2019

Kidney Ailment Prediction under Data Imbalance

Ranaa Mahveen
rm0043@mix.wvu.edu

Kidney Ailment Prediction under Data Imbalance

Ranaa Mahveen

Thesis submitted to the

Benjamin M. Statler College of Engineering and Mineral Resources

at West Virginia University

In partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Donald Adjeroh, Ph.D., Chair

Yanfang ye, Ph.D

Lee Pyles, MD, MS

Lane Department of Computer Science and Electrical Engineering

West Virginia University

Morgantown, West Virginia

2019

Keywords: CKD, Data Imbalance, Classification, Random Re-sampling

ABSTRACT

Kidney Ailment Prediction under Data Imbalance

Ranaa Mahveen

Chronic Kidney Disease (CKD) is the leading cause for kidney failure. It is a global health problem affecting approximately 10% of the world population and about 15% of US adults. Chronic Kidney Diseases do not generally show any disease specific symptoms in early stages thus it is hard to detect and prevent such diseases. Early detection and classification are the key factors in managing Chronic Kidney Diseases.

In this thesis, we propose a new machine learning technique for Kidney Ailment Prediction. We focus on two key issues in machine learning, especially in its application to disease prediction. One is related to class imbalance problem. This occurs when at least one of the classes are represented by significantly smaller number of samples than the others in the training set. The problem with imbalanced dataset is that the classifiers tend to classify all samples as majority class, ignoring the minority class samples. The second issue is on the specific type of data to be used for a given problem. Here, we focused on predicting kidney diseases based on patient information extracted from laboratory and questionnaire data. Most recent approaches for predicting kidney diseases or other chronic diseases rely on the usage of prescription drugs. In this study, we focus on biomarker and anthropometry data of patients to analyze and predict kidney-related diseases.

In this research, we adopted a learning approach which involves repeated random data sub-sampling to tackle the class imbalance problem. This technique divides the samples into multiple sub-samples, while keeping each training sub-sample completely balanced. We then trained classification models on the balanced data to predict the risk of kidney failure. Further, we developed an intelligent fusion mechanism to combine information from both the biomarker and anthropometry data sets for improved prediction accuracy and stability. Results are included to demonstrate the performance.

Acknowledgements

I would like to express my deepest gratitude to my research advisor Dr. Donald Adjeroh for his invaluable assistance, support and encouragement throughout my research. This thesis would not have been possible without his intellectual support and the resources he made available to me. I am grateful to him for believing in me and providing me an opportunity to work in his enthusiastic research group and making my master's program a memorable experience. I am thankful to my committee members Dr. Yanfang Ye and Dr. Lee Pyles.

I am also thankful to Tim Mitchem for providing me an opportunity to work as a Graduate Service Assistant at CEHS, WVU. I would also like to thank the Lane Department of Computer Science and Electrical Engineering for giving me the opportunity to pursue my master's degree in such an academically vibrant environment.

Last but not the least, I owe my sincere thanks to my family and friends both back home and here, for their continued love and support whose love and sacrifice for me is beyond anything I will ever understand.

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Problem and Motivation

Chronic kidney disease (CKD) [1] is a worldwide public health problem. In the United States, there is a rising incidence and prevalence of kidney failure, with poor outcomes and high cost. The most common outcome of CKD is kidney failure which requires treatment with transplantation and dialysis. Disorders like diabetes, high blood pressure may trigger CKD [2]. However, cardiovascular disease (CVD) is also frequently associated with CKD. CVD in CKD is treatable and potentially preventable, and CKD appears to be a risk factor for CVD [3].

With an estimated prevalence of 8-16% worldwide, CKD is a major noncommunicable disease. CKD may be the cause of premature mortality and loss of disability – adjusted life year [4]. Early diagnosis and effective interventions with CKD can be challenging due to variety in terms of causes, progression mechanisms and histopathological manifestations [5].

In addition, CKD is a major drain on health resources, CKD and end-stage renal disease (ESRD) cost Medicare in the United States over $98 billion [6]. Owing to the increasing occurrence of CKD, China faces a great financial burden. In field of health informatics, the definitions and boundaries of big data is highly debatable [7]. Big data is defined as consisting of extensive datasets in terms of volume, variability, velocity and variety by the US National Institute of Standards and Technology that need a scalable architecture for proficient storage and analysis [8,9].

Over two million people around the world undergo dialysis or kidney transplant to stay alive, this represent only 10% of people who require treatment to live [10]. The majority of the people who receive treatment for kidney failure are in five relatively wealthy countries, which is 12% of the global population. On an average more than one million people in 112 lower – income countries die from untreated kidney failure annually, because of the huge financial burden of kidney transplantation treatment or dialysis [11].

Thus, there is an urgent need for early detection, controlling, and management of the disease. It is necessary to predict the progression of CKD with reasonable accuracy because of its dynamic and covert nature in the early stages, and patient heterogeneity. CKD is often

described by severity stages. Therefore, Machine learning can play a major role in extracting hidden patterns from the large patient medical and clinical dataset that physicians frequently collect from patients to obtain insights about the diagnostic information, and to implement precise treatment plans. Machine learning techniques are applied and used widely in various contexts and fields. With machine learning techniques we could predict, classify, filter and cluster data. The goal or prediction attribute refers to the algorithm processing of a training set containing a set of attributes and outcomes.

1.2 Thesis Contributions

The contributions of this thesis are summarized as follows:

- A detailed study on resampling methods to handle data imbalance problem
- Developed Improved resampling methods to handle data imbalance problem
- Proposed an intelligent data fusion method and analyze the stability and reliability of the results
- Analysis of results to discover the best model among all and evaluation of results with the baseline results

1.3 Thesis Outline

Chapter 2 presents a detailed review of the existing literature on topics related to this thesis. Chapter 3 introduces various methods to tackle data imbalance problem, all the pre-processing stages of biomarker and anthropometric data, classification methods for prediction and our methodology to deal with the data imbalance problem in predicting kidney failure. Chapter 4 provides information of the datasets used in this study and the results with performance analysis using different classification methods. Finally, Chapter 5 presents our overall conclusions and future work.

Chapter 2: Background

In this chapter, we broadly discuss the existing methodologies used in Kidney ailment predictions, the strategies to deal with data imbalance problem and classification techniques used in biomedical informatics, especially in disease prediction.

2.1 Kidney Ailment Prediction Methods

The main function of kidneys is to filter the blood. Blood passes through the kidneys several times a day. The kidneys remove wastes, control the body's fluid balance, and regulate the balance of electrolytes. Each kidney contains around a million units called nephrons, each of which is a microscopic filter for blood. Disorders affect kidney function and structure in varying forms. It's possible to lose as much as 90% of kidney function without experiencing any symptoms or problems [11].

Prediction methods that can identify individuals at high risk of developing kidney failure have great clinical value. These prediction methods can be used in determining the right time to refer to consult a nephrologist. CKD prediction methods might also help in improving health policies and risk stratification [12].

Machine learning is a field of study concerned with study of large sets of data. It involves algorithms, techniques for analysis, computational learning theory and it is evolved from pattern recognition. Machine learning is a promising field in medical science's perspective, it can help physicians make optimal diagnosis to choose medications for their patients and improve patient's condition by minimizing expenses.

Machine learning and data mining techniques together have good success rate in prediction and diagnosis of many critical diseases. Machine learning techniques can often be applied to predict critical diseases, since they improve the efficiency of the systems. The features used to in predicting the diseases can be continuous, categorical or binary. If the samples are given with the corresponding correct outputs or outcomes, then the concerned data is called supervised and corresponding learning is called supervised learning, on the other hand, in unsupervised learning samples are unlabeled or the outcome of the feature set is unknown. Classification is a function that assigns items in a collection to target categories or classes. The goal of classification is to predict the target class for each instance in the data. Different

classification approaches and machine learning algorithms are applied for prediction of chronic diseases. Often, chronic kidney disease is diagnosed as a result of screening of people known to be at risk of kidney problems, such as those with high blood pressure or diabetes and those with a blood relative with CKD. It is differentiated from acute kidney disease in that the reduction in kidney function must be present for over 3 months.

Major *et al.* [13] discuss the Kidney Failure Risk Equation (KFRE) which uses 4 variables, age, sex, urine albumin-to-creatinine ratio (ACR) and Glomerular filtration rate (GFR) in people who already have CKD to predict the risk of end stage renal disease (ESRD). This predicts kidney failure and the need for dialysis or kidney transplant within next 2 to 5 years. These prediction models are referred as clinical risk prediction models.

The aim of clinical risk prediction models is to estimate the risk of an event for an individual using their related information. Prognostication in clinical practice, to assist research planning, to aid treatment decisions in relation to clinical trials, to asses resource management, and healthcare systems are the 3 main purposes of risk prediction models [14].

Risk models are updated using processes like recalibration, this is a common way and is likely to enhance the performance of a model in different geographical and temporal settings [15]. Few of the risk models are externally validated in other populations or their potential impact is studied [16].

Meta-analysis of data samples from 31 cohorts of predominately North American CKD populations is used to develop prediction tools for ESRD. Subsequently, 3 ESRD prediction equations were derived based on 4, 6 or 8 variables. These models included variables of age, sex, GFR, urine albumin-to-creatinine ratio (ACR) along with additional variables of hypertension or serum albumin, diabetes mellitus, bicarbonate, calcium and phosphate [17].

The model based on parsimony was recommended for implementation into clinical practice, however the 4-varibale Kidney Failure Risk Equation's (KFRE) performance was same as that of the other 2 equations. A calibration factor was proposed as the overall risk was found to be lower in non-North American cohorts [17].

Jena and Kamila [18] predicted and analyzed kidney disease using different algorithms like Support Vector Machines (SVM), Naïve Bayes classifier, Multilayer perceptron, conjunctive rule, J48 classifier in Waikato Environment for Knowledge Analysis (WEKA) tool.

For efficient prediction of kidney diseases, different techniques have been proposed by exploiting patient's medical data. Chatterjee *et al.* [19] presented a Cuckoo Search trained neural network (NN-CS) method for the identification of CKD [19]. Initially, the issues that exist in local search-based learning methods are being resolved by this model. The Cuckoo Search algorithm helps to efficiently selecting the input weight vector of the Neural Network.

Chen *et al.* [20] proposed two fuzzy classifiers known as fuzzy rule-building expert system (FuRES) and fuzzy optimal Associative Memory (FOAM) for the identification of CKD. FuRES generates a classification tree which comprises a minimal NN. It creates the classification rules to determine the weight vector with the least fuzzy entropy. The two fuzzy classifiers are employed for the identification of 386 CKD patients. Also, FuRES is better compared to FOAM especially in situations where the training, as well as the prediction process, contain a similar intensity of noise. FuRES and FOAM attained better performance in the identification of CKD; at the same time, FuRES more efficient than FOAM.

K.R.Lakshmi et.al [21] proposed performance evaluation of three data mining techniques for predicting kidney dialysis survivability. In this research, various data mining techniques are used to extract knowledge about the interaction between these variables and patient survival. The concepts introduced in this research have been engaged and tested using a data collected at different dialysis sites. Finally, ANN is suggested for kidney dialysis survivability analysis for improved performance in terms of accuracy.

Several studies have analyzed patient data to predict kidney diseases using machine learning techniques, most of the research has been focused on the prediction and the classification algorithms. In our study, most of the patients in our dataset do not suffer from any kidney ailment making the dataset highly imbalanced. This leads to the problem of data imbalance which makes prediction highly unstable. In Section 2.2, we discuss the problem of class/data imbalance problem in detail.

2.2 Class/Data Imbalance Problem

Data imbalance problem in classification has been addressed with hundreds of algorithms in the past decade. In this section, an overview of the imbalanced learning techniques that have been used are discussed. We discuss two basic strategies used for handling imbalanced learning, namely preprocessing and cost-sensitive learning. Resampling methods which are conducted in the sample space and feature selection methods that improve the performance of the feature space, this is one of the approaches in preprocessing. In Section 2.1, we give an overview of basic strategies for tackling with imbalanced learning.

2.2.1 Basic strategies for dealing with imbalanced learning

2.1.1 Preprocessing techniques

Preprocessing of the data is done before training a learning model to gain an appropriate input data. Two classical techniques are often used as preprocessor considering the representation spaces of data.

2.1.1.1 Resampling

Resampling techniques are often used to balance the imbalanced data in the sample space to improve the effect of the skewed class distribution in the learning. López et al., 2013 categorized resampling methods to be more versatile because these methods are independent of the classifier. These methods further fall into three groups based on the method employed to rebalance the class distribution.

- Over-sampling methods: This involves increasing the class distribution by creating new minority class samples. Chawla et al., 2002 [22] discuss SMOTE as a method to create synthetic minority classes and the other method is randomly duplicating the minority samples.
- Under-sampling methods: This involves eliminating the intrinsic samples in the majority class, thereby balancing the class distribution. Tahir et al., 2009[23] described Random Under Sampling to be the most effective method, this involves elimination of majority class examples.
- Hybrid methods: These methods are the combination of over-sampling and under-sampling methods.

Most of the reviewed papers on data imbalance techniques use resampling techniques, this indicates resampling is a popular strategy for handling imbalanced data. Under-sampling, over-sampling, hybrid-sampling are predominantly used to deal with data imbalance problem. Others developed new techniques based on cluster methods, generic algorithms and distance methods. All the resampling methods balance the data up to certain ratio desired by the user, and it is not required to balance the number of majority and minority classes equally. Zhou (2013) [24] recommended different sample ratio for different data sizes, Lu et al., 2016 [25] studied ways to automatically decide optimal sampling rate for different problem settings and imbalanced ratios.

Napierala and Stefanowski (2015) [26] tried different types minority class samples and their effects on learning classifiers from the imbalanced data. From all the methods that have been implemented, some of the major insights are: When the data has hundreds of minority class observations, an under-sampling method was considered superior to an over-sampling method with regard to computational time. When there are few minority class observations, SMOTE (an over-sampling method) was considered as a better option. A combination of SMOTE and under-sampling is found to be a better choice when the size of training sample is too large and SMOTE is an effective method in recognizing outliers.

2.1.1.2 Feature Selection and Extraction:

Li et al., 2016c [27] discussed the importance of feature selection when compared to the resampling methods. They found that removing irrelevant features in the features space is more efficient because under imbalanced cases and minority class samples can easily be eliminated as noise using resampling techniques. The aim of feature selection is to allow a classifier to achieve optimal performance by selecting a subset of k features from the feature space, where k is a user specified parameter. Guyon and Elisse- eff, 2003[28] divided feature selection into filters, wrappers and embedded methods. Saeys et al. (2007) [29] discussed the advantages and disadvantages of these methods.

Motoda and Liu, 2002[30] discussed feature extraction as the other way to deal with dimensionality reduction. Dimensionality reduction converts the data into a low-dimensional space, this is related to feature extraction. This technique of feature extraction is quite different

from feature selection. Feature extraction uses functional mapping to create new features from the existing features, whereas a subset of the original features is returned with feature selection.

