the test is quiet and temperature-controlled environment. Subject need to stay at supine position for first 10 minutes and are tilted to 70 degrees for next 20 minutes. 800 micrograms of GTN were injected to patient immediately once the subject is titled as a pharmacologic provocation.

After obtaining the data from supervisor, data study was carried out to understand the parameter that collected under beat-to-beat measurement such as beat, cardiac index (CI), cardiac output (CO), heart rate (HR), RR-interval (RRI), stiffness index (SI), stroke volume (SV), oxygen saturation (SpO2), total peripheral resistance (TPR), total peripheral resistance index (TPRI), tine of each test's section, diastolic blood pressure (dBP), mean blood pressure (mBP), systolic blood pressure (sBP), as shown in Figure 3.1. The purpose of data study is to ensure the process of feature extraction to carry out with more smoothly which able to prevent extracting irrelevant features and causing low accuracy of the classification.

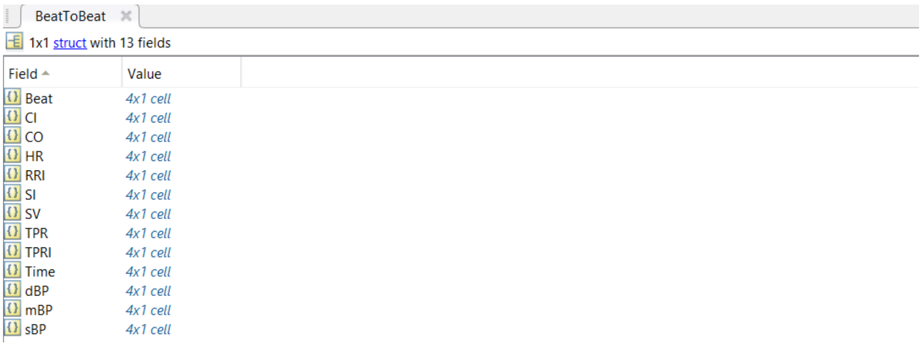

Figure 3.1: Parameters under beat-to-beat measurement.

3.4 Feature Extraction

After data study was done, some important features are extracted from the data set by using MATLAB. The mean, SD, CV, ARV, RMSRV, SDRV and mean of LF/HFof HR, sBP and dBP were extracted from the raw data obtained. By using the formulas below, those features able to be extracted:

Standard Deviation (SD) (Galie et al., 2009):

$$= \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{3.1}$$

Coefficient of Variance (CV) (Alpert, 2019):

$$= \frac{SD}{mean}\ x\ 100\% \tag{3.2}$$

Average Real Variability (ARV) (Miranda and Silva, 2016):

$$= \frac{\sum_{i=1}^{n-1} D_i}{n-1};\quad where\ D_{i=|x_{i+1}-x_i|} \tag{3.3}$$

Root Mean Square of Real Variability (RMSRV) (Adkisson and Benditt, 2017):

$$= \sqrt{\frac{\sum_{i=1}^{n-1}(D_i)^2}{n-1}} \tag{3.5}$$

Standard Deviation of Real Variability (SDRV) (He et al., 2021):

$$= \sqrt{\frac{\sum_{i=1}^{n-1}(D_i - \bar{D})^2}{n-1}} \tag{3.6}$$

where the x represents the beat-to-beat heart rate (HR), diastolic blood pressure (DBP) and systolic blood pressure (SBP), $\bar{x}$ is the mean of corresponding parameter and n is the total number of beats of the chosen parameter.

Based on Figure 3.1, there are 13 parameters in the beat-to-beat measurement. HR, sBP and dBP were selected for the feature extraction. Inside each parameter there will be few sets of the data and those data are corresponding to the different section of the HUTT, which are ‘Start Measurement’, ‘Start Recording’, ‘Tilt’, ‘GTN 2X’, ‘GTN’, ‘TTT’, ‘Front Load’, ‘End’ and ‘Stop Recording’. Only data that under section ‘Start Recording’, ‘Tilt’, ‘GTN 2X’, ‘GTN’, ‘TTT’ and ‘Front Load’ are useful for the feature extraction. ‘Start Recording’ section represent the data collected on the supine position while ‘Tilt’, ‘GTN 2X’, ‘GTN’, ‘TTT’ and ‘Front Load’ are representing the data during tilting process. The priority of the parameter selection for tilting process is ‘GTN 2X’, ‘GTN’, ‘Tilt’, ‘TTT’ and ‘Front Load’. After the features has successfully extracted, it was saved in CSV format for imputation purpose and the remaining process were done by using Python.

3.5 Imputation

In order to manage the missing data value, mean imputation was done after feature extracted. By finding the mean of each set of features that has been extracted, all the missing values able to replace with the mean values. ‘mean’ function was used in Python for calculating all the mean value of each set of features. Imputation was continued with ‘fillna’ function to replace the ‘Not A Number’(‘NaN’) with respectively mean value.

3.6 Feature Selection

Four techniques of feature selection have been done in this study which are sequential backward selection (SBS), sequential forward selection (SFS), recursive feature elimination (RFE) and genetic algorithm (GA). The process of develop these 4 techniques are mainly same where the only difference is the function and parameter used in each technique are different.

In order to build SBS and SFS model, a same function ‘SequentialFeatureSelector’ was imported from ‘sklearn.feature_selection’ module. The direction parameter was used to define whether forward or backward selection will be carried out, direction= ‘forward’ represented forward feature selection while direction= ‘backward’ indicated backward feature selection. ‘RFE’ function was imported from ‘sklearn.feature_selection’ while ‘GeneticSelectionCV’ was imported from ‘genetic_selection’ library for developing RFE and GA feature selection algorithm.

Estimator is one the parameter that need to be set, where it was the method that used to train the model for feature. In each of the estimator, random state was fix to 42, where it was used to drive the random number generator for shuffling and splitting data. When the random state number was fix, it able to ensure same pattern of shuffling no matter how many times the algorithm run, for ensure the consistency and reproductive of algorithm. Two estimators such as logistic regression and random forest were integrated with SFS and SBS, random forest, SVM and decision tree were integrated with RFE and GA integrated with decision tree.

3.7 Synthetic Minority Over-sampling Technique

As the data is imbalance, SMOTE is needed for preventing the result become bias. In order to create the synthetic instance, it is created by interpolation between several minority class instances that around defined neighbourhood. Based on Figure 3.2, after selected the minority class instance, x_i as a base, the nearest neighbourhood of same class which is point x_{i1} to x_{i4} chose according to a distance metric. At last, a randomized interpolation was done to obtain new instances r_1 to r_4 (Fernández et al., 2018).

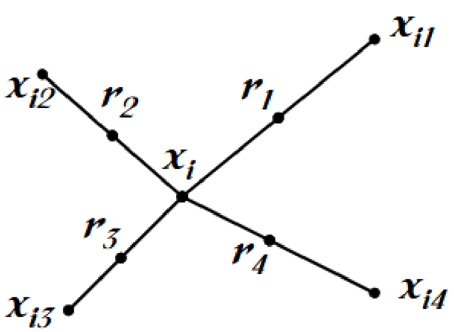

Figure 3.2: Illustration on how to create synthetic data in SMOTE (Fernández et al., 2018).

‘SMOTE’ function was imported from ‘imblearn.over_sampling’ library. By fixing the random state of SMOTE into 42, the input dataset, each single set of selected features, was fit into 'SMOTE' function to resample the data set.

3.8 Classification

After SMOTE was done, the process was continued to develop the classification algorithm. Classifiers such as random forest, decision tree and logistic regression were chosen in this study. Before proceeding to classification, train-test splitting was done. 'train_test_split' function was imported from 'sklearn.model_selection' library. Random state of splitting was set to 42 and test size was set to 0.20, to achieve an 80:20, where 80% of the input data set was used to train the model and the remaining 20% used to test the model.

Grid SearchCV was performed after the data splitting. Reason of implement Grid SearchCV is to fine tune the hyperparameters of every classifier, in order to obtain the best value and combination of parameter. 'GridSearchCV' function was imported from same library, 'sklearn.model_selection'. Since there are three types of classifiers selected, the parameter of each classifier also different.

Table 3.3: Value of Parameters in Each Grid SearchCV

Classifier	**Parameters**	**Values**
Random Forest	max_depth (maximum depth of the tree)	3,5,20
	n_estimator (number of the tree)	10,100,200
	max_features (number of features consider for best split)	2,3,5
	min_sample_leaf (minimum number of sample to be at leaf node)	1,2,3
Decision Tree	criterion (function to measure the quality of a split)	gini, entropy
	max_depth (maximum depth of the tree)	2,4,6,8,10
	min_samples_split (minimum number of samples to split an internal node)	2,4,6,8,10
	min_samples_leaf (minimum number of samples to be at a leaf node)	1,2,3,4,5
	max_features (number of features consider for best split)	sqrt, log2
	C (inverse of regularization strength)	0.01,0.1,1,10,100

Logistic Regression	penalty (norm)	l1, l2, elasticnet
	solver (algorithm to use in optimization problem)	liblinear, saga, lbfgs
	tol (tolerance for stopping criteria)	0.001, 0.0001, 0.00001

After Grid SearchCV was done, the best hyperparameter for each classification model was then fit into respective classifier with train set data to train the classification model and the process continue with using test set data for evaluating the trained classification model. Function for all three types of classifier were imported from sklearn library, where ‘DecisionTreeClassifier’ imported from ‘sklearn.tree’, ‘LogisticRegression’ imported from ‘sklearn.linear_model’ and ‘RandomForestClassifier’ imported from ‘sklearn.ensemble’. A 5-fold stratified cross-validation was applied to all the model for ensuring the generalizability of the model by importing ‘StratifiedKFold’ and ‘cross_validate’ function from ‘sklearn.model_selection’ library.