Hartmann, 2004[31] proposed a variety of techniques for feature extraction, Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and f Non-negative Matrix Factorization (NMF) are a few of them. Feature Extraction methods are used more frequently for unstructured data like images, text and speech.

In feature selection methods, filter and wrapper feature selection methods were most frequently used. Heuristic choice was a common choice to rank the features for wrapper methods and different metrics were used for filter methods. Casañola-Martin et al., 2016 [32] feature selection and feature extraction in solving real world problems such as disease diagnosis. Zhang et al., 2015a [33] used feature selection in textual sentiment analysis, Lima and Pereira, 2015 [34] used in fraud detection and other rare events detection problems

2.1.2 Cost-sensitive learning

Cost-sensitive learning can be integrated at the algorithmic level as well as the data level considering higher costs for misclassification of minority class samples with respect to majority class samples. Costs are identified as cost matrices, Ghazikhani et al., 2013b [35] described cost matrices in a specific domain can be determined using data stream scenarios and expert opinion. When compared with re-sampling techniques, cost-sensitive learning is found to be computationally efficient, therefore it is more suitable for big data streams. Nevertheless, cost-sensitive learning was not being used much when compared to the resampling methods.

Krawczyk et al. (2014) [36] stated that there may be two potential reasons, one is that setting the values in the cost matrix is difficult because in most of the cases, cost of the misclassification is not known from the data and cannot be determined by the expert. However, an alternate way to handle this difficulty was discussed by Castro and Braga, 2013[37] where the majority class misclassification cost was set at 1 while setting the penalty minority class value as equal to the imbalanced ratios. Ensemble models and single ensemble models directly implement resampling methods, unlike cost-sensitive learning which requires modification in the learning algorithm. Re-sampling methods are predominantly used instead of cost-sensitive learning.

2.2.2 One – Class Classification (OOC)

The multi-class classification aims to classify an unknown data object into one of several pre-defined categories. A problem arises when the unknown data object does not belong to any of those categories. Let us assume that we have a training data set comprising of instances of fruits and vegetables. Any binary classifier can be applied to this problem, if an unknown test object is given for classification. But if the test data object is from an entirely different domain, the classifier will always classify a cat as either a fruit or a vegetable, which is a wrong result in both the cases. Sometimes the classification task is just not to allocate a test object into predefined categories but to decide whether it even belongs to any of the classes or not. In the above example, an apple belongs to class fruits and the cat does not.

Juszczak [38] defined One-Class Classifiers (OCC) as class descriptors that are able to learn restricted domains in a multi-dimensional pattern space using primarily just a positive set of examples. In OCC one of the classes is well characterized by instances in the training data, while the other class has either no instances or very few of them, or they do not form a statistically representative sample of the negative concept [39]. For instance, in automatic diagnosis of a disease, positive data can be easily obtained when compared to negative data since other patients in the database cannot be assumed to be negative cases if they have never been tested, and such tests can be expensive. Thus, OCC can be viewed as one approach to the data imbalance problem

Classification methods and problems have been considered a major part of Machine Learning as a large amount of applications have been using these methods. Machine Learning is considered to be a broad concept which includes supervised, unsupervised and semi-supervised problems. Each data input is assigned with a class label in Supervised learning problem, the main task is to learn a model that gives the same labeling for the unknown data. Whereas, in unsupervised learning problem, data samples are unlabeled, and the task is to discover and analyze the structure of the data. This is mainly useful when there are differential clusters or groups in the data. Semi-supervised learning is also a broad research area on Machine Learning, its main objective is that when compared with labeled data, unlabeled data is easily available, and this data is crucial for decision functions in most of the situations.

2.2.3 PU learning

PU learning is learning from Positive and Unlabeled data, it is a special case of binary classification. The labeling mechanism is a key concept in PU learning, the goal is same as binary classification. However, only some of the positive examples in the training data are labeled and none of the negative sample are labeled during the learning phase.

PU leaning has been explored for text mining [40], these algorithms share a two-step framework. The other related studies which explored PU learning are disease gene identification [41] and protein function identification [42].

Gieseke, F *et al.* (2014) [43] used semi-supervised classification for obtaining better classifiers, in this setting of semi-supervised classification, both labeled and unlabeled samples are used during the construction of the classifier model to balance the information obtained. Lee et al. [44] pointed out that sometimes unsupervised learning is applied to get labels for training classifiers or to get some other parameters of the classification models.

The man aim of supervised classification algorithms is to divide the classes of the problem using only the training data. The problem is considered as binary classification if the output has two possible outcomes, and the problem is referred to as multi-class classification if there are more than two classes.

In our study, we predict the outcome of kidney failure from different feature set. The considered dataset is 2 class, so this is a binary class problem. The ratio of patients who have kidney ailments to the patients who do not is very high. Thus, this can be considered as a special case of PU learning where the data or an unbalanced data set problem.

Chapter 3: Methodology

3.1 Empirical Analysis of Data Imbalance Problem

3.1.1 Re-sampling Techniques

In this study, we have used various re-sampling techniques to address the problem of data imbalance. The objective of re-sampling is to balance the class distribution, this method is the most direct way to deal with class distribution. There are many different forms of re-sampling such as active sampling techniques, random under-sampling, random over-sampling and combinations of above-mentioned techniques. We employed random over-sampling technique SMOTE and our own random sub-sampling technique to deal with data imbalance problem in our Biomarker and Anthropometric Data set.

3.1.1.1 Random over-sampling

Random over-sampling replicates the minority class randomly, this method can increase the likelihood of over-fitting because it creates copies of minority class. There are many heuristic over-sampling methods, such as SMOTE, and its variations. We tried to oversample the minority classes using SMOTE which balanced the data by creating synthetic samples for minority class. The modification of amount of class data using sampling methods gives a balanced class distribution. Various sampling methods have been proposed to tackle the problem of data imbalance. Chawla et.al [22] proposed an over-sampling approach called **SMOTE**, which stands for **Synthetic Minority Oversampling Technique**. This approach has been widely accepted and gives the best result when dealing with imbalanced datasets. Kubat et al. [45] created their own training datasets by selectively under-sampling the number of data points of majority class by keeping the number of minority class constant. This method of under-sampling the majority class has a scope to build better classifiers. However, a combination of under-sampling and over-sampling approaches did not result in classifiers that are better than those built using only under-sampling approach. Therefore, over-sampling the minority class does not usually improve the accuracy of predicting the minority class. [46]

SMOTE is an over-sampling approach in which the minority class is over-sampled by creating synthetic examples rather than by over-sampling with replacement. In this method, synthetic examples are generated by operating in feature space rather than in data space. In this approach,

the minority class is over-sampled by taking each minority class sample and introducing synthetic samples along any of the *k* minority class nearest neighbors. *K* nearest neighbors are randomly chosen depending on the required amount of over-sampling. For example, if the required amount of over-sampling is 400% only four neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each.

Synthetic samples are usually generated in this way: S_{min} is a subset of minority class from the whole set S, for each instance $x_i \in S_{min}$, find its K nearest neighbors by using Euclidean distance. To generate a new synthetic sample, randomly select one of the K-nearest neighbors, calculate the feature difference between x_i and its neighbor. Multiply this difference by a random number between 0 and 1, add this to the feature vector under consideration to get the synthetic sample x_{new}. This selects a random point between two specific features, making the decision region of the minority class more general.

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \alpha$$

In the above equation, $x_i \in S_{min}$ is an instance of minority class in original dataset, $\hat{x}_i$ is one of the K-nearest neighbors of x_i, and $\alpha \in [0,1]$ is a real random number. So, the new synthetic sample is a data point between x_i and the randomly selected K nearest neighbor $\hat{x}_i$. Larger and fewer specific regions are created with the synthetic samples. The SMOTE model shown below in Algorithm 1 and Algorithm 2 using a pseudo code.

3.2 Data Preprocessing Techniques

Data preprocessing is one of the major part of data analytics, the collected data is crude and might contain noise. The attributes required for the analysis are not always in the format we need it to be. Most times, the attributes in the dataset are in different scales, if we use any analytic technique without preprocessing, we end up with unrealistic results. The other issue with the data is occurrence of extreme data points, which are considered as outliers. When the dataset consists of these points, the performance of classification or regression models decline. These are the two major issues with the datasets. To handle scaling issues, data transformation techniques are employed, and various techniques are used to deal with outliers in the dataset.

Algorithm 1: SMOTE

Input: T:Training set; N : $N\%$ amount of synthetic samples ; k: Number of nearest neighbors.

Output: S: Set of synthetic samples

```
begin
    X ⟵ MinorityInstances(T);
    n ⟵ NumberOfInstances(X);
    p ⟵ NumberOfVariables(X);
    S ⟵ ∅;
    if N < 100 then
        n ⟵ (N/100) × n ;                          /* N% will be SMOTEd */
        X ⟵ RandomSample(X, p);
        N ⟵ 100;
    end
    N ⟵ N/100;
    for i ← 1 to n do
        X̂_i ⟵ KNN(i, X) ;                  /* Compute k nearest neighbor for i */
        while N ≠ 0 do
            β ⟵ RandomeNumber(1, k) ;              /* chose one neighbor of i */
            for j ← 1 to p do
                α ⟵ RandomeNumber(0, 1);
                S_ij ⟵ X_ij + (X̂_iβj − X_ij) × α;
            end
            N ⟵ N − 1;
        end
    end
    return S
end
```

Algorithm 2: SMOTE with Categorical Variables

Input: T:Training set; N : $N\%$ amount of synthetic samples ; k: Number of nearest neighbors.

Output: S: Set of synthetic samples

```
begin
    X ⟵ MinorityInstances(T);
    n ⟵ NumberOfInstances(X);
    p ⟵ NumberOfVariables(X);
    S ⟵ ∅;
    if N < 100 then
        n ⟵ (N/100) × n ;
        X ⟵ RandomSample(X, m);
        N ⟵ 100;
    end
    N ⟵ N/100;
    for i ← 1 to n do
        X̂_i ⟵ KNN(i, X) while N ≠ 0 do
            β ⟵ RandomeNumber(1, k) ;
            for j ← 1 to p do
                if variable j is categarical variable then
                    S_ij ⟵ MajorityVote(X̂_ij) ; /* compute categorical variable */
                else
                    α ⟵ RandomeNumber(0, 1);
                    S_ij ⟵ X_ij + (X̂_iβj − X_ij) × α;
                end
            end
            N ⟵ N − 1;
        end
    end
    return S
end
```

3.2.1 Data Transformation

Data transformation such as normalization, represent an important data pre-processing technique in machine learning. An attribute of a dataset is normalized by scaling its values so that all attributes fall within a small-specified range, such as 0.0 to 1.0. Normalization is particularly useful for classification algorithms involving neural networks, or distance measurements such as nearest neighbor classification and clustering. Normalization performs data smoothing and data normalization preparatory to modeling. The technique is easy to apply by using standard mathematical transformations such as min-max normalization to numerical columns, z-score normalization, log normalization, or decimal scaling normalization. Extreme values in data can make it difficult to detect patterns. When the data is very irregular, has very high or very low values, or values are scattered or do not follow a Gaussian distribution, normalizing the data can help fit the data to a distribution that better supports modeling.

3.2.1.1 Min-Max Normalization

This method rescales the features or outputs from one range of values to a new range of values. More often, the features are rescaled to lie within a range of [0,1] or from [-1, 1]. The rescaling is often accomplished by using a linear interpretation formula, such as:

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Where is the maximum value of attribute, is the minimum value of attribute for $(new\_max_A - new\_min_A) = 0$, When $(max_A - min_A) = 0$, it indicates a constant value for that feature in the data. When a feature value is found in the data with a constant value, it should be removed because it does not provide any information to the machine learning algorithms. When the min-max normalization is applied, each feature will lie within the new range of values which will remain the same for all features. Min-max normalization has the advantage of preserving exactly all relationships in the data.

3.2.1.2 Decimal Scaling Normalization

Normalization by decimal scaling normalizes by moving the decimal point of attribute value. The number of decimal points moved depends on the maximum absolute value the attribute. For a given attribute A, the decimal scale normalization is performed as follows:

$$A' = \frac{A}{10^m}$$

Where m is the smallest integer such that Max |A'|<1.

3.2.1.3 Z-Score Normalization (Statistical)

Z-score normalization is also called zero-mean normalization; this technique uses the mean and standard deviation for each feature across a set of training data to normalize each input feature vector. The mean and standard deviation are computed for each feature. The transformation is given in the general formula:

$$v' = \frac{v - \mu_A}{\sigma_A}$$

Where μ_A is mean of attribute A and σ_A = standard deviation of attribute A. This produces data where each feature has a zero mean and a unit variance. Z-Score normalization technique is applied to all the feature vectors in the data set first; creating a new training set and then training is started. Once the means and standard deviations are computed for each feature over a set of training data, they must be retained and used as weights in the final system design. In our research, we employed Z – score normalization to normalize the biomarkers and body measures. The normalized data was then trained with different classification models.

As the dataset we had was highly skewed, we employed various outlier treatments to reduce the noise in the data before we train the classification models.

3.2.2 Identifying Outliers

Outliers are the data points lying far away from most of the other data points. Outlier identification should be performed before data analysis because most of the statistical tests assume that data is normally distributed. There are various methods to identify outliers. To

determine an outlier, one of the methods measures the distance between data point and the center of all data points to find an outlier. In this method, outliers are determined depending on the Standard Deviation (SD), i.e., the data points which do not fall within certain SD of the mean are considered as outliers. Nevertheless, this method of using SD and mean are not regarded as proper as the SD and mean are statistically sensitive to the presence of outliers. On the other hand, the quartile range and median are more efficient because these are less sensitive to outliers [47].

In our study, we used boxplots to identify outliers. Boxplots are another way of discovering the outliers by differentiating the data points based on the placement of the points within and outside the fence lines. Figure 3.1 shows a box plot with fence lines; fence lines are used to determine the points to be considered as outliers. The data points that lie outside the upper or lower fence lines are considered as outliers [48].