In order to evaluate the performance of classification model, true positive (TP), false positive (FP), true negative (TN) and false negative (FN) of the classification were calculated. Confusion matrix were then proceeded to calculate the performance metrics such as specificity, sensitivity (recall), F1-score and accuracy, according to the formula below:

Sensitivity (recall):

$$\frac{TP}{(TP+FN)} \tag{3.7}$$

Specificity:

$$\frac{TN}{(TN+FP)} \tag{3.8}$$

F1-score:

$$\frac{TP}{TP+\frac{1}{2}(FP+FN)} \tag{3.9}$$

Accuracy:

$$\frac{(TP+FN)}{(TP+TN+FP+FN)} \tag{3.10}$$

3.9 Summary

In summary, the task listed in Gantt chart able to be finish on time. In order to construct classification algorithm, data was obtained from the UMMC by HUT test and carried out data study. Important features such as mean, SD, CV, ARV, RMSRV and SDRV of HR and BP are extracted. After mean imputation was done, the process continues with the feature selection by using SFS, SBS, RFE and GA to select important feature. SMOTE was done to manage the imbalance data issue and the selected feature continued as input data for random forest, decision tree and logistic regression classifier.

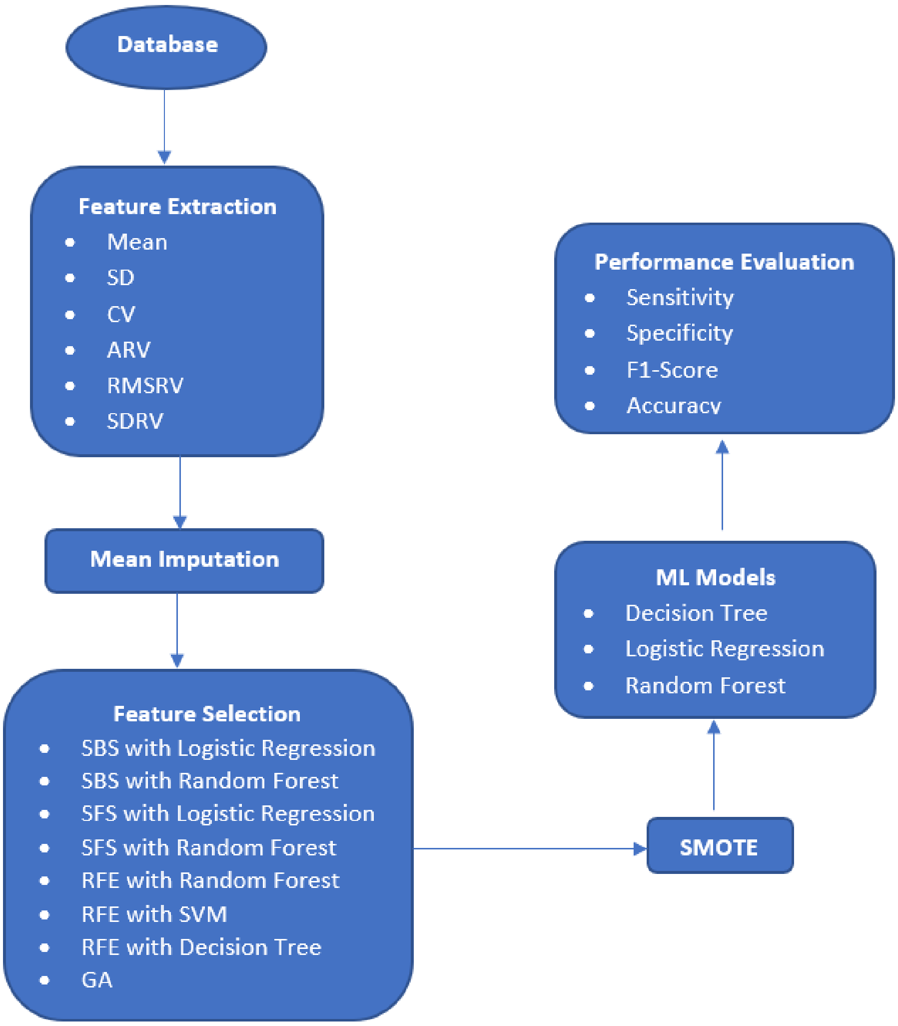

Figure 3.3: Flowchart of Whole Algorithm Construction

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter included the result of feature extraction, feature selection and the performance of classification.

4.2 Outputs from Feature Extraction and Feature Selection Algorithms

There are 144 subjects involved in this classification model which 56 syncope positive and 88 syncope negative. Total of 42 features has extracted from ECG and BP signal in position of supine and 70 degrees of tilting through time domain and frequency domain. Total of 8 feature selection model constructed with different techniques and different type of estimator.

According to Table 4.1, all of the feature selection model generated 5 best features excepted GA, generated 3 best features among 42 features as the parameter that determined the number of selected features for GA is different with others selector where GA required the maximum number of features selected while other selectors required number of features selected. SBS, SFS, RFE and GA are the techniques that used to select the feature while logistic regression, random forest, decision tree and SVM are the method that used to train the model for feature selection.

Table 4.1: Result of Feature Selection

SBS with Logistic Regression	CV_SBP_SP Mean_DBP_LF_HF_SP SDRV_DBP_T SDRV_HR_T SDRV_SBP_T
SBS with Random Forest	ARV_DBP_T Mean_SBP_LF_HF_T Mean_SBP_T SD_SBP_T SDRV_DBP_T
SFS with Logistic Regression	Mean_DBP_SP Mean_SBP_LF_HF_T Mean_SBP_SP Mean_SBP_T SDRV_DBP_T
SFS with Random Forest	ARV_HR_T ARV_SBP_T

	Mean_DBP_LF_HF_SP Mean_SBP_LF_HF_T SDRV_DBP_T
RFE with Random Forest	Mean_SBP_LF_HF_T Mean_SBP_SP Mean_SBP_T SD_SBP_T SDRV_DBP_T
RFE with SVM	CV_SBP_SP Mean_DBP_LF_HF_SP Mean_SBP_SP Mean_SBP_T SDRV_HR_SP
RFE with Decision Tree	ARV_SBP_SP Mean_SBP_LF_HF_T SD_SBP_T SDRV_DBP_T SDRV_SBP_SP
GA	Mean_SBP_LF_HF_T SDRV_DBP_SP SDRV_SBP_T

Note: CV: coefficient of variance, SBP: systolic blood pressure, DBP: diastolic blood pressure, LF_HF: low frequency/high frequency, SDRV: standard deviation of real variability, HR: heart rate, ARV: average of real variability, SD: standard deviation, T: tilting, SP: supine

4.3 Outputs and Performance of Classifiers

Decision tree, logistic regression and random forest were selected as the classifier for this study and all 8 set of selected features were underwent each classification techniques, resulted total 24 classification model.

4.3.1 Evaluation of Train-Test Split

Table 4.2 shows the performance metric such as sensitivity, specificity and accuracy of all the classification model by using test data. Random forest with GA as feature selector achieved the highest performance, 94.44% of sensitivity, 100% of specificity, 97.14% of F1 score and 97.22 % of accuracy. The hyperparameters that used to build this model were max depth=20, max features=2, min samples leaf=2 and n estimator= 10. The sensitivity ranges from 66.67% to 94.44%, specificity ranged from 72.22% to 100%, F1-score ranged from 68.57% to 97.14% and accuracy of classifier after the train-test-split ranges from 69.44 to 97.22%, where the lowest performance classifier is decision tree where SBS with random forest as the feature selection.

Table 4.2: Performance of Classifier after Train-Test Split in term of %

Classifiers / **Feature Selection**	**Decision Tree**	**Random Forest**	**Logistic Regression**
SBS with Logistic Regression	Sensitivity:88.89 Specificity:83.33 F1-score: 86.49 Accuracy: 86.11	Sensitivity:94.44 Specificity: 77.78 F1-score: 84.21 Accuracy:86.11	Sensitivity:88.89 Specificity:77.78 F1-score: 87.18 Accuracy: 83.33
SBS with Random Forest	Sensitivity:66.67 Specificity:72.22 F1-score: 68.57 Accuracy: 69.44	Sensitivity: 88.89 Specificity: 77.78 F1-score: 91.89 Accuracy: 83.33	Sensitivity:94.44 Specificity:88.89 F1-score: 84.21 Accuracy: 91.67
SFS with Logistic Regression	Sensitivity:83.33 Specificity:94.44 F1-score:88.24 Accuracy:88.89	Sensitivity: 94.44 Specificity: 88.89 F1-score:91.89 Accuracy: 91.67	Sensitivity:94.44 Specificity:88.89 F1-score: 91.89 Accuracy: 91.67
SFS with Random Forest	Sensitivity: 83.33 Specificity:88.89 F1-score:85.71 Accuracy:86.11	Sensitivity:94.44 Specificity:88.89 F1-score: 94.44 Accuracy: 91.67	Sensitivity:94.44 Specificity:94.44 F1-score: 91.89 Accuracy: 94.44