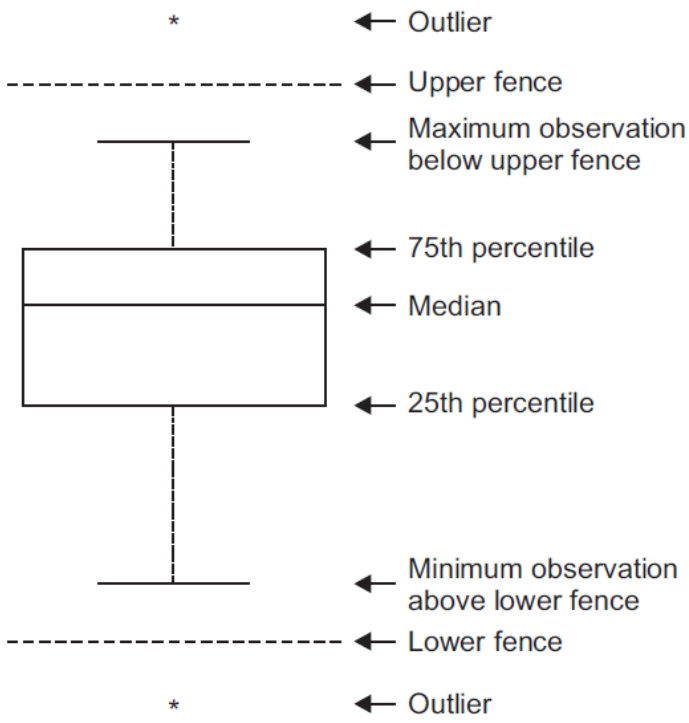

Figure 3.1: Box plot

Different techniques have been explored in many studies with respect to outlier identification. To detect outliers, regression analysis uses simple residuals, which are adjusted by

standardized residuals and predicted values against observed values [49]. For this same purpose a Support Vector Regression is also used [50]. When the same type of information is collected from different groups or if information is collected repeatedly from a single participant, the need for outlier detection increases. In some cases, outlier detection is studied based on the mean and variance of group data [51]. Univariate outliers are can be determined by a simple boxplot. To identify multivariate outliers, statistical tests which consider the relationships between variables are required.

3.2.2.1 Treatment of Outliers

There are basically three methods for treating outliers in a data set. One method is to remove outliers as a means of trimming the data set. Another method involves replacing the values of outliers or reducing the influence of outliers through outlier weight adjustments. The third method is used to estimate the values of outliers using robust techniques. In our research, we employed trimming to remove outliers based on Cook's distance [52].

Cook's distance is a measure computed with respect to a given regression model and therefore is impacted only by the X variables included in the model. It computes the influence exerted by each data point on the predicted outcome. The cook's distance for each observation *i* measures the change in $\hat{Y}$(fitted Y) for all observations with and without the presence of observation *i*, so we know how much the observation *i* impacted the fitted values. Mathematically, cook's distance D_i for observation *i* is computed as follows:

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \times MSE}$$

where,

- $\hat{Y}_j$ is the value of j_{th} fitted response when all the observations are included
- $\hat{Y}_{j(i)}$ is the value of j_{th} fitted response, where the fit does not include observation *i*.
- *MSE* is the mean squared error.
- *p* is the number of coefficients in the regression model.

The observations that have a Cook's distance (D_i) greater than 2 times the mean are classified as influential in our study.

3.2.3 Feature Selection Methods with Boosting Ensemble

The function *featureSelection* in R implements a feature selection algorithm leveraging the ideas from backpropagation and randomness. In this algorithm, a new random stability matrix with two columns is initialized. Name of the feature and stability score indicating the empirical relevance score of a feature are the contents of the two columns. The random matrices are updated over the course of component-wise boosting models on random subsets of the feature space. The updating process works as follows:

- If a feature was contained in a subset, but was not selected in the boosting, it's score in the randomly initialized matrix is reduced by the amount of the penalty.
- If a feature was contained in a subset and was selected in a boosting, it's score in the randomly initialized matrix is increased by the amount of the reward.
- After n_mods models in each n_rounds rounds the n_rounds updated stability matrices are combined by simply averaging the scores for each feature across all matrices.

3.3 Improved Resampling Method

Our proposed study uses Biomarker and Anthropometry data to predict kidney ailments. The collected data has majority of the patients without kidney disease making the data highly imbalanced. The biomarker data and anthropometric data are preprocessed separately, and classification algorithms are used on each dataset for predicting the risk of kidney failure.

Before proceeding with data resampling method, the dataset is treated to remove the outliers using Cook's distance as discussed in section 3.2.3. Further, to select the features which contribute the most to prediction outcome, automated feature selection method is used. We used automated feature selection function *featureSelection* in R to get the stability matrix for feature set of anthropometry and biomarker data. The top 6 features were selected based on the feature importance values.

Now, the dataset is free from outliers and feature space of the data is optimal, but still the dataset is not balanced. To overcome this imbalance problem, we employed repeated data sub-sampling on the data after removing the outliers. Negatively labelled samples are divided into sub-samples equal to the number of positively labelled samples, this makes the training set completely balanced. All the sub-sample sets are trained on the classification models, and the trained models are used to predict the outcome on the test set. We trained the balanced data on

Random Forest (RF) and SVM models. The performance of the classification models for each group is recorded for both anthropometry and biomarker data.

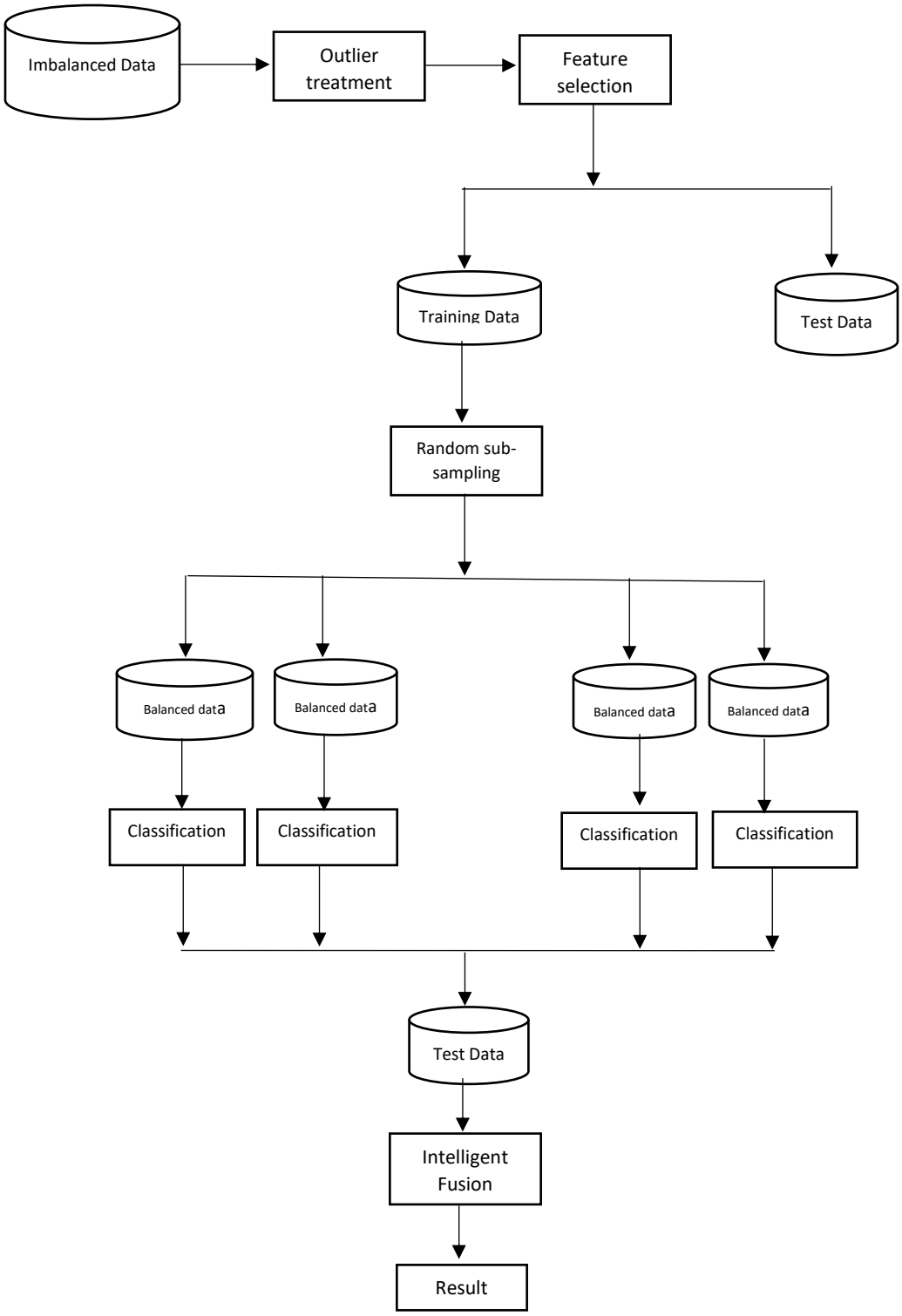

Figure 3.2: Schematic diagram for improved resampling method

Algorithm 3: Pseudo code for Improved Re-sampling method

D: Anthropometry/Biomarker Dataset, $\mathbf{P_i}$: Positively labelled training samples (Minority), N_i: Negatively labelled training samples (Majority)

```
1. begin
2.   Treat the D to remove outliers
3.   Select appropriate features
4.   Divide D into train (T_i) and test set (T_j) set
5.   s ← N_i/P_i;
6.   for i = 1 → s
7.     n_i ← N_i/ s;
8.     t_i ← merge (P_i,n_i);
9.     m_i ← svm (t_i) or rf(t_i);      /* SVM or random forest algorithms*/
10.    r_i ← predict (m_i, T_j);
11.  end for
12.  a ← averageAccuracy(r_i);        /* majority vote rule*/
13.  p ← selectTopPredictors(m_i);              /*based on test accuracy */
14.  h ← selectTopPredictor(m_i);
15.  c ← compareAccuracies(a, p, h);
16.  return c;
17. end
```

3.4 Intelligent Data Fusion

In this section, we discuss the methods used to select best predictors from the set of predictors used to predict the test data. To select the predictors which are more reliable on varied datasets, we propose different fusion methods. Data fusion methods are as follows:

3.4.1 Combining Results after resampling

The prediction results obtained from improved resampling method are analyzed to get a specific pattern for the predictor groups. Majority vote method (Decision-level fusion) i.e., for an individual, class predicted by majority of the predictor groups is considered as the final predicted

outcome, and this outcome is used to calculate the average accuracy of the model of all predictors, top 5, top 10 and top 15 predictor outcomes. These set of predictors are further analyzed to check the accuracy pattern. The standard deviation is calculated for the average predictors to check the consistency in accuracy for all the predictor groups. This analysis is done on RF and SVM classification model results for both anthropometry and biomarker data.

3.4.2 Feature-level fusion

Further, to exploit the full potential of data fusion methods, we fused the selected features of anthropometry and biomarker after removing the outliers. The dataset with fused feature set is balanced using resampling method. The resulting group of sub-samples are completely balanced, it is then trained using classification models. The trained models are used to classify the test data, the performance of each model is recorded for further analysis. The results are combined to look for certain patterns in the predictor groups, this is done same the analysis on the individual dataset results using the majority vote method.

3.4.3 Combining Feature-level and Decision-level fusion

In this method, we improvise the data fusion technique in a more intelligent way by combining the feature-level and decision-level fusion techniques. By using the majority vote method on combined feature, anthropometry and biomarker prediction results, we decide the predicted result of an individual. We, then calculate the accuracy of the model based on decided outcome for the classification models. The performance is analyzed based on the average accuracy of top predictors for combined features data, anthropometry and biomarker datasets and standard deviation is calculated to show the stability of each result.

Further, the results of anthropometry and biomarker dataset are grouped into 4 groups based on the test dataset's predicted outcome of the classification models. Table 3.1 shows the criteria for the groups.

Groups	Classification Models
Anthropometry (0) – Biomarker (0)	Both the datasets predicted incorrectly
Anthropometry (0) – Biomarker (1)	anthropometry predicted incorrectly – biomarker predicted correctly
Anthropometry (1) – Biomarker (0)	anthropometry predicted correctly – biomarker predicted correctly
Anthropometry (1) – Biomarker (1)	Both the datasets predicted correctly

Table 3.1: Groups based on prediction results

3.5 Classification Methods

Different classification algorithms have been used to predict the risk of kidney failure in the recent past. The main process of classification in Machine Learning is to train classifier to recognize patterns from a given training samples and to classify test examples with the trained classifier. For several reasons, training a classifier that is as accurate as possible in classifying new samples is demanding. Several problems need to be considered when building the classifiers, the efficiency of the classifiers depend on the many factors. One of the problems is related to the dataset, if the training set is small, it becomes difficult to capture the underlying distribution of the data. Another problem is related to the model, mainly the model complexity and the its capabilities. If the classifier is too simple, it becomes difficult to capture the underlying structure of the data. On the other hand, if the classifier is complex and there are too many unnecessary parameters noise might be assimilated in the model leading to over-fitting. This leads to high accuracy in training model but performs poorly on test samples. There are two main types of learning schemes in machine learning i.e., supervised learning and unsupervised learning. Our research focuses on supervised binary classification, in the following section we summarize the supervised learning algorithms which were being employed to classify the patient data.

3.5.1 Support Vector Machines (SVM)

Support Vector Machines is a supervised machine learning algorithm used both for classification and regression challenges. SVM performs classification by finding the hyperplane that maximizes the margin between the two classes [53]. The vectors which define the hyperplane are the support vectors.

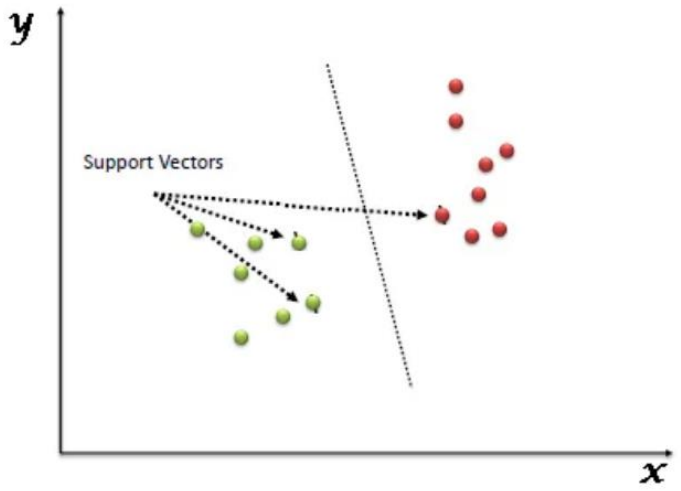

Figure 3.3: Support Vector Machines

SVM uses the input dataset with binary classes to train the training model in order to classify new observation to one of the two classes by creating a separating hyperplane [54]. Figure 3.3 shows the support vectors and the hyperplane dividing the data points into 2 halves, this is an ideal example for a binary class classification problem.

The algorithm labels the new examples or the unknown data samples through the created hyperplane. In this work, we performed SVM training using Rstudio with R's inbuilt function 'svm' from the package 'e1071' and the classification of new data samples or the prediction is performed using the function 'predict'. SVM can be performed with four different kernels; Linear, Radial Basis Function (RBF), Polynomial, and Quadratic. These can be accessible in R as parameters within in the 'svm' function. We used Linear kernel for our dataset. The mathematical formulation for each kernel is shown here [55]:

• Linear: $K(x, y) = w(x.\ y) + b$. The vector w is known as the weight vector and b is called the bias.

• Radial basis function – RBF: For some positive number σ:

$$K\ (x, y) = \exp\left[\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right]$$

$x_i$and x_j will have either one becoming the support vector and the other will be the testing data point.