RFE with Random Forest	Sensitivity:88.89 Specificity:88.89 F1-score:88.89 Accuracy:88.89	Sensitivity:94.44 Specificity:88.89 F1-score:88.89 Accuracy: 91.67	Sensitivity:88.89 Specificity:88.89 F1-score: 91.89 Accuracy: 88.89
RFE with SVM	Sensitivity:77.78 Specificity:61.11 F1-score: 71.79 Accuracy:69.44	Sensitivity:94.44 Specificity:83.33 F1-score: 86.49 Accuracy: 88.89	Sensitivity:88.89 Specificity:83.33 F1-score: 89.47 Accuracy: 86.11
RFE with Decision Tree	Sensitivity:94.44 Specificity:83.33 F1-score: 89.47 Accuracy:88.89	Sensitivity:94.44 Specificity:94.44 F1-score: 94.44 Accuracy:94.44	Sensitivity:94.44 Specificity:94.44 F1-score: 94.44 Accuracy: 94.44
GA	Sensitivity:94.44 Specificity:77.78 F1-score:87.18 Accuracy:86.11	Sensitivity:94.44 Specificity:100.00 F1-score:97.14 Accuracy: 97.22	Sensitivity:94.44 Specificity:83.33 F1-score:89.47 Accuracy: 88.89

4.3.2 Evaluation after Cross Validation

A stratified fold cross validation was carried out in all the classifier to evaluate the performance and generalizability of ML model on unseen or new data with lower bias. Evaluation of train-test split is only able to assess the model's performance over training data where didn't focusing on evaluation of new data (Hussain et al., 2021). Evaluation of train-test split didn't provide any significant predictive accuracy due to bias where the bias might be due to the clustering of data point, one of the cluster stuck at training set while another stuck at test set (Gunasegaran and Cheah, 2017). Train dataset was used to carry out stratified k-fold cross validation. Reason of choosing stratified k-fold cross validation is this technique able to return each fold contains approximately same percentage of sample of each target class as the complete set.

Since it is 5-fold cross validation, the performance metric of each fold is different, minimal value, maximal value, mean and standard deviation of each performance metric for each set of classifiers has been record in Table 4.3 to 4.5. The minimum value represented the minimum value of the performance metric while maximum value indicated the highest performance metric value among the cross validation. Standard deviation of the performance metric indicated how well the data cluster or dispersed in the relation to mean. According to Table 4.6, the range of mean accuracy from 77.86% to 88.57%. Mean sensitivity ranged from 71.43% to 87.14%, mean specificity ranged from 74.29% to 94.29% and mean F1-score ranged from 75.30% to 88.45%, according to Table 4.3 and 4.5 respectively.

According to Table 4.3, the highest mean sensitivity is 87.14%, where the models are combination of random forest and SFS with random forest, RFE with decision tree and GA. The performance metric of random forest combined SFS with random forest is 90% of mean specificity, 88.45% of mean F1-score and 88.57% mean accuracy; random forest combined RFE with decision tree is 85.71% of mean specificity, 86.70% of mean F1-score and 86.43% of mean accuracy; GA is 82.86% of mean specificity, 85.24% of mean F1-score and 85% of mean accuracy. The lowest mean sensitivity is 71.43%, which is the combination of decision tree and SBS with logistic regression.

From Table 4.3, the minimal sensitivity of combination of random forest and SFS with random forest is 78.57% and the maximal sensitivity is 100% with a 0.07 standard deviation; combination of random forest and RFE with decision tree has a 71.43% minimal sensitivity, 92.86% of maximal sensitivity and 0.07 standard deviation; combination of random forest and GA achieved 71.43% of the minimal sensitivity and 100% of maximal sensitivity with a 0.1143 standard deviation. The highest minimal sensitivity is 78.57%, where there are 5 combinations achieved, combination of decision tree and SFS with logistic regression, RFE with decision tree, combination of random forest and SBS with logistic regression, SFS with random forest and RFE with decision tree. Total 8 combinations achieved 100% of the maximal sensitivity, where there are combination of decision tree and SFS with logistic regression, RFE with random forest, GA, combination of logistic regression and RFE with SVM, GA, combination of random forest and SFS with logistic regression, SFS with random forest, RFE with decision tree and GA.

However, the standard deviation of sensitivity for each combination for cross validation is quite high, compared to accuracy, as the standard deviation ranged from 0.07 to 0.1629, where it indicated the sensitivity value is more diverse to the mean value and less consistent. The standard deviation of the highest mean sensitivity is 0.07 except combination of random forest and GA is 0.1143.

Table 4.3: Sensitivity of the Classifier after Stratified Cross Validation in term of % (k-fold=5)

Classifier	Feature selector	Min	Max	Mean	SD
Decision Tree	SBS with Logistic Regression	42.86	85.71	71.43	0.1629
	SBS with Random Forest	64.29	92.86	75.71	0.0969
	SFS with Logistic Regression	78.57	100	85.71	0.0783
	SFS with Random Forest	71.43	92.86	81.43	0.0728
	RFE with Random Forest	64.29	100	81.43	0.1161
	RFE with SVM	64.29	85.71	78.57	0.0782
	RFE with Decision Tree	78.57	85.71	81.43	0.0350
	GA	57.14	100	82.86	0.1470
Logistic Regression	SBS with Logistic Regression	71.43	92.86	85.71	0.0782
	SBS with Random Forest	64.28	92.86	75.71	0.0969
	SFS with Logistic Regression	64.29	92.86	80.00	0.0948
	SFS with Random Forest	71.43	92.86	82.86	0.0728
	RFE with Random Forest	64.29	92.86	81.43	0.0969
	RFE with SVM	71.43	100	82.86	0.1161
	RFE with Decision Tree	71.43	92.86	82.86	0.0728
	GA	71.43	100	85.71	0.0903

Random Forest	SBS with Logistic Regression	78.57	92.86	84.29	0.0700
	SBS with Random Forest	64.29	92.86	77.14	0.0948
	SFS with Logistic Regression	71.43	100	85.71	0.0904
	SFS with Random Forest	78.57	100	87.14	0.0700
	RFE with Random Forest	71.43	92.86	84.29	0.0700
	RFE with SVM	57.14	92.86	80.00	0.1229
	RFE with Decision Tree	78.57	100	87.14	0.0700
	GA	71.43	100	87.14	0.1143

The highest mean specificity is 94.29%, combination of logistic regression and SFS with logistic regression, RFE with random forest. Combination of logistic regression and SFS with logistic regression achieved 80% of mean sensitivity, 85.92% of mean F1-score and 87.14% of mean accuracy while combination of logistic regression and RFE with random forest achieved 81.43% of mean sensitivity, 86.77% of mean F1-score and 87.86% of mean accuracy.

The minimal, maximal specificity and standard deviation of these two combinations is same, where 85.71% of minimal specificity and 100% of maximal specificity, with a 0.0535 standard deviation. 0.0535 standard deviation means the value of each specificity for each fold is closer to the mean specificity.

Table 4.4: Specificity of the Classifier after Stratified Cross Validation in term of % (k-fold=5)

Classifier	Feature selector	Min	Max	Mean	SD
Decision Tree	SBS with Logistic Regression	78.57	100	87.14	0.0833

	SBS with Random Forest	71.43	92.86	80.00	0.0833
	SFS with Logistic Regression	85.71	100	91.43	0.0700
	SFS with Random Forest	64.29	100	87.14	0.1229
	RFE with Random Forest	71.43	100	88.57	0.1069
	RFE with SVM	78.57	92.86	87.14	0.0536
	RFE with Decision Tree	85.71	92.86	88.57	0.0350
	GA	64.29	100	80.00	0.1457
Logistic Regression	SBS with Logistic Regression	71.43	92.86	82.86	0.0969
	SBS with Random Forest	57.14	100	74.29	0.1471
	SFS with Logistic Regression	85.71	100	94.29	0.0535
	SFS with Random Forest	78.57	100	81.43	0.0833
	RFE with Random Forest	85.71	100	94.29	0.0535
	RFE with SVM	78.57	100	91.43	0.0700
	RFE with Decision Tree	78.57	100	90.00	0.0969
	GA	64.29	100	84.29	0.1457
Random Forest	SBS with Logistic Regression	71.43	100	88.57	0.1161
	SBS with Random Forest	71.43	100	85.71	0.1010
	SFS with Logistic Regression	78.57	100	94.13	0.0833
	SFS with Random Forest	78.57	100	90.00	0.0857

	RFE with Random Forest	78.57	100	91.43	0.0833
	RFE with SVM	71.43	100	88.57	0.1161
	RFE with Decision Tree	71.43	100	85.71	0.1195
	GA	64.29	100	82.86	0.1325

F1-score is the harmonic mean of precision and sensitivity, which combined precision and sensitivity into a single metric. F1-score is an important metric for imbalanced data situation (Lipton et al., 2014). According to Table 4.5, combination of random forest and SFS with random forest achieved highest mean F1-score, 88.45%. The performance metric of random forest combined SFS with random forest is 87.14% mean sensitivity, 90% of mean specificity, 88.45% of mean F1-score and 88.57% mean accuracy.

Combination of random forest and SFS with random forest has a 78.57% of minimal F1-score and 93.33% of maximal F1-score with a 0.0563 of standard deviation. The range of the mean F1-score is considered as small where it only ranged from 75.30% to 88.45%.

Table 4.5: F1-score of the Classifier after Stratified Cross Validation in term of % (k-fold=5)

Classifier	Feature selector	Min	Max	Mean	SD
Decision Tree	SBS with Logistic Regression	52.17	92.31	76.75	0.1366
	SBS with Random Forest	71.43	83.87	77.17	0.0471
	SFS with Logistic Regression	81.48	93.33	88.18	0.0548
	SFS with Random Forest	68.97	89.66	84.02	0.0772
	RFE with Random Forest	78.26	93.33	84.22	0.0562
	RFE with SVM	72.00	88.89	81.96	0.0600
	RFE with Decision Tree	81.48	88.89	84.44	0.0279

	GA	64.00	88.89	81.40	0.0910
Logistic Regression	SBS with Logistic Regression	76.92	92.86	84.51	0.0578
	SBS with Random Forest	66.67	83.33	75.30	0.0721
	SFS with Logistic Regression	75.00	92.31	85.92	0.0600
	SFS with Random Forest	80.00	92.31	86.54	0.0397
	RFE with Random Forest	75.00	92.31	86.77	0.0606
	RFE with SVM	80.00	100.00	86.26	0.0735
	RFE with Decision Tree	80.00	92.31	85.95	0.0426
	GA	71.43	92.31	85.32	0.0763
Random Forest	SBS with Logistic Regression	75.86	96.30	86.32	0.0793
	SBS with Random Forest	75.00	86.67	80.47	0.0440
	SFS with Logistic Regression	74.07	96.55	88.19	0.0786
	SFS with Random Forest	78.57	93.33	88.45	0.0563
	RFE with Random Forest	80.00	92.31	87.40	0.0461
	RFE with SVM	69.57	92.31	83.34	0.0914
	RFE with Decision Tree	78.57	92.31	86.70	0.0612
	GA	76.92	92.31	85.24	0.0500

The highest mean accuracy model is 2 from random forest, SFS with logistic regression and SFS with random forest; 1 from decision tree where SFS with logistic regression, achieved 88.57%, the highest accuracy. Referring Table 4.3 to 4.5, the combination of random forest as classifier and SFS with logistic regression as feature selection achieved 85.71% mean sensitivity, 94.13%

mean specificity and 88.19% of mean F1-score; combination of random forest as classifier and SFS with random forest as feature selection achieved 87.14% mean sensitivity, 90.00% mean specificity and 88.45% mean F1-score; while the combination of decision tree as classifier and SFS with logistic regression achieved 85.71% mean sensitivity, 91.43% mean specificity and 88.18% of mean F1-score .

By further interpreting the performance of the highest accuracy model, combination of decision tree and SFS with logistic regression has the minimal accuracy of 82.14% and maximal accuracy of 92.86% with a 0.0525 standard deviation, combination of random forest and SFS with logistic regression has a 75% of minimal accuracy, but 96.43% of the maximal accuracy, 0.0763 of standard deviation. Combination of random forest and SFS with random forest achieved 78.57% of the minimal accuracy and 95.86% of maximal accuracy with a standard deviation of 0.0571. This showed that the highest mean accuracy not achieved highest minimal and maximal accuracy as the highest minimal accuracy is 82.14% and highest minimal accuracy is 100%. By studying the standard deviation in term of accuracy for each set of classifiers, the best performed model didn’t achieved lowest standard deviation, where it justified that the accuracy of each fold in cross validation didn’t cluster well or close enough to the mean value. The model with the lowest standard deviation is combination of decision tree and RFE with decision tree, where it achieved 0.0267, with 82.14% of minimal, 85.71% of maximal and 85% of mean accuracy.

As accuracy is calculated by TP and TN, divided by total population, it represents how well the model able to predict or classify the data correctly in term of TP and TN, acts as the most importance performance metric compared to other metric, which included the information of sensitivity and specificity. Thus, accuracy is the priority performance metric among other performance metrics. Since there are three combinations that achieved highest mean accuracy, by comparing the standard deviation, combination of decision tree and SFS with logistic regression is the best performance classification model as it has lowest standard deviation, where it indicated that the accuracy of each fold is closer to the mean accuracy. The chosen hyperparameter by Grid SearchCV were critetion=gini, max depth=2, max features=sqrt, min sample leaf=1 and min samples split=2.

Table 4.6: Accuracy of the Classifier after Stratified Cross Validation in term of % (k-fold=5)

Classifier	Feature selector	Min	Max	Mean	SD
Decision Tree	SBS with Logistic Regression	60.71	92.86	79.29	0.1069
	SBS with Random Forest	71.43	92.86	77.86	0.0416
	SFS with Logistic Regression	82.14	92.86	88.57	0.0525
	SFS with Random Forest	67.85	89.29	88.29	0.0833
	RFE with Random Forest	78.57	92.86	85.00	0.0525
	RFE with SVM	75.00	89.29	82.86	0.0525
	RFE with Decision Tree	82.14	85.71	85.00	0.0267
	GA	67.86	89.29	81.43	0.0795
Logistic Regression	SBS with Logistic Regression	78.57	92.86	84.29	0.0580
	SBS with Random Forest	64.29	85.71	75.00	0.0782
	SFS with Logistic Regression	78.57	92.86	87.14	0.0484
	SFS with Random Forest	82.14	92.86	87.14	0.0364
	RFE with Random Forest	78.57	92.86	87.86	0.0484
	RFE with SVM	82.14	100	87.14	0.0662
	RFE with Decision Tree	82.14	92.86	86.43	0.0416
	GA	71.43	92.86	85.00	0.0795

Random Forest	SBS with Logistic Regression	75.00	96.43	86.43	0.0827
	SBS with Random Forest	75.00	85.71	81.43	0.0417
	SFS with Logistic Regression	75.00	96.43	88.57	0.0763
	SFS with Random Forest	78.57	92.86	88.57	0.0571
	RFE with Random Forest	82.14	92.86	87.86	0.0429
	RFE with SVM	75.00	92.85	84.29	0.0802
	RFE with Decision Tree	78.57	92.86	86.43	0.0655
	GA	78.57	92.86	85.00	0.0474

Table 4.7 shows the confusion matrix of cross validation for combination of decision tree and SFS with logistic regression. By using the confusion matrix to calculate the performance metric, sensitivity is 85.71% specificity is 91.43% and accuracy is 88.57%, which matched with the performance metric that reported in Table 4.3, 4.4 and 4.6.

Table 4.7: Confusion Matrix of Cross Validation for Combination of Decision Tree and SFS with Logistic Regression

		Actual Value	
		True	**False**
Predicted Value	**True**	True Positive 60	False Positive 6
	False	False Negative 10	True Negative 64

4.4 Evaluation of Performance of Designed Algorithm with State-Of-Art Algorithm

By referring Table 4.8, performance of the proposed model is not the worst performance compared to state-of-the-art. The accuracy of proposed model has exceeded the accuracy of Khodor et al., Carmody et al and He et al, only the performance of Zhang et al. is better than the proposed model. In the study of Zhang et al., logistic regression has been utilised to carry out classification. ECG and BP signal was collected and extracted a series of features. By applying multivariate logistic regression with forward selection as feature selection technique, SBP, DBP and HR has been selected. The outcome of the classification model has 90.9% accuracy with 89.3% sensitivity and 80.0% specificity.

In order to complete a classification model, the main step is to carry out feature extraction, feature selection and train the classification model. The studies that included in Table 4.8 has undergone the main process of building a classification. However, throughout the whole process, they didn’t mention any missing data management, imbalance data management and fine tuning of the hyperparameter. In the absence of imbalance data management, the standard classifier tends to be overwhelmed by the majority classes and ignoring the minority class, causing a high overall in the model’s accuracy, and has poor performance on the minority class, which vital importance in medical diagnosis (Krawczyk, 2016, Chawla et al., 2002).

Fine tuning hyperparameter is importance process in a classification model as it able to select the best combination of hyperparameter and maximum the performance of model. By applying Grid SearchCV, it able to reduce the overfitting as the cross-validation is done during the grid search by ensuring the model does not overfit on the training data. Thus, fine tuning of hyperparameter able to increase the performance of model.

Thus, mean imputation for missing data management, SMOTE for imbalance data management and Grid SearchCV for fine tuning the hyperparameter has been done in this study and contributed to the high performance of proposed model.

Table 4.8: Comparison of Designed Algorithm with State-Of-Art Algorithm

Studies	**Classifier**	**Sensitivity, (%)**	**Specificity, (%)**	**Accuracy, (%)**
Proposed Model	Decision Tree	85.71	91.43	88.57
Khodor, N., et al. (2014)	KSVM	88.5	80.6	84.8
Carmody, M., et al. (2020)	Univariate classifier, multivariable classifier	84.3	72.9	80.9
Zhang, Z. N., et al. (2020)	Logisitic Regression	89.3	80.8	90.2
He, Z., et al. (2021)	SVR	86.0	82.0	84.2

In syncope classification model, feature that used to build the model is one of the factors that affecting the performance. As the best performance model is the combination of decision tree and SFS with logistic regression, combination of random forest and SFS with logistic regression also achieved highest accuracy, indicated that the features used to build the model will affect the performance of model. Features that used to build both classification models were mean of SBP in tilting and supine position, mean of DBP in supine position, SDRV of DBP in tilting position and mean of LF/HF of SBP in tilting position. According to Zhang et al. (2020), the feature that used to build the classification model were SBP, DBP and HR. He et al. (2021) and Carmody et al. (2020) also have selected SBP and DBP as one the main feature to build classification model, showed that SBP and DBP are important in syncope classification. As orthostatic hypotension was the primary cause of syncope, study of Atkins et al. (1991) showed that presence of orthostatic hypotension increase the risk of syncope, where the orthostatic blood pressure changes is abnormal (more than 20 mmHg). Orthostatic hypotension is situation which blood pressure suddenly drops when the person stands up from a seated or lying position, which considered as one of the symptom that can be classify as syncope (Thijs et al., 2004). In the review of Thijs et al, (2005), the author mentioned 69% of the neurocardiogenic syncope studies defined hypotension was one of the symptom of syncope and 58% of the neurally mediated syncope studies defined hypotension was the symptom of syncope. This showed that changes of blood pressure can considered as one of the majority or importance variable to classify the occurrence of syncope.