• Polynomial: For some positive integer d:

o $K(x, y) = (1 + < x. y >)d$. Where d is the polynomial's degree

• Quadratic: $K(x, y) = (< x. y >)2$

SVM classifier is widely used in bioinformatics due to its high efficiency, theoretical importance regarding over-fitting, ability to deal with high-dimensional data, and flexibility in modeling diverse sources of data [56].

3.5.2 Random Forest

Random Forest is a supervised classification algorithm, it consists of many individual decision trees that function as a group. Each tree in the random forest gives out a class prediction and the class with the most votes becomes the model's prediction. The accuracy of random forest depends on the number of trees, as the number of trees increase the accuracy increases.

Random Forests grow many classification trees. The input vector is put down each of the trees in the forest to classify a new object from an input vector. Each tree results in a classification and it is considered as a vote for that class. The algorithm chooses the classification with a maximum number of votes [57]

Following steps show the growth of each tree in the forest:

1. If N is the number of cases in the training set, N cases are sampled at random with replacement from the original data. This sample taken as the training set for growing the tree.
2. If there are M variables, a number m<<M is declared such that at each node, m variables are chosen at random out of the M and the best number on these m is used to split the node. The number m is kept constant during the forest growth
3. Each tree is grown to the maximum extent possible, there is no cutting.

The forest rate depends on two major attributes:

- One is the correlation between any two trees in the forest, forest error rate increases with increase in correlation.
- The other attribute is the strength of each tree in the forest. A tree with a low error rate is considered as a string classifier. Forest error rate decreases with increase in correlation.

The Random Forest algorithm can be stated as follows:

1. Draw ntree bootstrap samples from the original data

2. Grow an unpruned classification tree with the below modification for each of the bootstrap samples: Rather than choosing the best split at each node from all the features, sample mtry (number of features available for splitting) of the predictors and select the best split from those features.
3. Predict unknown data by grouping the predictions of the ntree trees, i.e., majority votes for classification.

An estimate of the error rate can be observed on the training data by the following:

1. At each bootstrap iteration, predict the data not in the bootstrap sample, which is also call out-of-bag (OOB) [58] using the grown tree with the bootstrap sample.

2. On an average, each data point would be out-of-bag around 36% of the times, so group these OOB predictions. Calculate the error rate, which is called the OOB estimate of error rate.

Chapter 4: Results and Evaluation

This chapter gives an overview of the achieved results, the data used, and the methods used to achieve the given result.

In this chapter, we present and analyze experimental results of various methods used to improve prediction accuracy and stability. This chapter is organized as follows: Section 4.1 focuses on the dataset we used. In section 4.2, we present baseline results obtained on imbalanced dataset Section 4.3 focuses on the results from our Improved Resampling method. In section 4.4, we discuss the results obtained after applying data fusion technique on the anthropometric and biomarker data. This section also summarizes the results from the classification methods employed on the combined data by fusing high impact features from anthropometry and biomarker data together.

Lastly, in section 4.2.4 we discuss the stability and improvement in the prediction accuracy obtained from the fusion mechanism developed by combining the features from anthropometry and biomarker data sets.

4.1 Dataset

In this work, we used National Health and Human Nutrition Examination Surveys (NHANES) 2003 – 2006 anthropometric and biomarker datasets. NHANES is a program of studies to assess the health and nutritional status of adults and children in the United States. This survey combines interviews and physical examinations of patients and presents their information in organized data files [5]. The anthropometric data is collected by interviewing each patient and its obtained directly from the NHANES and used in our work. The Biomarker data is a combination of the required biomarker readings obtained from different laboratory data files.

Table 4.1 summarizes information of both anthropometric and biomarker [5] National Health and Nutrition Examination Survey retrieved from https://www.cdc.gov/nchs/nhanes/index.htm

Dataset	Total no. of instances	No. of positively labelled Instances	No. of Negatively labelled Instances	No. of attributes	No. of classes
Biomarker	19214	410	18804	15	2
Anthropometry	19214	410	18804	18	2

Table 4.1: Summary of NHANES dataset

We randomly chose 15367 individuals for training and 3847 individuals for testing. All the methods are applied based on this partitioning of the dataset.

Table 4.2 and Table 4.3 show the attributes used in both anthropometry and biomarker datasets respectively.

Attribute	Average $\pm$ SD
Weight(kg) - BMXWT	75.49 ± 16.54
Height(cm) - BMXHT1	167.83 ± 10.14
Body Mass Index(kg/m^2) - BMXBMI	26.72 ± 4.95
Upper Arm Length(cm) - BMXARML	37.16 ± 2.75
Arm Circumference (cm) - BMXARMC	31.57 ± 4.19
Waist Circumference (cm) - BMXWAIST	93.56 ± 13.62
Triceps Skinfold (mm) - BMXTRI	17.92 ± 8.01
Subscapular Skinfold (mm) - BMXSUB	19.95 ± 7.80
Vertical Trunk Circumference(cm) - VTC	159.00 ± 10.28
Neck Circumference (cm) - NC	39.67 ± 2.70
A body Shape Index ($m^{11/6}kg^{-2/3}$) - ABSI	0.08 ± 0.01
Body Surface Area (cm^2) – BSA	18235.73 ± 2223.73
Surface - based body shape index - SBSI	0.12 ± 0.01
Waist-to-Height Ratio - WHtR	0.56 ± 0.08
BSA to VTC Ratio - BSAbyVTC	114.28 ± 6.73
VTC to NC Ratio - VTNR	4.01 ± 0.08
VTC to H Ratio - VTCbyHT	0.95 ± 0.05
VTC to WC Ratio - VTCbyWC	1.72 ± 0.18

Table 4.2: Anthropometry Attributes

Attribute	Average $\pm$ SD
Glycohemoglobin (%) - LBXGH	5.51 $\pm$ 0.90
Serum Albumin (g/dL) - LBXSAL	4.29 $\pm$ 0.37
Total Cholesterol(mg/dL) -LBXTC	196.58 $\pm$ 42.03
Serum urea nitrogen (mg/dL) - LBXSBU	13.14 $\pm$ 5.63
Serum Alkaline phosphatase (U/L) - LBXSAPSI	71.98 $\pm$ 26.50
Systolic blood pressure (mm Hg) - BPXSY1	123.99 $\pm$ 20.33
Diastolic Blood pressure (mm Hg) - BPXDI1	69.24 $\pm$ 13.55
Pulse (30 sec. pulse $\times$ 2) - BPXPLS	71.93 $\pm$ 12.36
Total cholesterol (mg/dL) - LBDHDL	196.58 $\pm$ 42.03
Hemoglobin (g/dL) - LBXHGB	14.31 $\pm$ 1.53
Lymphocyte percent (%) - LBXLYPCT	30.08 $\pm$ 8.64
White blood cell count (1000 cells/uL) - LBXWBCSI	71.9 $\pm$ 2.49
Hematocrit (%) - LBXHCT	42.05 $\pm$ 4.45
Red blood cell count (million cells/uL) - LBXRBCSI	4.68 $\pm$ 0.52
Platelet count (1000 cells/uL) - LBXPLTSI	259. 14 $\pm$ 67.33

Table 4.3: Biomarker Attributes

4.2 Baseline Results

In this section, we discuss the baseline results of classification methods on the imbalanced dataset without applying data preprocessing

4.2.1 Results without data imbalance treatment

In this section, the results for the imbalanced data sets of anthropometry and biomarker are discussed. The entire data samples of both the datasets were trained separately using SVM and Random Forest models. The accuracy of the models is shown in Table 4.4

Accuracy (%)	Anthropometry	Anthropometry - minority class (YES)	Biomarker	Biomarker – minority class (YES)
Random Forest	41.48	16.43	62.40	21.76
SVM	50.01	18.45	57.43	20.8

Table 4.4: Accuracy of imbalance data set

4.2.2 Results with SMOTE

In this section, the results using SMOTE over-sampling method for anthropometry and biomarker are discussed. The minority data samples are over-sampled using SMOTE which generates synthetic samples. The balanced dataset is trained using SVM and random forest classification models and tested those models on the test data. The accuracy of the models is shown in table 4.5.

Accuracy (%)	Anthropometry	Anthropometry – minority class (YES)	Biomarker	Biomarker – minority class (YES)
Random Forest	53.63	22.76	55.34	23.7
SVM	60.46	24.87	62.28	26.66

Table 4.5: Accuracy using SMOTE

4.3 Results with Improved Resampling

In this section, we summarize the results after applying outlier treatment and feature selection on anthropometry and biomarker data separately and then sub-sample the dataset using random resampling technique. The dataset is now, divided into 46 completely balanced sub-samples. We trained the classification models with each balanced subsample with different number of feature combinations based on feature importance. We then used these 46 trained models to predict kidney failure risk for test data.

Figure 4.1 shows the average prediction accuracy of all 46 predictors for biomarker data based on number of features used. After analyzing the results based on the features and stability, the prediction results of all predictors using 6 features are stable.

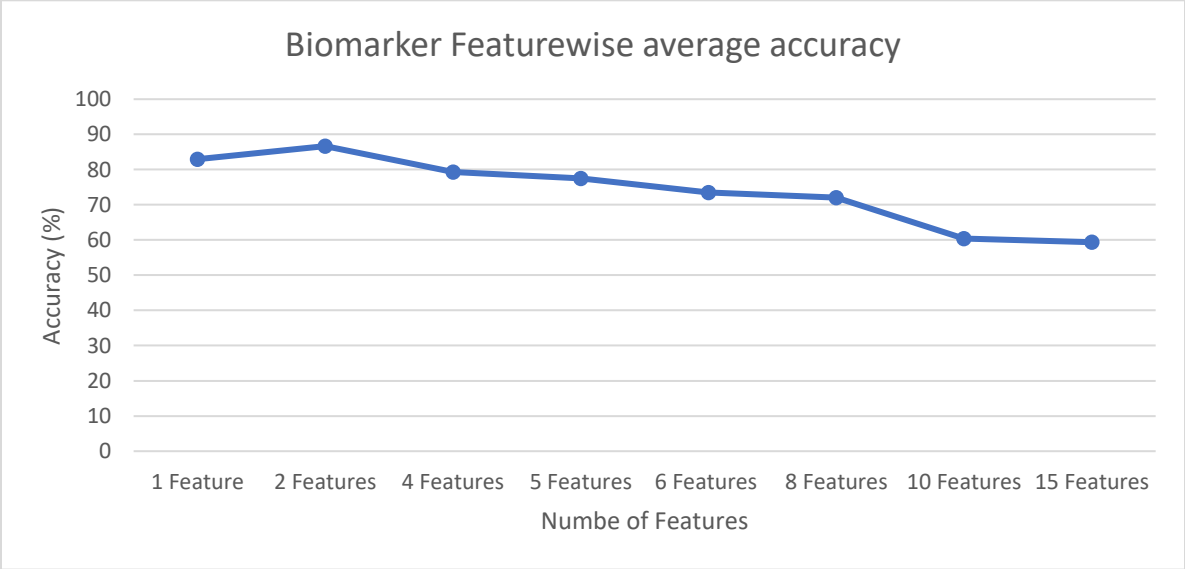

Figure 4.1: Average prediction accuracy for biomarker data based on number of features

Figure 4.2 shows the average prediction accuracy of all 46 predictors for anthropometry data based on number of features used. From the results, it can be observed that the average prediction accuracy using 6 features is better than other results.

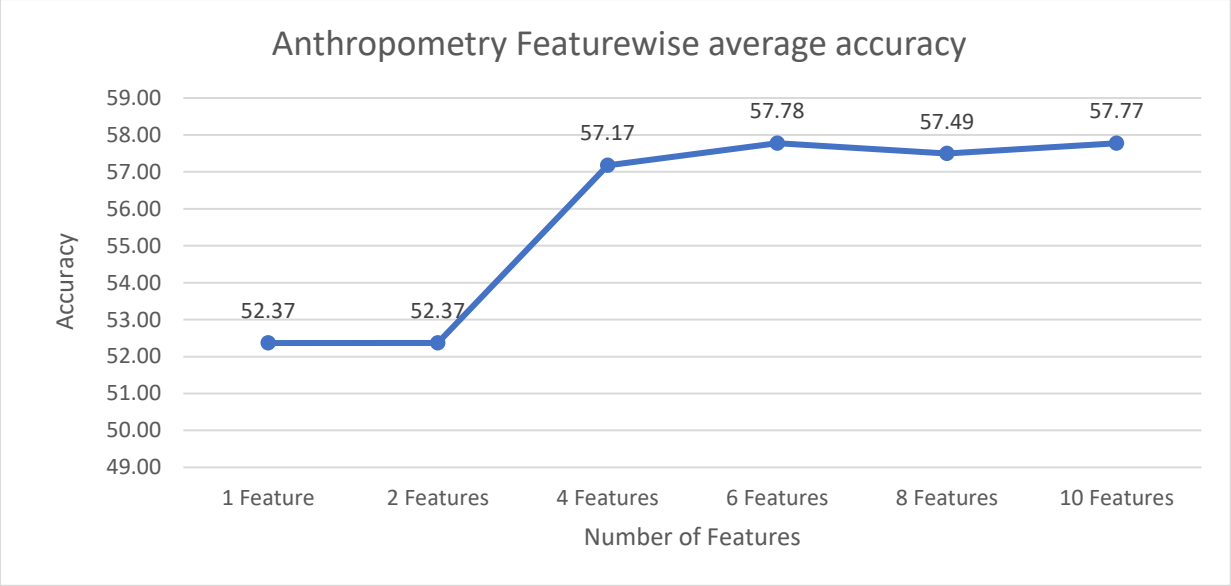

Figure 4.2: Average prediction accuracy for anthropometry data based on number of features

Based on the prediction results obtained on test dataset using different number of features, we obtained better accuracy and stable results using 6 features. The detailed results for anthropometry and biomarker with 6 selected features are discussed below. Figure 4.3 and Figure 4.4 show the accuracy of each predictor based on predictor ranking using Random Forest model.

The accuracy of all the predictors is not consistent in both anthropometry and biomarker data. In anthropometry, the top predictor's accuracy is 66.34% and the accuracy drops significantly for all the other predictors. We calculated the average result of all the predictors by taking prediction result based on majority vote result for everyone from all the predictor results.

Figure 4.5 and figure 4.6 show the results based on above method for all predictors, top predictor, top 5, top 10 and top 15 predictors. We also calculate the standard deviation (SD) for each average to see which of these predictor group is more reliable.