4.5 Future Trend of Machine Learning in Syncope

As classification syncope by using ML able to aid clinician in syncope diagnosis, it will be an addition resource or standard to be refer by clinician before decision making. With the advancement of AI and ML, the performance of syncope classification can be improved from time to time. ML algorithm will keep gaining and learning new data, which they able to learn from big data and different type of physiological signal that acquired from HUTT. However, process of collecting the suitable dataset requires high effort and time consuming, Mossello et al. (2018) has spent three and half years to collect related data from 372 subjects, which is inefficient. Hence, the advantage of open access medical research database such as PhysioNet, able to reduce the time taken for collecting data. However, dataset that obtained from open access database should handle carefully as the process of data collection for syncope is different. This challenge can be solved by standardized the method for data acquisition, which able to ensure the characteristic of data collected are same.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this study, total of 8 different techniques of feature selection has been implemented and 3 types of classifiers utilised. A syncope classification model that utilised sequential forward selection method with logistic regression as feature selection and decision tree as the classifier has achieved 85.71% sensitivity, 91.43% specificity, 88.18% F1-score and 88.57% accuracy. Comparing the proposed model with state-of-the-art algorithm, although proposed model is not the best model among the comparison, it able to achieve 88.57% accuracy which is higher than other studies.

5.2 Recommendations for Future Work

Since there are many different types of classification, future work can be proceeded on developing syncope classification model with other classifier such as SVM and kNN, to obtain the best performance model in a series of model. External cross validation can be part of future work as the process of collecting external data for cross validation is difficult and time consuming. Another future work is integrating proposed model into clinical application to evaluate the function and performance of it. With the application on clinical, advantage and disadvantage of the proposed model can be identified and improvement can be carried out.

REFERENCE

Adkisson, W. O. and Benditt, D. G. 2017. Pathophysiology of reflex syncope: A review. *Journal of Cardiovascular Electrophysiology,* 28**,** 1088-1097.

Ali, J., Khan, R., Ahmad, N. and Maqsood, I. 2012. Random Forests and Decision Trees. *International Journal of Computer Science Issues(IJCSI),* 9.

Alpert, J. S. 2019. Syncope in the elderly. *The American Journal of Medicine,* 132**,** 1115-1116.

Atkins, D., Hanusa, B., Sefcik, T. and Kapoor, W. 1991. Syncope and orthostatic hypotension. *The American Journal of Medicine,* 91**,** 179-185.

Babatunde, O. H., Armstrong, L. , Leng, J. , and Diepeveen, D. (2014). A Genetic Algorithm-Based Feature Selection. *International Journal of Electronics Communication and Computer Engineering*, 5(4), 899-905.

Baraldi, A. N. and Enders, C. K. 2010. An introduction to modern missing data analyses. *Journal of School Psychology,* 48**,** 5-37.

Bartoletti, A., Alboni, P., Ammirati, F., Brignole, M., Del Rosso, A., Foglia Manzillo, G., Menozzi, C., Raviele, A. and Sutton, R. 2000. 'The Italian Protocol': a simplified head-up tilt testing potentiated with oral nitroglycerin to assess patients with unexplained syncope. *EP Europace,* 2**,** 339-342.

Bisignani, A., De Bonis, S., Mancuso, L., Ceravolo, G. and Bisignani, G. 2019. Implantable loop recorder in clinical practice. *Journal of Arrhythmia,* 35**,** 25-32.

Brignole, M. 2007. Diagnosis and treatment of syncope. *Heart,* 93**,** 130-6.

Brignole, M., Alboni, P., Benditt, D., Bergfeldt, L., Blanc, J., Bloch Thomsen, P., Van Dijk, J. G., Fitzpatrick, A., Hohnloser, S. and Janousek, J. 2001. Guidelines on management (diagnosis and treatment) of syncope. *European Heart Journal,* 22**,** 1256-1306.

Brignole, M., Moya, A., De Lange, F. J., Deharo, J.-C., Elliott, P. M., Fanciulli, A., Fedorowski, A., Furlan, R., Kenny, R. A., Martín, A., Probst, V., Reed, M. J., Rice, C. P., Sutton, R., Ungar, A. and Van Dijk, J. G. 2018a. 2018 ESC Guidelines for the diagnosis and management of syncope. *European Heart Journal,* 39**,** 1883-1948.

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. 2012. DBSMOTE: density-based synthetic minority over-sampling technique. *Applied Intelligence,* 36**,** 664-684.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research,* 16**,** 321-357.

Chumerin, N. and Van Hulle, M. M. Comparison of two feature extraction methods based on maximization of mutual information. *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing,* 2006. IEEE, 343-348.

Couceiro, R., Carvalho, P., Paiva, R. P., Muehlsteff, J., Henriques, J., Eickholt, C., Brinkmeyer, C., Kelm, M. and Meyer, C. 2015. Real-time prediction of neurally mediated syncope. *IEEE Journal of Biomedical and Health Informatics,* 20**,** 508-520.

Couceiro, R., Carvalho, P., Paiva, R. P., Muehlsteff, J., Henriques, J., Eickholt, C., Brinkmeyer, C., Kelm, M. and Meyer, C. 2016. Real-Time Prediction of Neurally Mediated Syncope. *IEEE Journal of Biomedical and Health Informatics,* 20**,** 508-520.

Da Silva, R. M. 2014. Syncope: epidemiology, etiology, and prognosis. *Frontiers in Physiology,* 5**,** 471.

Donders, A. R. T., Van Der Heijden, G. J. M. G., Stijnen, T. and Moons, K. G. M. 2006. Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology,* 59**,** 1087-1091.

Ferdowsi, M., Goh, C.-H. and Kwan, B.-H. Classification of Syncope in Front-Loaded Head-Up Tilt Test with Support Vector Machine. *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA),* 2022. IEEE, 187-191.

Fernández, A., Garcia, S., Herrera, F. and Chawla, N. V. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research,* 61**,** 863-905.

Feuilloy, M., Schang, D. and Nicolas, P. Comparison of feature selection methods for syncope prediction. *2006 IEEE International Conference on Evolutionary Computation*, 2006. IEEE, 2756-2763.

Galie, N., Hoeper, M. M., Humbert, M., Torbicki, A., Vachiery, J.-L., Barbera, J. A., Beghetti, M., Corris, P., Gaine, S. and Gibbs, J. S. 2009. Guidelines for the diagnosis and treatment of pulmonary hypertension: the Task Force for the Diagnosis and Treatment of Pulmonary Hypertension of the European Society of Cardiology (ESC) and the European Respiratory Society (ERS), endorsed by the International Society of Heart and Lung Transplantation (ISHLT). *European Heart Journal,* 30**,** 2493-2537.

Gunasegaran, T. and Cheah, Y. N. Evolutionary cross validation. *2017 8th International Conference on Information Technology (ICIT),* 17-18 May 2017 2017. 89-95.

Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K. KNN model-based approach in classification. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2003. Springer, 986-996.

He, Z., Du, L., Du, S., Wu, B., Fan, Z., Xin, B., Chen, X., Fang, Z. and Liu, J. 2021. Machine learning for the early prediction of head-up tilt testing outcome. *Biomedical Signal Processing and Control,* 69**,** 102904.

Huang, M. L., Hung, Y. H., Lee, W. M., Li, R. K. and Jiang, B. R. 2014. SVM-RFE based feature selection and Taguchi parameters optimization for multiclass SVM classifier. *Scientific World Journal,* 2014**,** 795624.

Hussain, S., Raza, Z., Giacomini, G. and Goswami, N. 2021. Support vector machine-based classification of vasovagal syncope using head-up tilt test. *Biology,* 10**,** 1029.

Jadhav, S. D. and Channe, H. 2016. Comparative study of K-NN, naive Bayes and decision tree classification techniques. *International Journal of Science and Research (IJSR),* 5**,** 1842-1845.

Kenny, R. A. and Krahn, A. D. 1999. Implantable loop recorder: evaluation of unexplained syncope. *Heart,* 81**,** 431-433.

Khalid, S., Khalil, T. and Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. *2014 Science and Information Conference*, 2014. IEEE, 372-378.

Krawczyk, B. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence,* 5**,** 221-232.

Lipton, Z. C., Elkan, C. and Narayanaswamy, B. 2014. Thresholding classifiers to maximize F1 score. *arXiv preprint arXiv:1402.1892*.