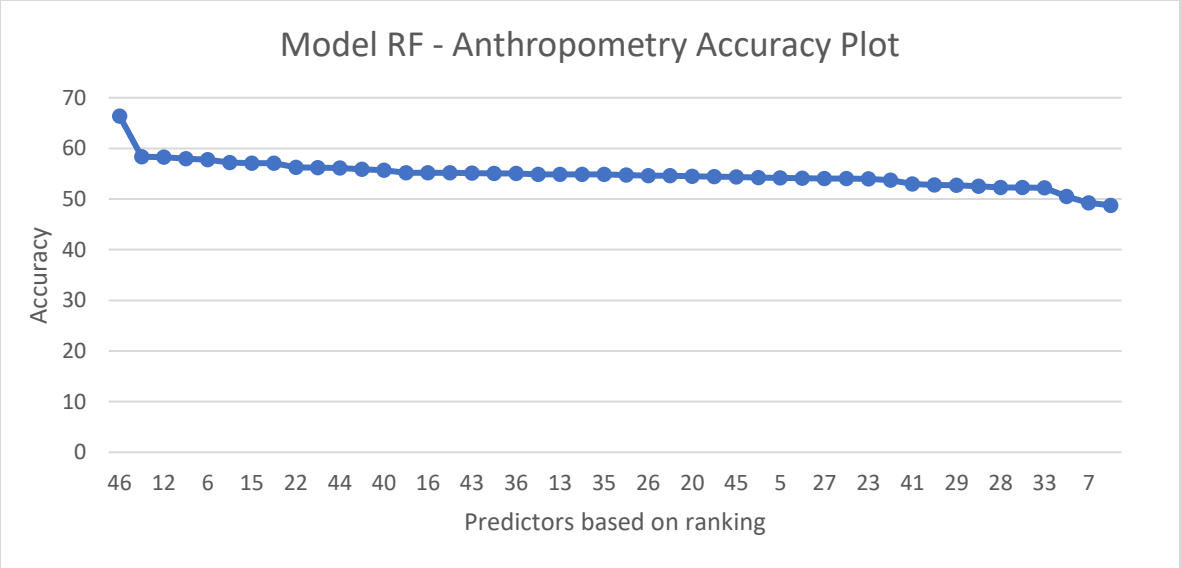

Figure 4.3: Random Forest – prediction accuracy for anthropometry

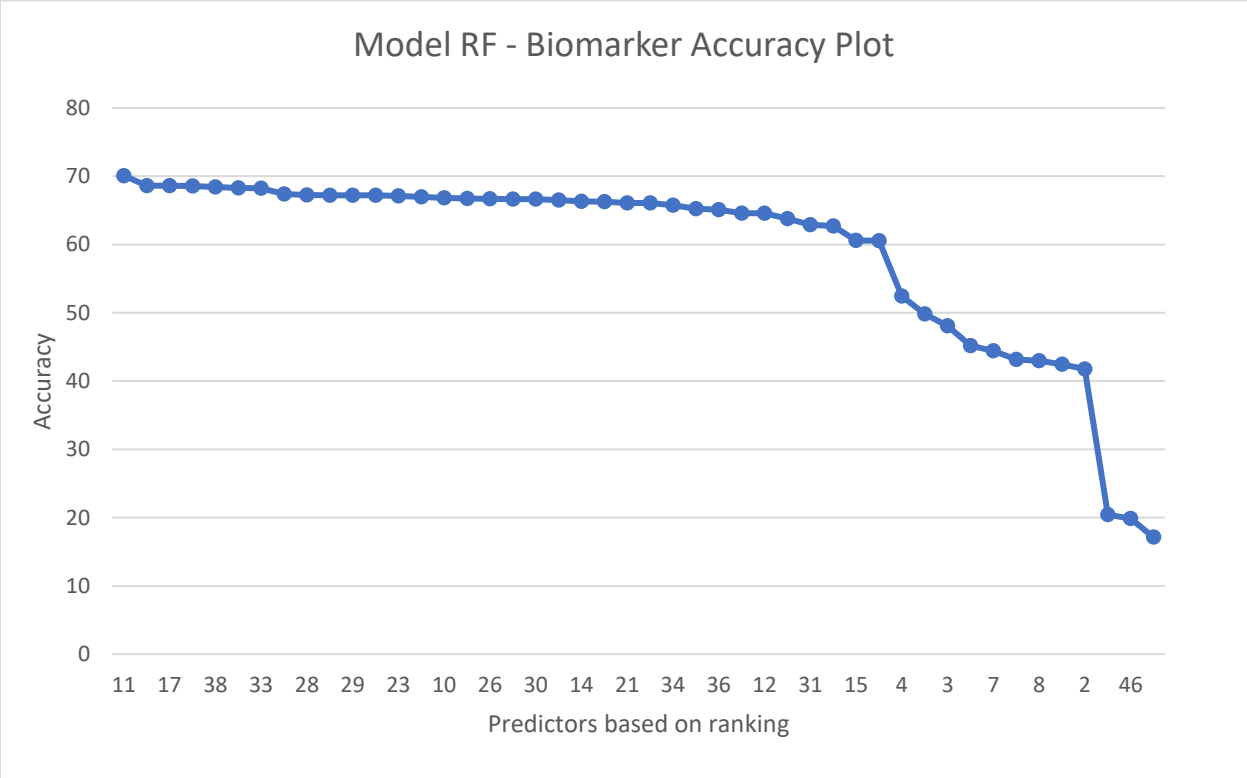

Figure 4.4: Random Forest – prediction accuracy for biomarker

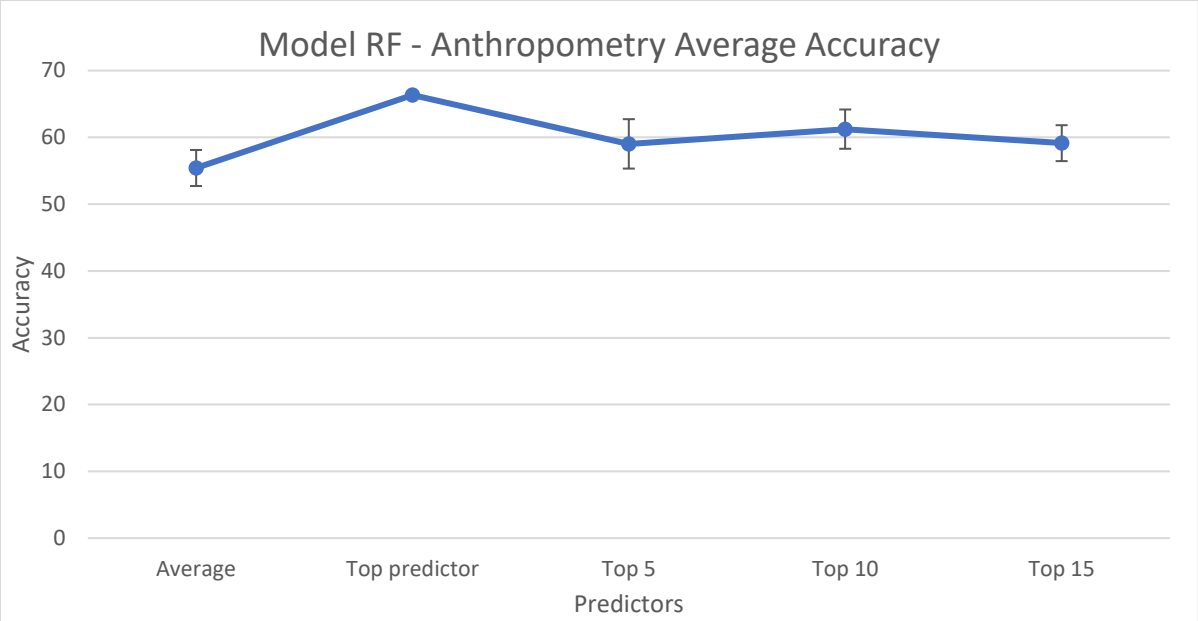

Figure 4.5: Anthropometry - Average accuracy for set of predictor groups

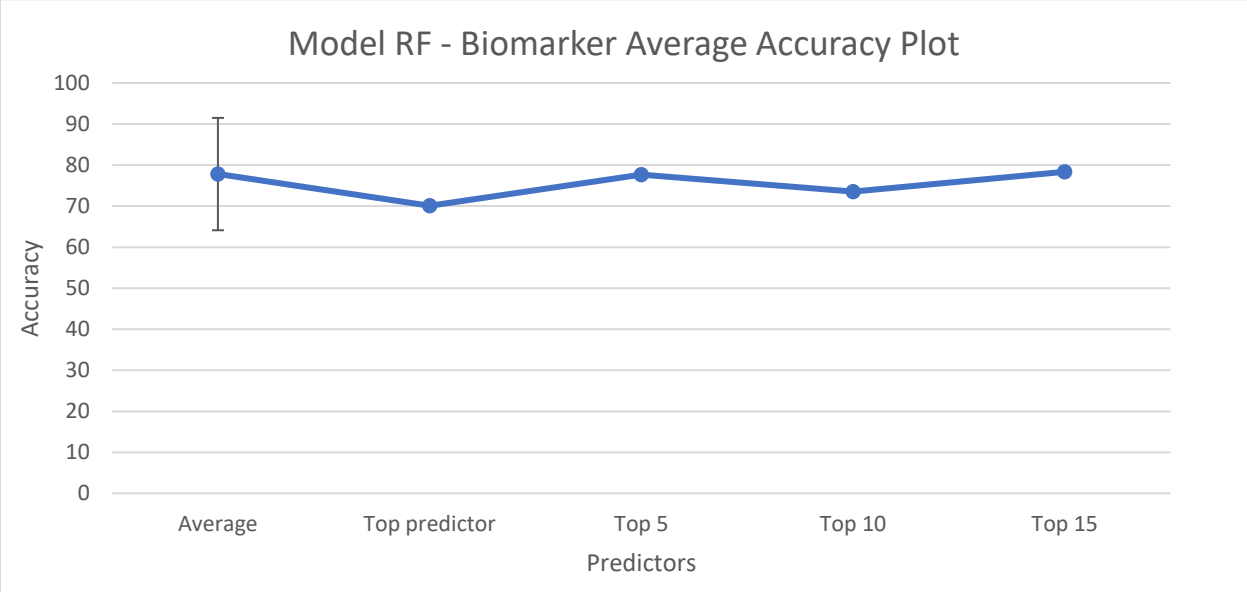

Figure 4.6: Biomarker - Average accuracy for a set of predictor groups

Table 4.4 shows the accuracy with SD values; biomarker accuracy is consistent among all the groups, but the SD is very high when the overall average is concerned which makes it less stable. However, the prediction accuracy of anthropometry is less than 60% but the SD is considerably low making it more reliable.

Random Forest Model	Average of all predictors ± SD	Top predictor	Top 5 predictors ± SD	Top 10 predictors ±SD	Top 15 predictors ±SD
Anthropometry	55.42± 2.7	66.34	59.03± 3.7	61.24± 2.94	59.14± 2.69
Biomarker	77.80 ± 13.69	70.08	77.67 ± 0.69	73.51 ± 0.85	78.35 ± 0.90

Table 4.6: Model RF - Average accuracy with SD

Random Forest (YES %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Anthropometry	67.86	70.24	54.76	67.86	69.05
Biomarker	53.57	72.62	53.57	52.38	52.38

Table 4.7: Model RF: Accuracy for minority (YES) class

Random Forest (NO %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Anthropometry	55.14	66.91	59.13	61.09	58.92
Biomarker	78.34	70.18	78.21	73.98	78.93

Table 4.8: Model RF: Accuracy for majority (NO) class

Figure 4.7 and Figure 4.8 show the prediction accuracy of all the predictors using SVM model, the series is based on predictor ranking. The top predictor for anthropometry gives an accuracy of 78.27%, the accuracy of most of the predictors in the set is 60% and below. On the other hand, the top predictor for biomarker gives an accuracy of 80.43%, most of the predictor accuracy is in between 60% to 80%. The average accuracy for SVM model is calculated for both anthropometry and biomarker data with 6 selected features.

Figure 4.9 and Figure 4.10 show the average accuracy of SVM model for anthropometry and biomarker data separately.

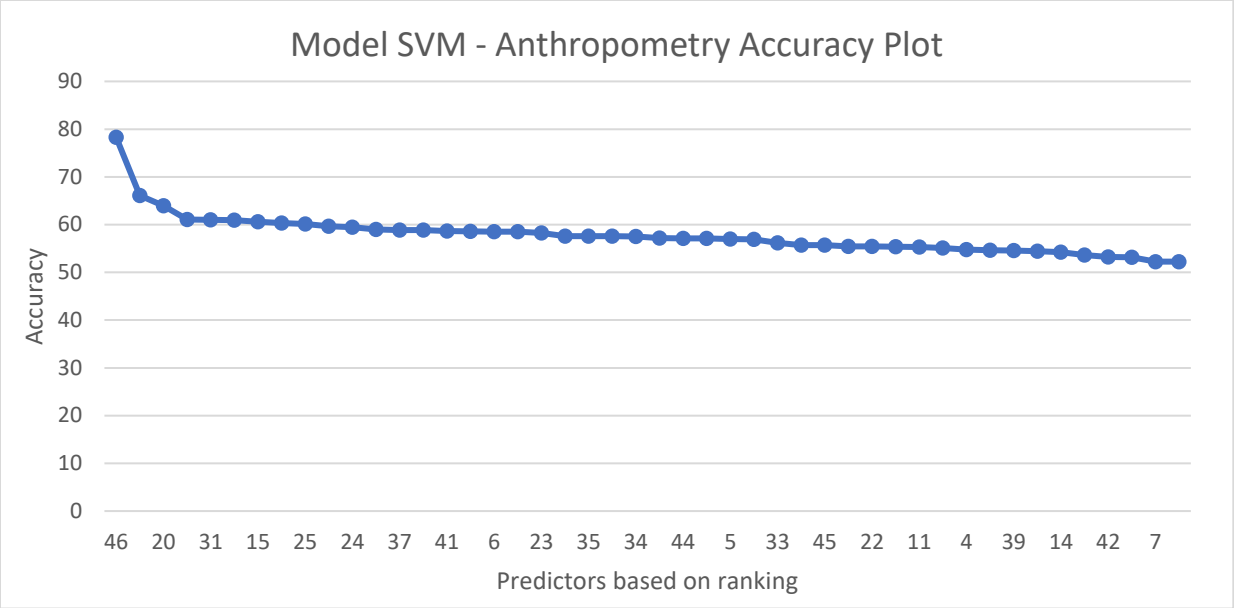

Figure 4.7: SVM – prediction accuracy for anthropometry

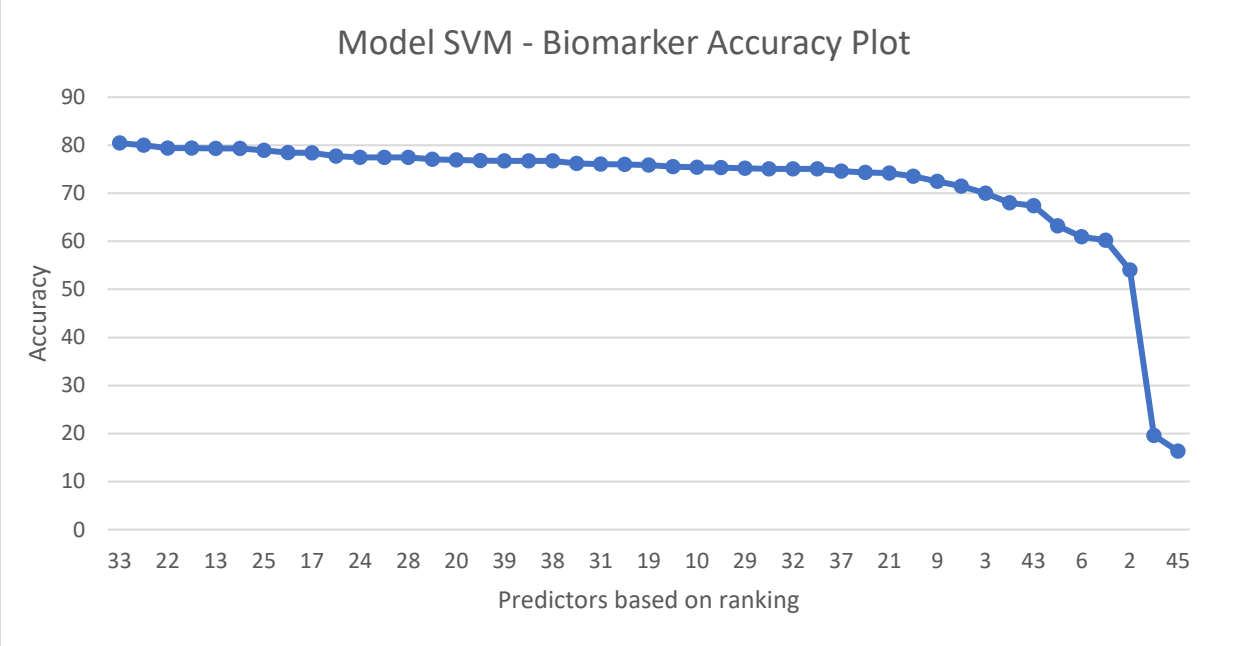

Figure 4.8: SVM – prediction accuracy for anthropometry

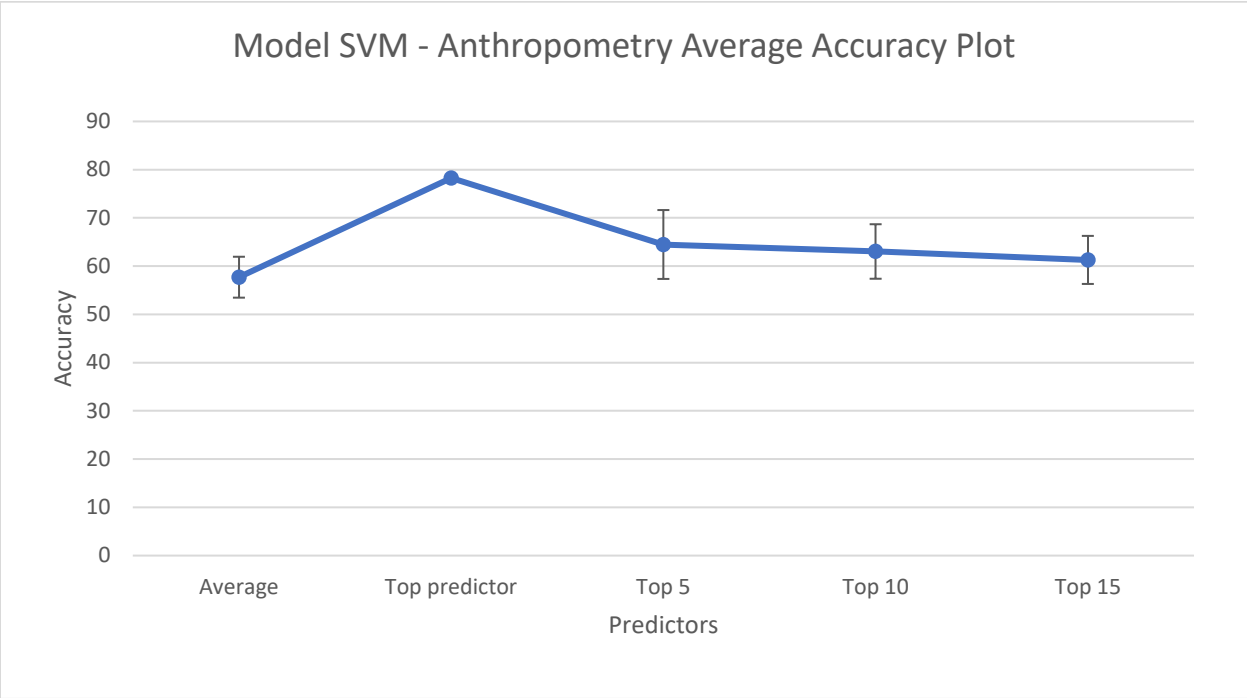

Figure 4.9: Anthropometry **-** Average accuracy for set of predictor groups

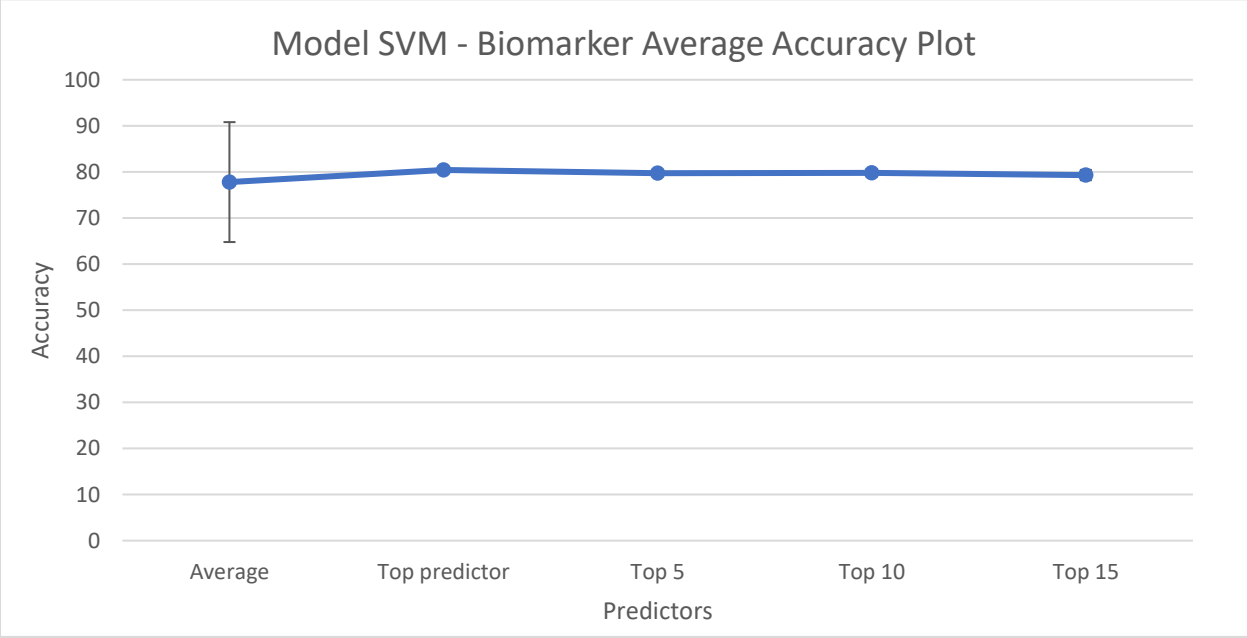

Figure 4.10*:* Anthropometry **-** Average accuracy for set of predictor groups

Table 4.5 shows the average accuracy with SD values; biomarker accuracy is consistent among all the groups, but the SD is very high when the overall average is concerned which makes it less stable. Apart from the that, the top 5, top 10, top 15 predictors accuracy is stable with

considerable SD value. However, the average prediction accuracy of anthropometry is less than 60% and SD is also equally same for all the predictors.

SVM Model	Average of all predictors ± SD	Top predictor	Top 5 predictors ± SD	Top 10 predictors ±SD	Top 15 predictors ±SD
Anthropometry	57.71 ± 4.24	78.27	64.49 ± 7.14	63.04 ± 5.65	61.29 ±4.98
Biomarker	77.80 ± 13.01	80.43	79.75 ± 0.48	79.78 ± 0.79	79.33 ± 1.11

Table 4.9: Model SVM - Average accuracy with SD

SVM (YES %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Anthropometry	61.90	70.24	50	60	58.33
Biomarker	53.57	76.19	50	51.19	51.19

Table 4.10: Model SVM - Accuracy for minority (YES) class

SVM (NO %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Anthropometry	57.61	79.35	64.82	63.11	61.36
Biomarker	78.34	81	80.41	80.41	79.96

Table 4.11: Model SVM - Accuracy for majority (YES) class

4.4 Results with Data Fusion

In this section, we discuss the prediction results achieved using data fusion methods applied on the balanced dataset. Random Forest is trained on training dataset with selected features of anthropometry and biomarker. The prediction accuracy of each predictor group using Random Forest model on test dataset with feature – level fusion is shown in Figure 4.11

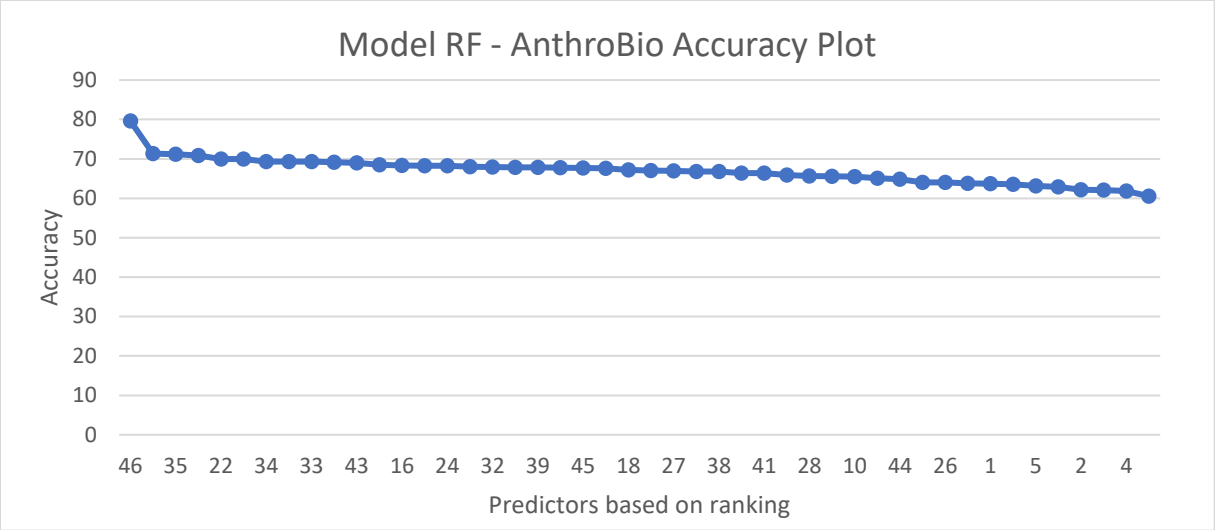

Figure 4.11: Random Forest – prediction accuracy for feature-level fusion

Figure 4.12 gives the prediction accuracy of each predictor group using SVM model with feature – level fusion.

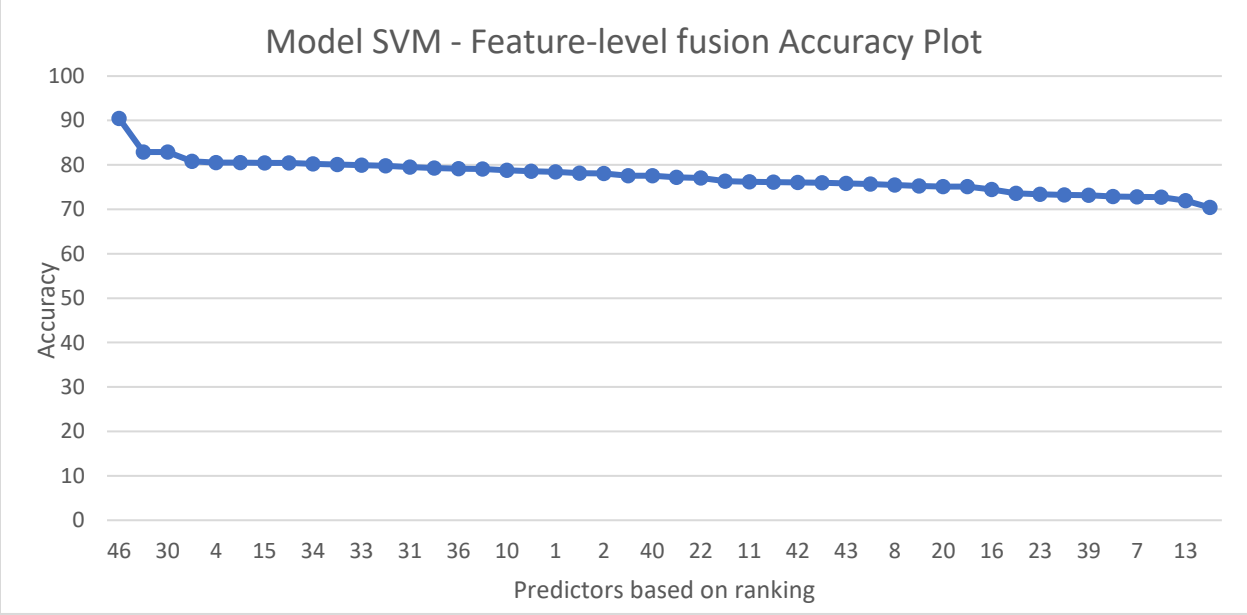

Figure 4.12: SVM – prediction accuracy for feature-level fusion

The average prediction accuracy of top predictors using Random Forest and SVM models for feature-level fusion are shown in Figure 4.13 and Figure 4.14.