Mccarthy, K., Ward, M., Ortuno, R. R. and Kenny, R. A. 2020. Syncope, Fear of Falling and Quality of Life Among Older Adults: Findings From the Irish Longitudinal Study on Aging (TILDA). *Frontiers in Cardiovascular Medicine,* 7**,** 12.

Miranda, C. M. and Da Silva, R. 2016. Analysis of Heart Rate Variability Before and During Tilt Test in Patients with Cardioinhibitory Vasovagal Syncope. *Arquivos Brasileiros de Cardiologia,* 107**,** 568-575.

Miranda, C. M. and Silva, R. M. F. L. D. 2016. Analysis of heart rate variability before and during tilt test in patients with cardioinhibitory vasovagal syncope. *Arquivos Brasileiros de Cardiologia,* 107**,** 568-575.
Mossello, E., Ceccofiglio, A., Rafanelli, M., Riccardi, A., Mussi, C., Bellelli, G., et al. 2018. Differential diagnosis of unexplained falls in dementia: Results of “Syncope & Dementia” registry. *European Journal of Internal Medicine,* 50, 41-46.

Nasteski, V. 2017. An overview of the supervised machine learning methods. *Horizons. b,* 4**,** 51-62.

Osisanwo, F., Akinsola, J., Awodele, O., Hinmikaiye, J., Olakanmi, O. and Akinjobi, J. 2017. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT),* 48**,** 128-138.

Pal, M. 2005. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing,* 26**,** 217-222.

Roh, Y., Heo, G. and Whang, S. E. 2019. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering,* 33**,** 1328-1347.

Runser, L. A., Gauer, R. L. and Houser, A. 2017. Syncope: Evaluation and Differential Diagnosis. *American Family Physician,* 95**,** 303-312.

Rustam, Z. and Kharis, S. Comparison of support vector machine recursive feature elimination and kernel function as feature selection using support vector machine for lung cancer classification. *Journal of Physics: Conference Series*, 2020. IOP Publishing, 012027.

Sanz, H., Valim, C., Vegas, E., Oller, J. M. and Reverter, F. 2018. SVM-RFE: selection and visualization of the most relevant features through non-linear kernels. *BMC Bioinformatics,* 19**,** 432.

Saravanan, R. and Sujatha, P. A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. *2018 Second International Conference on Intelligent Computing and Control Systems*, 2018. IEEE, 945-949.

Scheffer, J. 2002. Dealing with missing data. *Research Letters in the Information and Mathematical Sciences,* 3**,** 153-160.

Sen, P. C., Hajra, M. and Ghosh, M. 2020. Supervised classification algorithms in machine learning: A survey and review. *Emerging Technology in Modelling and Graphics.* Springer.

Shen, W. K., Sheldon, R. S., Benditt, D. G., Cohen, M. I., Forman, D. E., Goldberger, Z. D, et al. 2017 ACC/AHA/HRS Guideline for the Evaluation and Management of Patients With Syncope: Executive Summary A Report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines and the Heart Rhythm Society. *Journal of the American College of Cardiology*, 70**,** 620-663.

Singh, A., Thakur, N. and Sharma, A. A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom),* 2016. IEEE, 1310-1315.

Song, Y.-Y. and Ying, L. 2015. Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry,* 27**,** 130.

Sutton, R. 2013. Clinical classification of syncope. *Progress in Cardiovascular Diseases,* 55**,** 339-344.

Thijs, R. D., Wieling, W., Kaufmann, H. and Van Dijk, G. 2004. Defining and classifying syncope. *Clinical Autonomic Research,* 14**,** i4-i8.

Ungar, A., Sgobino, P., Russo, V., Vitale, E., Sutton, R., Melissano, D., Beiras, X., Bottoni, N., Ebert, H. H., Gulizia, M., Jorfida, M., Moya, A., Andresen, D., Grovale, N., Brignole, M. and Int Study Syncope Uncertain, E. 2013. Diagnosis of neurally mediated syncope at initial evaluation and with tilt table testing compared with that revealed by prolonged ECG monitoring. An analysis from the Third International Study on Syncope of Uncertain Etiology (ISSUE-3). *Heart,* 99**,** 1825-1831.

Vergara, J. R. and Estévez, P. A. 2014. A review of feature selection methods based on mutual information. *Neural computing and applications,* 24**,** 175-186.
Walsh, K., Hoffmayer, K. & Hamdan, M. H. 2015. Syncope: Diagnosis and Management. *Current Problems in Cardiology,* 40**,** 51-86.

Wong, C. W. 2018. Complexity of syncope in elderly people: a comprehensive geriatric approach. *Hong Kong Medical Journal,* 24**,** 182-190.