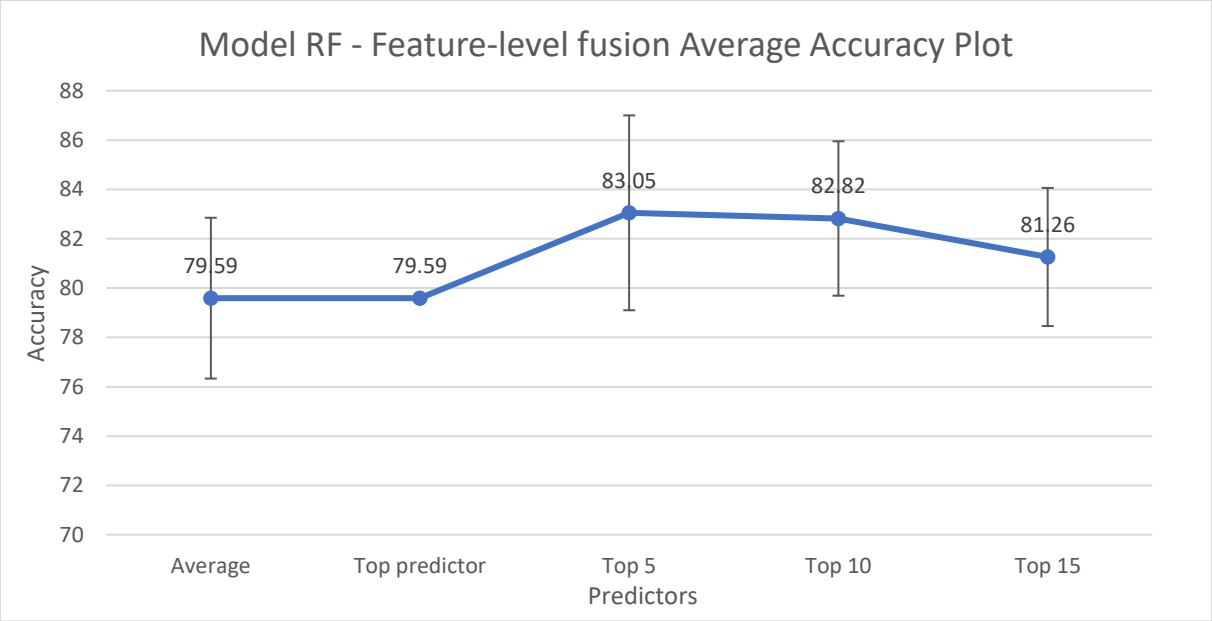

Figure 4.13: Random Forest – Average accuracy for a set of predictor groups

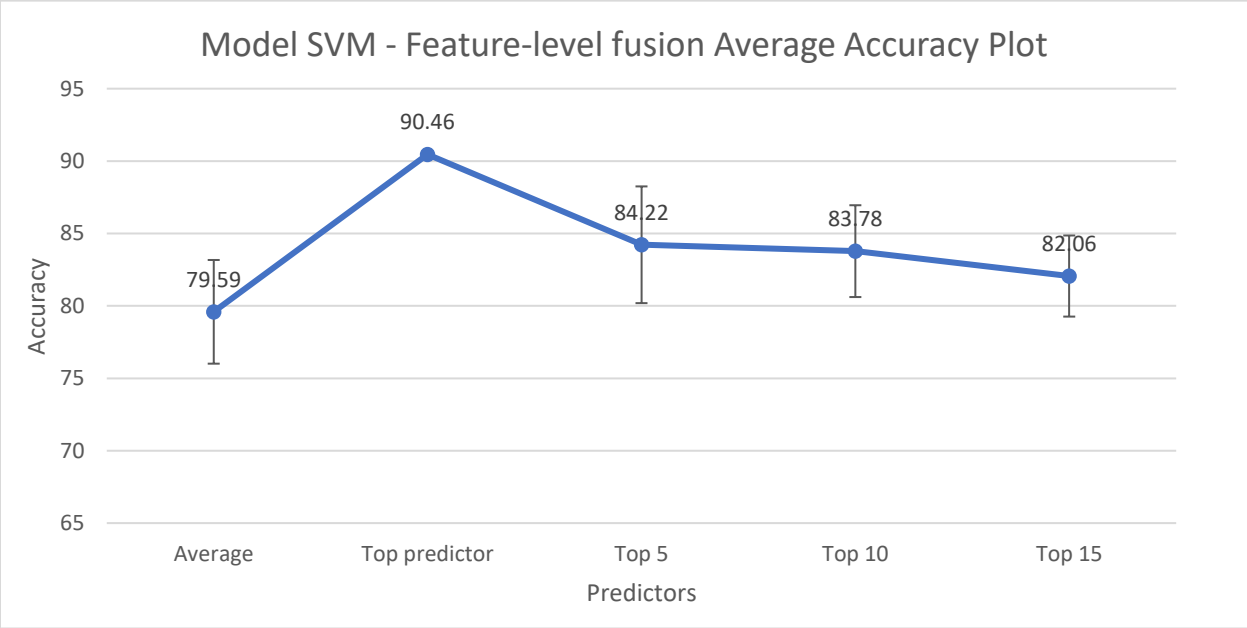

Figure 4.14: SVM Model – Average accuracy for a set of predictor groups

Table 4.6 gives the summary of average prediction results obtained from Random Forest and SVM models with SD. The top predictor using SVM model gives the highest accuracy of 90.46% over all the other predictors. However, the average accuracy is consistent over all the groups considered and SD about the same for both the models.

Feature-level fusion	**Average of all predictors ± SD**	**Top predictor**	**Top 5 predictors ± SD**	**Top 10 predictors ±SD**	**Top 15 predictors ±SD**
Random Forest	79.59 ± 3.26	79.59	83.05± 3.95	82.82± 3.13	81.26± 2.80
SVM	79.59 ± 3.58	90.46	84.22 ± 4.03	83.78 ± 3.17	82.06 ± 2.80

Table 4.12: Feature-level fusion - Average accuracy with SD

Figure 4.15 summarizes the average and top predictor prediction accuracy of all combination of datasets using SVM model.

Feature-level fusion (YES %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Random Forest	53.57	73.81	51.19	52.38	51.19
SVM	53.57	65.48	51.19	52.38	52.38

Table 4.13: Accuracy for minority (YES) class

Feature-level fusion (NO %)	Average of all predictors	Top predictor	Top 5 predictors	Top 10 predictors	Top 15 predictors
Random Forest	80.18	80.31	83.76	83.50	81.93
SVM	80.18	91.58	84.96	84.48	82.73

Table 4.14: Accuracy for majority (YES) class

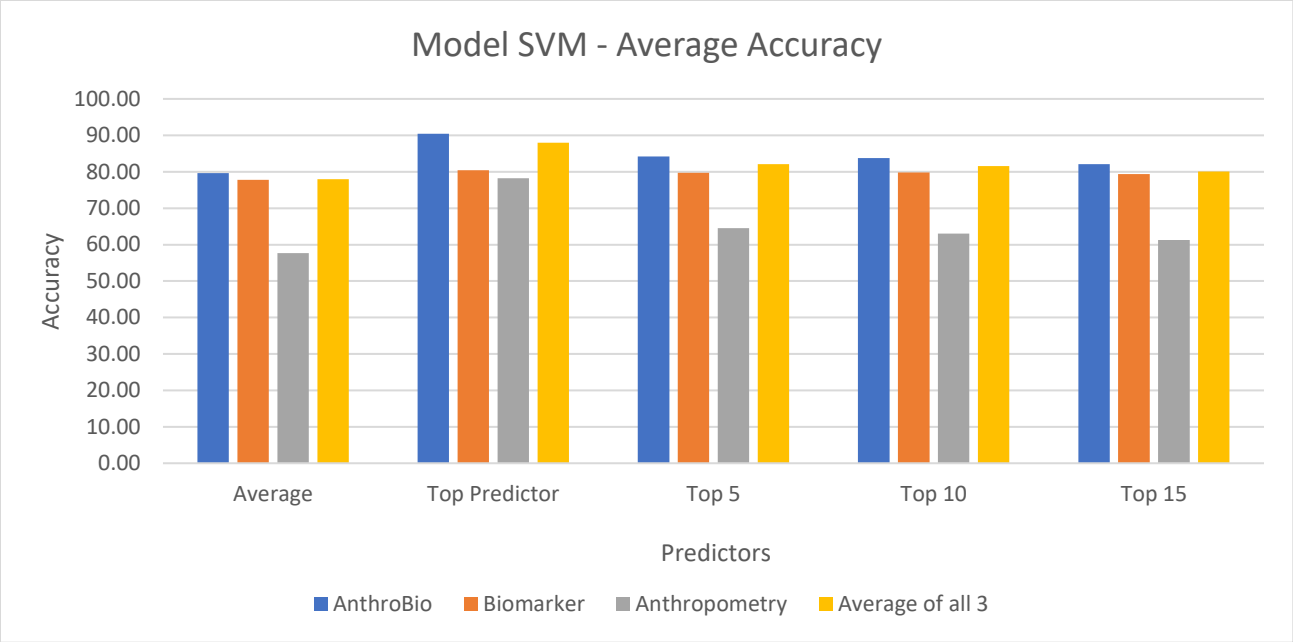

Figure 4.15: Model SVM- Average accuracy for feature-level fusion, biomarker, anthropometry and decision-level fusion

Table 4.15 gives the overall average prediction accuracy with SD. Feature-level fusion technique gives better prediction accuracy over other techniques used. However, the SD is high when compared to the results obtained from biomarker data. Anthropometry gives the least prediction accuracy, but the accuracy is improved when its features are fused with biomarker.

SVM	**Feature-level fusion**	**Biomarker**	**Anthropometry**	**Average of all 3 (Majority Vote)**
Average $\pm$ SD	79.59 $\pm$ 3.58	77.80 $\pm$ 13.01	57.71 $\pm$ 4.24	77.98
Top Predictor	90.46	80.43	78.27	87.99
Top 5 $\pm$ SD	84.22 $\pm$ 4.03	79.75 $\pm$ 0.48	64.49 $\pm$ 7.14	82.06
Top 10 $\pm$ SD	83.78 $\pm$ 3.17	79.78 $\pm$ 0.79	63.04 $\pm$ 5.65	81.57
Top 15 $\pm$ SD	82.06 $\pm$ 2.80	79.33 $\pm$ 1.11	61.29 $\pm$ 4.98	80.09

Table 4.15*:* Model SVM **-** Average accuracy of data set combinations

Figure 4.16 summarizes the average and top predictor prediction accuracy of all combination of datasets using Random Forest model.

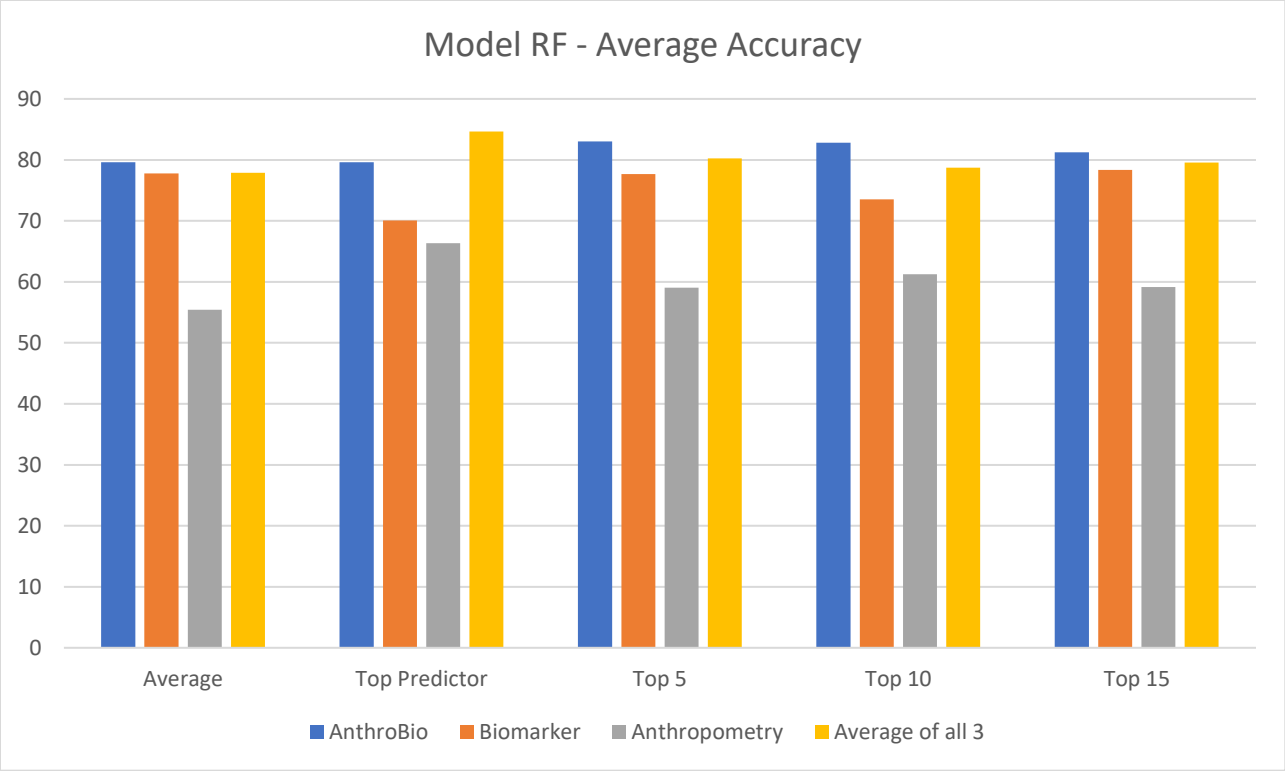

Figure 4.16: Model RF- Average accuracy for feature-level fusion, biomarker, anthropometry and decision-level fusion

Table 4.16 gives the overall average prediction accuracy with SD using Random Forest. The accuracy trend is similar to SVM model, with highest accuracy for feature-level fusion technique over other techniques.

Random Forest	**Feature-level fusion**	**Biomarker**	**Anthropometry**	**Average of all 3**
Average $\pm$ SD	79.59 $\pm$ 3.26	77.80 $\pm$ 13.69	55.42 $\pm$ 2.7	77.88
Top Predictor	79.59	70.08	66.34	84.66
Top 5 $\pm$ SD	83.05 $\pm$ 3.95	77.67 $\pm$ 0.69	59.03 $\pm$ 3.7	80.24
Top 10 $\pm$ SD	82.82 $\pm$ 3.13	73.51 $\pm$ 0.85	61.24 $\pm$ 2.94	78.74
Top 15 $\pm$ SD	81.26 $\pm$ 2.8	78.35 $\pm$ 0.9	59.14 $\pm$ 2.69	79.54

Table 4.16: Model RF - Average accuracy of data set combinations

Figure 4.17 shows the percentage of test data samples which fall under the groups shown in table 4.9.