APPENDIX

Appendix A: Code

Feature Extraction

```
promt = 'How many subject? : ';
sub_num = input(promt);

valuesOfMean_HR_SP= zeros(1,sub_num);
valuesOfSD_HR_SP= zeros(1,sub_num);
valuesOfCV_HR_SP= zeros(1,sub_num);
valuesOfARV_HR_SP= zeros(1,sub_num);
valuesOfRMSRV_HR_SP= zeros(1,sub_num);
valuesOfSDRV_HR_SP= zeros(1,sub_num);
valuesofMean_HRV_LF_HF_SP= zeros(1,sub_num);

valuesOfMean_SBP_SP= zeros(1,sub_num);
valuesOfSD_SBP_SP= zeros(1,sub_num);
valuesOfCV_SBP_SP= zeros(1,sub_num);
valuesOfARV_SBP_SP= zeros(1,sub_num);
valuesOfRMSRV_SBP_SP= zeros(1,sub_num);
valuesOfSDRV_SBP_SP= zeros(1,sub_num);
valuesOfMean_SBP_LF_HF_SP= zeros(1,sub_num);

valuesOfMean_DBP_SP= zeros(1,sub_num);
valuesOfSD_DBP_SP= zeros(1,sub_num);
valuesOfCV_DBP_SP= zeros(1,sub_num);
valuesOfARV_DBP_SP= zeros(1,sub_num);
valuesOfRMSRV_DBP_SP= zeros(1,sub_num);
valuesOfSDRV_DBP_SP= zeros(1,sub_num);
valuesOfMean_DBP_LF_HF_SP= zeros(1,sub_num);

valuesOfMean_HR_T= zeros(1,sub_num);
valuesOfSD_HR_T= zeros(1,sub_num);
valuesOfCV_HR_T= zeros(1,sub_num);
valuesOfARV_HR_T= zeros(1,sub_num);
valuesOfRMSRV_HR_T= zeros(1,sub_num);
valuesOfSDRV_HR_T= zeros(1,sub_num);
valuesofMean_HRV_LF_HF_T= zeros(1,sub_num);

valuesOfMean_SBP_T= zeros(1,sub_num);
valuesOfSD_SBP_T= zeros(1,sub_num);
valuesOfCV_SBP_T= zeros(1,sub_num);
valuesOfARV_SBP_T= zeros(1,sub_num);
valuesOfRMSRV_SBP_T= zeros(1,sub_num);
valuesOfSDRV_SBP_T= zeros(1,sub_num);
valuesOfMean_SBP_LF_HF_T= zeros(1,sub_num);

valuesOfMean_DBP_T= zeros(1,sub_num);
```

```
valuesOfSD_DBP_T= zeros(1,sub_num);
valuesOfCV_DBP_T= zeros(1,sub_num);
valuesOfARV_DBP_T= zeros(1,sub_num);
valuesOfRMSRV_DBP_T= zeros(1,sub_num);
valuesOfSDRV_DBP_T= zeros(1,sub_num);
valuesOfMean_DBP_LF_HF_T= zeros(1,sub_num);

for num=001 : +1 : sub_num
   try
      load(sprintf('SR%d.mat',num));
      try
          Index_SP=find(contains(IV.Name,'Start Recording'));
          if sum(strncmpi('GTN',IV.Name,3))==1
             Index_T=find(contains(IV.Name,'GTN','IgnoreCase',true));
          elseif sum(strncmpi('TILT',IV.Name,4))==1
             Index_T=find(contains(IV.Name,'TILT','IgnoreCase',true));
          elseif sum(strncmpi('TTT',IV.Name,3))==1
             Index_T=find(contains(IV.Name,'TTT','IgnoreCase',true));
          elseif sum(strncmpi('FRONT LOAD',IV.Name,5))==1
             Index_T=find(contains(IV.Name,'FRONT
LOAD','IgnoreCase',true));
          elseif sum(strncmpi('FRONT LOAD',IV.Name,5))==2
            a=find(strncmpi('FRONT LOAD',IV.Name,5)==1);
            [row1,col1]=size(BeatToBeat.HR{a(1,1),1});
            [row2,col2]=size(BeatToBeat.HR{a(2,1),1});
            if max(col1,col2)==col1
               Index_T=a(1,1);
            else
               Index_T=a(2,1);
            end
         elseif sum(strncmpi('ACTIVE STAND',IV.Name,5))==1
            Index_T=find(contains(IV.Name,'ACTIVE
STAND','IgnoreCase',true));
         else
            Index_T=find(contains(IV.Name,'PASSIVE
STAND','IgnoreCase',true));
          end
         Total_Di_HR_SP=0;
         Total_Di_sqr_HR_SP=0;
         Total_Di_HR_T=0;
         Total_Di_sqr_HR_T=0;

         Total_Di_SBP_SP=0;
         Total_Di_sqr_SBP_SP=0;
         Total_Di_SBP_T=0;
         Total_Di_sqr_SBP_T=0;

         Total_Di_DBP_SP=0;
         Total_Di_sqr_DBP_SP=0;
         Total_Di_DBP_T=0;
```

```
Total_Di_sqr_DBP_T=0;

HR_SP_ORI=BeatToBeat.HR{Index_SP,1};
HR_SP_ORI=HR_SP_ORI(~isnan(HR_SP_ORI));
HR_SP=hr_correct(HR_SP_ORI);
HR_T_ORI=BeatToBeat.HR{Index_T,1};
HR_T_ORI=HR_T_ORI(~isnan(HR_T_ORI));
HR_T=hr_correct(HR_T_ORI);

HRV_LF_HF_SP=HRV.LF_HF{Index_SP,1};
HRV_LF_HF_SP=HRV_LF_HF_SP(~isnan(HRV_LF_HF_SP));
HRV_LF_HF_T=HRV.LF_HF{Index_T,1};
HRV_LF_HF_T=HRV_LF_HF_T(~isnan(HRV_LF_HF_T));

SBP_SP=BeatToBeat.sBP{Index_SP,1};
SBP_SP=SBP_SP(~isnan(SBP_SP));
SBP_T=BeatToBeat.sBP{Index_T,1};
SBP_T=SBP_T(~isnan(SBP_T));

SBP_LF_HF_SP=BPVsBP.LF_HF{Index_SP,1};
SBP_LF_HF_SP=SBP_LF_HF_SP(~isnan(SBP_LF_HF_SP));
SBP_LF_HF_T=BPVsBP.LF_HF{Index_T,1};
SBP_LF_HF_T=SBP_LF_HF_T(~isnan(SBP_LF_HF_T));

DBP_SP=BeatToBeat.dBP{Index_SP,1};
DBP_SP=DBP_SP(~isnan(DBP_SP));
DBP_T=BeatToBeat.dBP{Index_T,1};
DBP_T=DBP_T(~isnan(DBP_T));

DBP_LF_HF_SP=BPV.LF_HF{Index_SP,1};
DBP_LF_HF_SP=DBP_LF_HF_SP(~isnan(DBP_LF_HF_SP));
DBP_LF_HF_T=BPV.LF_HF{Index_T,1};
DBP_LF_HF_T=DBP_LF_HF_T(~isnan(DBP_LF_HF_T));

[numRows_HR_SP, numCols_HR_SP]=size(HR_SP);
valuesOfDi_HR_SP= zeros(1,numCols_HR_SP-1);
[numRows_HR_T, numCols_HR_T]=size(HR_T);
valuesOfDi_HR_T= zeros(1,numCols_HR_T-1);

[numRows_SBP_SP, numCols_SBP_SP]=size(SBP_SP);
valuesOfDi_SBP_SP= zeros(1,numCols_SBP_SP-1);
[numRows_SBP_T, numCols_SBP_T]=size(SBP_T);
valuesOfDi_SBP_T= zeros(1,numCols_SBP_T-1);

[numRows_DBP_SP, numCols_DBP_SP]=size(DBP_SP);
valuesOfDi_DBP_SP= zeros(1,numCols_DBP_SP-1);
[numRows_DBP_T, numCols_DBP_T]=size(DBP_T);
valuesOfDi_DBP_T= zeros(1,numCols_DBP_T-1);

Mean_HR_SP= mean(HR_SP);
```

```
SD_HR_SP=std(HR_SP,1);
CV_HR_SP=SD_HR_SP/mean(HR_SP)*100;
Mean_HRV_LF_HF_SP=mean(HRV_LF_HF_SP);
Mean_HR_T= mean(HR_T);
SD_HR_T=std(HR_T,1);
CV_HR_T=SD_HR_T/mean(HR_T)*100;
Mean_HRV_LF_HF_T=mean(HRV_LF_HF_T);

Mean_SBP_SP= mean(SBP_SP);
SD_SBP_SP=std(SBP_SP,1);
CV_SBP_SP=SD_SBP_SP/mean(SBP_SP)*100;
Mean_SBP_LF_HF_SP= mean(SBP_LF_HF_SP);
Mean_SBP_T= mean(SBP_T);
SD_SBP_T=std(SBP_T,1);
CV_SBP_T=SD_SBP_T/mean(SBP_T)*100;
Mean_SBP_LF_HF_T= mean(SBP_LF_HF_T);

Mean_DBP_SP= mean(DBP_SP);
SD_DBP_SP=std(DBP_SP,1);
CV_DBP_SP=SD_DBP_SP/mean(DBP_SP)*100;
Mean_DBP_LF_HF_SP= mean(DBP_LF_HF_SP);
Mean_DBP_T= mean(DBP_T);
SD_DBP_T=std(DBP_T,1);
CV_DBP_T=SD_DBP_T/mean(DBP_T)*100;
Mean_DBP_LF_HF_T= mean(DBP_LF_HF_T);

for x=1: +1:numCols_HR_SP-1
   Di_HR_SP= abs(HR_SP(1,x+1)-HR_SP(1,x));
   valuesOfDi_HR_SP(x)=Di_HR_SP;
   Di_HR_SP_sqr=Di_HR_SP^2;
   Total_Di_HR_SP= Di_HR_SP+Total_Di_HR_SP;
   Total_Di_sqr_HR_SP= Di_HR_SP_sqr+Total_Di_sqr_HR_SP;
end

for x=1: +1:numCols_HR_T-1
      Di_HR_T= abs(HR_T(1,x+1)-HR_T(1,x));
      valuesOfDi_HR_T(x)=Di_HR_T;
      Di_HR_T_sqr=Di_HR_T^2;
      Total_Di_HR_T= Di_HR_T+Total_Di_HR_T;
      Total_Di_sqr_HR_T= Di_HR_T_sqr+Total_Di_sqr_HR_T;
end

for y=1: +1: numCols_SBP_SP-1
   Di_SBP_SP= abs(SBP_SP(1,y+1)-SBP_SP(1,y));
   valuesOfDi_SBP_SP(y)=Di_SBP_SP;
   Di_SBP_SP_sqr=Di_SBP_SP^2;
   Total_Di_SBP_SP= Di_SBP_SP+Total_Di_SBP_SP;
   Total_Di_sqr_SBP_SP= Di_SBP_SP_sqr+Total_Di_sqr_SBP_SP;
end
```

```
for y=1: +1: numCols_SBP_T-1
   Di_SBP_T= abs(SBP_T(1,y+1)-SBP_T(1,y));
   valuesOfDi_SBP_T(y)=Di_SBP_T;
   Di_SBP_T_sqr=Di_SBP_T^2;
   Total_Di_SBP_T= Di_SBP_T+Total_Di_SBP_T;
   Total_Di_sqr_SBP_T= Di_SBP_T_sqr+Total_Di_sqr_SBP_T;
end

for Z=1: +1: numCols_DBP_SP-1
   Di_DBP_SP= abs(DBP_SP(1,Z+1)-DBP_SP(1,Z));
   valuesOfDi_DBP_SP(Z)=Di_DBP_SP;
   Di_DBP_SP_sqr=Di_DBP_SP^2;
   Total_Di_DBP_SP= Di_DBP_SP+Total_Di_DBP_SP;
   Total_Di_sqr_DBP_SP= Di_DBP_SP_sqr+Total_Di_sqr_DBP_SP;
end

for Z=1: +1: numCols_DBP_T-1
      Di_DBP_T= abs(DBP_T(1,Z+1)-DBP_T(1,Z));
      valuesOfDi_DBP_T(Z)=Di_DBP_T;
      Di_DBP_T_sqr=Di_DBP_T^2;
      Total_Di_DBP_T= Di_DBP_T+Total_Di_DBP_T;
      Total_Di_sqr_DBP_T= Di_DBP_T_sqr+Total_Di_sqr_DBP_T;
end

ARV_HR_SP= Total_Di_HR_SP/numCols_HR_SP-1;
RMSRV_HR_SP= Total_Di_sqr_HR_SP/numCols_HR_SP-1;
SDRV_HR_SP= std(valuesOfDi_HR_SP,0);

ARV_HR_T= Total_Di_HR_T/numCols_HR_T-1;
RMSRV_HR_T= Total_Di_sqr_HR_T/numCols_HR_T-1;
SDRV_HR_T= std(valuesOfDi_HR_T,0);

ARV_SBP_SP= Total_Di_SBP_SP/numCols_SBP_SP-1;
RMSRV_SBP_SP= Total_Di_sqr_SBP_SP/numCols_SBP_SP-1;
SDRV_SBP_SP= std(valuesOfDi_SBP_SP,0);

ARV_SBP_T= Total_Di_SBP_T/numCols_SBP_T-1;
RMSRV_SBP_T= Total_Di_sqr_SBP_T/numCols_SBP_T-1;
SDRV_SBP_T= std(valuesOfDi_SBP_T,0);

ARV_DBP_SP= Total_Di_DBP_SP/numCols_DBP_SP-1;
RMSRV_DBP_SP= Total_Di_sqr_DBP_SP/numCols_DBP_SP-1;
SDRV_DBP_SP= std(valuesOfDi_DBP_SP,0);

ARV_DBP_T= Total_Di_DBP_T/numCols_DBP_T-1;
RMSRV_DBP_T= Total_Di_sqr_DBP_T/numCols_DBP_T-1;
SDRV_DBP_T= std(valuesOfDi_DBP_T,0);

valuesOfMean_HR_SP(num)= Mean_HR_SP;
valuesOfSD_HR_SP(num)= SD_HR_SP;
```

```
valuesOfCV_HR_SP(num)= CV_HR_SP;
valuesOfARV_HR_SP(num)= ARV_HR_SP;
valuesOfRMSRV_HR_SP(num)= RMSRV_HR_SP;
valuesOfSDRV_HR_SP(num)= SDRV_HR_SP;
valuesofMean_HRV_LF_HF_SP(num)= Mean_HRV_LF_HF_SP;

valuesOfMean_HR_T(num)= Mean_HR_T;
valuesOfSD_HR_T(num)= SD_HR_T;
valuesOfCV_HR_T(num)= CV_HR_T;
valuesOfARV_HR_T(num)= ARV_HR_T;
valuesOfRMSRV_HR_T(num)= RMSRV_HR_T;
valuesOfSDRV_HR_T(num)= SDRV_HR_T;
valuesofMean_HRV_LF_HF_T(num)= Mean_HRV_LF_HF_T;

valuesOfMean_SBP_SP(num)= Mean_SBP_SP;
valuesOfSD_SBP_SP(num)= SD_SBP_SP;
valuesOfCV_SBP_SP(num)= CV_SBP_SP;
valuesOfARV_SBP_SP(num)= ARV_SBP_SP;
valuesOfRMSRV_SBP_SP(num)= RMSRV_SBP_SP;
valuesOfSDRV_SBP_SP(num)= SDRV_SBP_SP;
valuesOfMean_SBP_LF_HF_SP(num)= Mean_SBP_LF_HF_SP;

valuesOfMean_SBP_T(num)= Mean_SBP_T;
valuesOfSD_SBP_T(num)= SD_SBP_T;
valuesOfCV_SBP_T(num)= CV_SBP_T;
valuesOfARV_SBP_T(num)= ARV_SBP_T;
valuesOfRMSRV_SBP_T(num)= RMSRV_SBP_T;
valuesOfSDRV_SBP_T(num)= SDRV_SBP_T;
valuesOfMean_SBP_LF_HF_T(num)= Mean_SBP_LF_HF_T;

valuesOfMean_DBP_SP(num)= Mean_DBP_SP;
valuesOfSD_DBP_SP(num)= SD_DBP_SP;
valuesOfCV_DBP_SP(num)= CV_DBP_SP;
valuesOfARV_DBP_SP(num)= ARV_DBP_SP;
valuesOfRMSRV_DBP_SP(num)= RMSRV_DBP_SP;
valuesOfSDRV_DBP_SP(num)= SDRV_DBP_SP;
valuesOfMean_DBP_LF_HF_SP(num)= Mean_DBP_LF_HF_SP;

valuesOfMean_DBP_T(num)= Mean_DBP_T;
valuesOfSD_DBP_T(num)= SD_DBP_T;
valuesOfCV_DBP_T(num)= CV_DBP_T;
valuesOfARV_DBP_T(num)= ARV_DBP_T;
valuesOfRMSRV_DBP_T(num)= RMSRV_DBP_T;
valuesOfSDRV_DBP_T(num)= SDRV_DBP_T;
valuesOfMean_DBP_LF_HF_T(num)= Mean_DBP_LF_HF_T;


catch
  fprintf('!!!number %d got error in feature extraction for tilt position!!!\n', num);
```

```
            fprintf('enter to next number\n');
            pause;
            close all;
        end
    catch
        fprintf('@@@number %d not found!!!\n', num);
    end
end

B=cat(1,valuesOfMean_HR_SP,valuesOfSD_HR_SP,valuesOfCV_HR_SP,va
luesOfARV_HR_SP,valuesOfRMSRV_HR_SP,valuesOfSDRV_HR_SP,
valuesofMean_HRV_LF_HF_SP,...
valuesOfMean_HR_T,valuesOfSD_HR_T,valuesOfCV_HR_T,valuesOfARV
_HR_T,valuesOfRMSRV_HR_T,valuesOfSDRV_HR_T,
valuesofMean_HRV_LF_HF_T,...
valuesOfMean_SBP_SP,valuesOfSD_SBP_SP,valuesOfCV_SBP_SP,valuesO
fARV_SBP_SP,valuesOfRMSRV_SBP_SP,valuesOfSDRV_SBP_SP,valuesO
fMean_SBP_LF_HF_SP,...
valuesOfMean_SBP_T,valuesOfSD_SBP_T,valuesOfCV_SBP_T,valuesOfA
RV_SBP_T,valuesOfRMSRV_SBP_T,valuesOfSDRV_SBP_T,valuesOfMean
_SBP_LF_HF_T,...
valuesOfMean_DBP_SP,valuesOfSD_DBP_SP,valuesOfCV_DBP_SP,values
OfARV_DBP_SP,valuesOfRMSRV_DBP_SP,valuesOfSDRV_DBP_SP,value
sOfMean_DBP_LF_HF_SP,...
valuesOfMean_DBP_T,valuesOfSD_DBP_T,valuesOfCV_DBP_T,valuesOfA
RV_DBP_T,valuesOfRMSRV_DBP_T,valuesOfSDRV_DBP_T,valuesOfMea
n_DBP_LF_HF_T);
```