Groups	Classification Models
Anthropometry (0) – Biomarker (0)	Both the datasets predicted incorrectly
Anthropometry (0) – Biomarker (1)	anthropometry predicted incorrectly – biomarker predicted correctly
Anthropometry (1) – Biomarker (0)	anthropometry predicted correctly – biomarker predicted correctly
Anthropometry (1) – Biomarker (1)	Both the datasets predicted correctly

Table 4.17: Groups based on prediction results

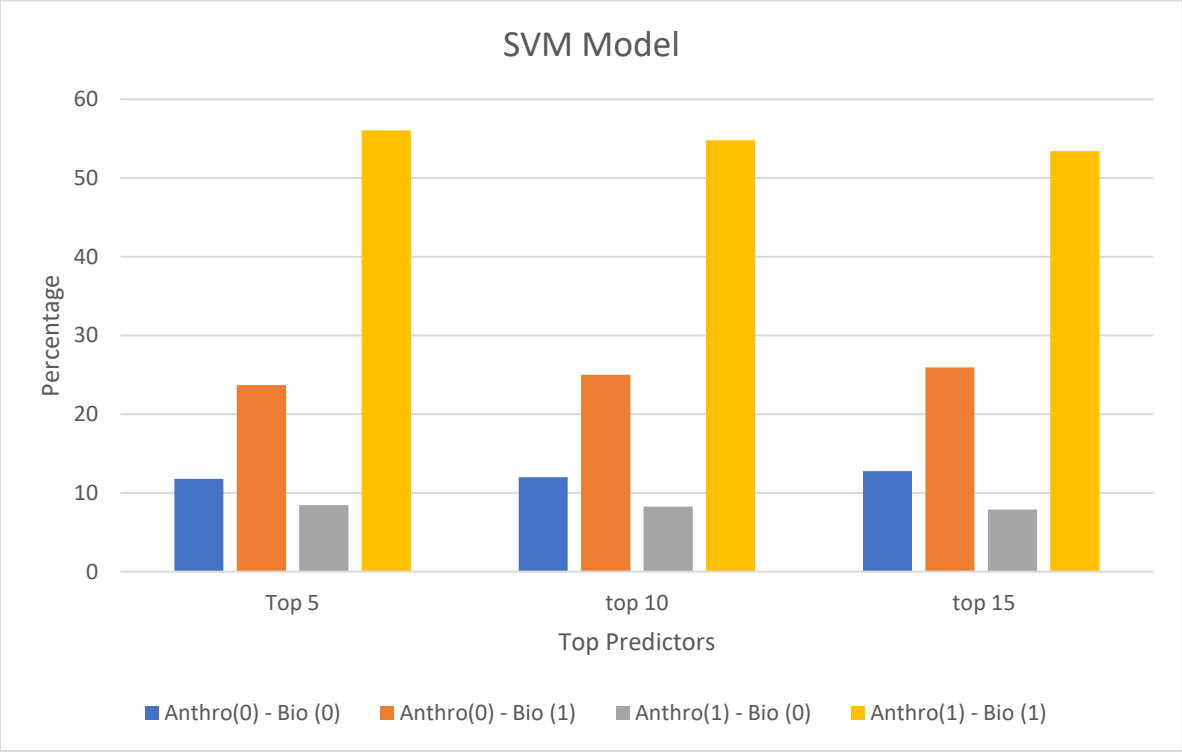

Figure 4.17: Percentage of data samples in each predictor group

	Top 5	top 10	top 15
Anthropometry (0) - Biomarker (0)	11.8014	11.98336	12.76319
Anthropometry (0) - Biomarker (1)	23.70678	24.9805	25.94229
Anthropometry (1) - Biomarker (0)	8.448141	8.240187	7.902262
Anthropometry (1) - Biomarker (1)	56.04367	54.79594	53.39225

Table 4.18: Percentage of data samples in each group

4.5 Discussion

In our implementation, biomarker feature space gave a better prediction results over anthropometry feature space after applying improved data balancing techniques. However, feature-level fusion improved the results which shows the importance of feature space in prediction methods. Further, we observed that the majority vote method showed the results could be better if data fusion is carried out in an intelligent way.

Chapter 5: Conclusion and Future work

In our study, we have addressed the problem of data imbalance in prediction of kidney ailments. We studied different techniques to overcome data imbalance problem and proposed a new improved resampling method to balance the anthropometric and biomarker data obtained from NHANES dataset, then applied machine learning techniques to predict kidney ailments on the balanced dataset.

The results from the classification models are analyzed to discover certain patterns in the prediction accuracy for different combination of features. We employed feature-level fusion and decision-level fusion techniques to analyze the performance of the classification models on the test data. Further, we fused both feature-level and decision-level prediction results to select the best predictor group which could be used to predict unknown data samples. The feature-level fusion prediction accuracy for SVM model was better than the results obtained from individual feature sets of anthropometry and biomarker.

The feature-level fusion technique performed better when compared with the prediction models of individual data set results. The standard deviation shows the consistency of the results over all the predictors, which shows the performance stability and reliability. Our results using intelligent data fusion show more than 50% of data samples being predicted accurately by both anthropometry and biomarker dataset. Even though, this result is not dependable, there is scope to improve the data fusion methods further**.**

In future, we intend to improve the intelligent data fusion method to efficiently predict kidney ailments in patients. We would also increase the feature space by incorporating features like age, sex, demographics which might give interesting patterns of results. We would further apply these proposed methods to other disease predictions and look for better ways to incorporate feature space for prediction.

References

[1] National Heart, Lung, Blood Institute, National Institute of Diabetes, Digestive and Kidney Diseases (US), 1998. Clinical guidelines on the identification, evaluation, and treatment of overweight and obesity in adults: the evidence report (No. 98). National Heart, Lung, and Blood Institute.

[2] Pasadana, I.A., Hartama, D., Zarlis, M., Sianipar, A.S., Munandar, A., Baeha, S. and Alam, A.R.M., 2019, August. Chronic Kidney Disease Prediction by Using Different Decision Tree Techniques. In Journal of Physics: Conference Series (Vol. 1255, No. 1, p. 012024). IOP Publishing..

[3] Sarnak, M.J., Levey, A.S., Schoolwerth, A.C., Coresh, J., Culleton, B., Hamm, L.L., McCullough, P.A., Kasiske, B.L., Kelepouris, E., Klag, M.J. and Parfrey, P., 2003. Kidney disease as a risk factor for development of cardiovascular disease: a statement from the American Heart Association Councils on Kidney in Cardiovascular Disease, High Blood Pressure Research, Clinical Cardiology, and Epidemiology and Prevention. Circulation, 108(17), pp.2154-2169.

[4] Jha, V., Garcia-Garcia, G., Iseki, K., Li, Z., Naicker, S., Plattner, B., Saran, R., Wang, A.Y.M. and Yang, C.W., 2013. Chronic kidney disease: global dimension and perspectives. The Lancet, 382(9888), pp.260-272.

[5] Hansen, M.M., Miron-Shatz, T., Lau, A.Y.S. and Paton, C., 2014. Big data in science and healthcare: a review of recent literature and perspectives. Yearbook of medical informatics, 23(01), pp.21-26.

[6] Interoperability, N.B.D., 2015. NIST Big Data Public Working Group Definitions and Taxonomies Subgroup In: Framework: Definitions. NIST Special Publication, pp.1500-1.

[7] Mehta, N. and Pandit, A., 2018. Concurrence of big data analytics and healthcare: A systematic review. International journal of medical informatics, 114, pp.57-65.

[8] Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H. and Wang, Y., 2017. Artificial intelligence in healthcare: past, present and future. Stroke and vascular neurology, 2(4), pp.230-243.

[9] Zeng, X.X., Liu, J., Ma, L. and Fu, P., 2018. Big Data Research in Chronic Kidney Disease. Chinese medical journal, 131(22), p.2647.

[10] Couser, W.G., Remuzzi, G., Mendis, S. and Tonelli, M., 2011. The contribution of chronic kidney disease to the global burden of major noncommunicable diseases. Kidney international, 80(12), pp.1258-1270.

[11] Incontinence & Overactive Bladder Health Center, Retrieved from https://www.webmd.com/urinary-incontinence-oab/default.htm

[12] Kadatz, M.J., Lee, E.S. and Levin, A., 2016. Predicting progression in CKD: perspectives and precautions. American Journal of Kidney Diseases, 67(5), pp.779-786.

[13] Major, R.W., Shepherd, D., Medcalf, J.F., Xu, G., Gray, L.J. and Brunskill, N.J., 2019. The Kidney Failure Risk Equation for prediction of end stage renal disease in UK primary care: An external validation and clinical impact projection cohort study. PLoS medicine, 16(11).

[14] Royston, P. and Altman, D.G., 2013. External validation of a Cox prognostic model: principles and methods. BMC medical research methodology, 13(1), p.33.

[15] Steyerberg, E.W., Moons, K.G., van der Windt, D.A., Hayden, J.A., Perel, P., Schroter, S., Riley, R.D., Hemingway, H., Altman, D.G. and PROGRESS Group, 2013. Prognosis Research Strategy (PROGRESS) 3: prognostic model research. PLoS medicine, 10(2), p.e1001381.

[16] Riley, R.D., Ensor, J., Snell, K.I., Debray, T.P., Altman, D.G., Moons, K.G. and Collins, G.S., 2016. External validation of clinical prediction models using big datasets from e-health records or IPD meta-analysis: opportunities and challenges. bmj, 353, p.i3140..

[17] Tangri, N., Grams, M.E., Levey, A.S., Coresh, J., Appel, L.J., Astor, B.C., Chodick, G., Collins, A.J., Djurdjev, O., Elley, C.R. and Evans, M., 2016. Multinational assessment of accuracy of equations for predicting risk of kidney failure: a meta-analysis. Jama, 315(2), pp.164-174.

[18] Jena, L. and Kamila, N.K., 2015. Distributed data mining classification algorithms for prediction of chronic-kidney-disease. Int. J. Emerg. Res. Manag. &Technology, 9359(11), pp.110-118.

[19] Chatterjee, S., Banerjee, S., Basu, P., Debnath, M. and Sen, S., 2017, April. Cuckoo search coupled artificial neural network in detection of chronic kidney disease. In 2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech) (pp. 1-4). IEEE.

[20] Chen, Z., Zhang, Z., Zhu, R., Xiang, Y. and Harrington, P.B., 2016. Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers. Chemometrics and Intelligent Laboratory Systems, 153, pp.140-145.

[21] Lakshmi, K.R., Nagesh, Y. and Krishna, M.V., 2014. Performance comparison of three data mining techniques for predicting kidney dialysis survivability. International Journal of Advances in Engineering & Technology, 7(1), p.242.

[22] 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, pp.321-357.

[23] Tahir, M.A., Kittler, J., Mikolajczyk, K. and Yan, F., 2009, June. A multiple expert approach to the class imbalance problem using inverse random under sampling. In International Workshop on Multiple Classifier Systems (pp. 82-91). Springer, Berlin, Heidelberg.

[24] Zhou, L., 2013. Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. Knowledge-Based Systems, 41, pp.16-25.

[25] Lu, J., Zhang, C. and Shi, F., 2016, August. A Classification Method of Imbalanced Data Base on PSO Algorithm. In International Conference of Pioneering Computer Scientists, Engineers and Educators (pp. 121-134). Springer, Singapore.

[26] Napierala, K. and Stefanowski, J., 2016. Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems, 46(3), pp.563-597.

[27] Yijing, L., Haixiang, G., Xiao, L., Yanan, L. and Jinling, L., 2016. Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. Knowledge-Based Systems, 94, pp.88-104.

[28] Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.

[29] Saeys, Y., Inza, I. and Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. bioinformatics, 23(19), pp.2507-2517.

[30] Motoda, H. and Liu, H., 2002. Feature selection, extraction and construction. Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol, 5(67-72), p.2.

[31] Hartmann, W.M., 2004, June. Dimension reduction vs. variable selection. In International Workshop on Applied Parallel Computing (pp. 931-938). Springer, Berlin, Heidelberg.

[32] Casañola-Martin, G., Garrigues, T., Bermejo, M., González-Álvarez, I., Nguyen-Hai, N., Cabrera-Pérez, M.Á. and Le-Thi-Thu, H., 2016. Exploring different strategies for imbalanced ADME data problem: case study on Caco-2 permeability modeling. Molecular diversity, 20(1), pp.93-109.

[33] Zhang, D., Ma, J., Yi, J., Niu, X. and Xu, X., 2015, August. An ensemble method for unbalanced sentiment classification. In 2015 11th International Conference on Natural Computation (ICNC) (pp. 440-445). IEEE.

[34] Lima, R.F. and Pereira, A.C.M., 2015, December. A fraud detection model based on feature selection and undersampling applied to Web payment systems. In 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) (Vol. 3, pp. 219-222). IEEE.

[35] Ghazikhani, A., Monsefi, R. and Yazdi, H.S., 2013. Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams. Neural computing and applications, 23(5), pp.1283-1295.

[36] Krawczyk, B., Woźniak, M. and Schaefer, G., 2014. Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing, 14, pp.554-562.

[37] Castro, C.L. and Braga, A.P., 2013. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. IEEE transactions on neural networks and learning systems, 24(6), pp.888-899.

[38] Juszczak, P., 2006. Learning to recognise: A study on one-class classification and active learning. [39] Tax, D.M. and Duin, R.P., 2001. Uniform object generation for optimizing one-class classifiers. Journal of machine learning research, 2(Dec), pp.155-173.

[40] Liu, B., Lee, W.S., Yu, P.S. and Li, X., 2002, July. Partially supervised classification of text documents. In ICML (Vol. 2, pp. 387-394).

[41] Mordelet, F. and Vert, J.P., 2011. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. BMC bioinformatics, 12(1), p.389.

[42] Bhardwaj, N., Gerstein, M. and Lu, H., 2010. Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique. BMC bioinformatics, 11(1), p.S6.

[43] Gieseke, F., Airola, A., Pahikkala, T. and Kramer, O., 2014. Fast and simple gradient-based optimization for semi-supervised support vector machines. Neurocomputing, 123, pp.23-32.

[44] Lee, J.S., Grunes, M.R., Ainsworth, T.L., Du, L.J., Schuler, D.L. and Cloude, S.R., 1999. Unsupervised classification using polarimetric decomposition and the complex Wishart classifier. IEEE Transactions on Geoscience and Remote Sensing, 37(5), pp.2249-2258. [45] Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. InIcml 1997 Jul 8 (Vol. 97, pp. 179-186).

[46] Gao, T., Hybrid classification approach for imbalanced datasets (Unpublished doctoral dissertation), IOWA State University

[47] Mishra, D. and Soni, D., 2018. Outliers in Data Mining: Approaches and Detection. International Journal of Engineering & Technology, 7(4.39), pp.189-198.

[48] Gentleman, J.F. and Wilk, M.B., 1975. Detecting outliers. II. Supplementing the direct analysis of residuals. Biometrics, pp.387-410.

[49] Abusamra, H., A Comparative Study of Feature Selection and Classification Methods for Gene Expression Data (Unpublished master's thesis), King Abdullah University of Science and Technology

[50] Seo, H.S. and Yoon, M., 2011. Outlier detection using support vector machines. Communications for Statistical Applications and Methods, 18(2), pp.171-177.

[51] Burke, S., 1998. Missing values, outliers, robust statistics & non-parametric methods. Scientific Data Management, 1, pp.32-38.

[52] Cook, R.D., 1979. Influential observations in linear regression. Journal of the American Statistical Association, 74(365), pp.169-174.

[53] Introduction to Support Vector Machines August 10, 2013. retrieved from http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm. html

[54] Bottou, L. and Lin, C.J., 2007. Support vector machine solvers large scale kernel machines.

[55] Girma, H., 2009. A Tutorial on Support Vector Machine. Center of Experimental Mechanics, University of Ljubljana.

[56] Cristianini, N. and Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[57] Butt, K.J. A study of feature selection algorithms for accuracy estimation. (Unpublished master's thesis), Universitat Polit_ecnica de Catalunya, BarcelonaTech

[58] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.