<u>Feature Selection</u>

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SequentialFeatureSelector as SFS
from sklearn.ensemble import RandomForestClassifier

features=pd.read_csv(r'D:\Users\Acer
User\Documents\FYP\Document\mean2_imputed_feature.csv',header=None)
classlabel=pd.read_csv(r'D:\Users\Acer
User\Documents\FYP\Document\classlabel2.csv',header=None)
classlabel=np.ravel(classlabel)

lr=LogisticRegression(random_state=42)
sfs = SFS(lr,n_features_to_select=5)
sfs = sfs.fit(features, classlabel)
for i in range(features.shape[1]):
    if sfs.support_[i]==True:
      print('Column: %d, Selected %s'%(i, sfs.support_[i]))
```

<u>Classification Model</u>

```
# Import libraries and functions
```

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import recall_score
from sklearn.metrics import make_scorer
from sklearn.model_selection import cross_validate
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
from imblearn.over_sampling import SMOTE

# Import dataset
features=pd.read_csv(r'D:\Users\Acer
User\Documents\FYP\Document\forward_lr.csv',header=None)
classlabel=pd.read_csv(r'D:\Users\Acer
User\Documents\FYP\Document\classlabel2.csv',header=None)
classlabel=np.ravel(classlabel)

# Apply SMOTE to balance the dataset
Smote.= SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(features, classlabel)

#Split data into train and test
import random
random.seed(42)
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled,
test_size = 0.20,random_state=42)

# Apply GridSearchCV
param_grid= {'criterion': ['gini', 'entropy'],'max_depth': [2, 4, 6, 8,
10],'min_samples_split': [2, 4, 6, 8, 10],'min_samples_leaf': [1, 2, 3, 4,
5],'max_features': ['sqrt', 'log2']}
grid=
GridSearchCV(DecisionTreeClassifier(random_state=42),param_grid,refit=Tr
ue,cv=5,verbose=2)
grid.fit(X_train,y_train)
print(grid.best_params_)

# Training model
clf=DecisionTreeClassifier(criterion='gini', max_depth= 2, max_features ='sqrt',
min_samples_leaf= 1, min_samples_split= 2,random_state=42)
clf.fit(X_train,y_train)
y_pred= clf.predict(X_test)

def specificity_score(y_test, y_pred):
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    specificity = tn / (tn + fp)
```

```
    return specificity

# Print performance
print("Accuracy: %.4f" % metrics.accuracy_score(y_test, y_pred))
print("Precision:%.4f" % metrics.precision_score(y_test, y_pred))
print("Recall:%.4f" % metrics.recall_score(y_test, y_pred))
print("Specificity:%.4f" % specificity_score(y_test,y_pred))
print("F1-score:%.4f" % metrics.f1_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

# Cross Validation
kf=StratifiedKFold(n_splits=5)
scoring = {'sensitivity':'recall',
           'specificity': make_scorer(specificity_score),
           'accuracy':'accuracy',
           'precision':'precision',
          'f1_score':'f1'}
cv_results = cross_validate(clf, X_train, y_train, scoring=scoring, cv=kf)

# Print Performance of Cross Validation
print('Accuracy:%.4f'%cv_results['test_accuracy'].mean())
print('Sensitivity:%.4f'%cv_results['test_sensitivity'].mean())
print('Specificity:%.4f'% cv_results['test_specificity'].mean())
print('Precision:%.4f'%cv_results['test_precision'].mean())
print('F1-score:%.4f'%cv_results['test_f1_score'].mean())
print (cv_results['test_accuracy'].max())
print (cv_results['test_accuracy'].min())
print (cv_results['test_accuracy'].std())
print (cv_results['test_sensitivity'].max())
print (cv_results['test_sensitivity'].min())
print (cv_results['test_sensitivity'].std())
print (cv_results['test_specificity'].max())
print (cv_results['test_specificity'].min())
print (cv_results['test_specificity'].std())
print (cv_results['test_precision'].max())
print (cv_results['test_precision'].min())
print (cv_results['test_precision'].std())
print(cv_results['test_f1_score'].max())
print(cv_results['test_f1_score'].min())
print (cv_results['test_f1_score'].std())
```