September 2020

Feature Selection Via Random Subsets Of Uncorrelated Features

Long Kim Dang
University of South Florida

Feature Selection Via Random Subsets Of Uncorrelated Features

by

Dang Kim Long

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Lawrence Hall, Ph.D.
Dmitry B. Goldgof, Ph.D.
Yu Sun, Ph.D.

Date of Approval:
September 21, 2020

Keywords: Gene Expression, Gini Index, High Dimensional Data, Random Subspace Method,
Concordance Correlation Coefficient Random Subspace Method

Dedication

This work is dedicated to my parents who has given me the chance to be specialized in computer science which I can read, write and collaborate with others with confidence.

Acknowledgments

I would like to thank:

- Dr. Lawrence Hall, for giving me a chance to get exposed to Computer Science research work.
- Dr.Rahul Paul, for his assistance in running the CCC_RSM's program
- Dr.Thang Hoang, for his help in setting up Eclipse for developing Gini Index feature selector
- My family, classmates and friends.

Table of Contents

List of Tables

List of Figures

Abstract

The role of feature selection is crucial in many applications. A few of these include computational biology, image classification and risk management. In biology, gene expression micro array data sets have been used extensively in many areas of research. These data sets typically suffer from an important problem: the ratio between the number of features over the number of examples is very high. This problem mainly affects prediction accuracy because it is best to collect more labeled examples than features. A correlation based random subspace ensemble feature selector (CCC_RSM) was proposed to handle this problem [5]. In this approach, first it determines the most relevant prediction features. Next, it groups these features based on their correlation to each other. Then, a feature is randomly chosen from each correlated group so that the selected features form a feature subset. The CCC_RSM algorithm repeats the previous step a pre-defined number of times. The proposed algorithm's performance is evaluated by combining either multiple decision trees or Support Vector Machines. Joining these models' predictions together can significantly increase the prediction accuracy. In ensembles of these models, each classifier provides a vote and the majority vote is used to produce the final class prediction. This design modifies the random subspace method ensemble [13].

This study focuses on finding alternative feature selectors in the first step so that the CCC_RSM algorithm can obtain good, or even better classification performance. We used four micro array gene expression data sets in the experiments. Based on the original algorithm, we used the Gini Index in place of Relief-F. A detailed analysis of the alternative method's outputs was considered: (1) overall and average accuracy, (2) Sensitivity, (3) Specificity, (4) F-measure.

Consequently, the alternative method gave the highest F-measure score for the Leukemia (1.00), Breast (0.98), Colon (0.91) and CNS (0.81) data sets.

Chapter 1: Introduction

A single human cell contains a 5-6 foot long string of deoxyribonucleic acid (DNA). Humans have around 25,000 genes in the genome and when a cell divides, it must make a copy of its DNA. A gene is a region of DNA that encodes for a functional product (RNA or protein). The recent increase in DNA sequence information related to cancer and technical advances in mining this valuable information have accelerated cancer genomics. One of cancer genomics' goals is to identify abnormal genes which drive the development and growth of many types of cancer in order to ultimately enable the development of targeted drugs, diagnostic tests, and the discovery of certain cancer sub-types [5]. In machine learning and data mining research, microarray gene expression data sets have increasingly become the common benchmark for application evaluation. One major problem with these data sets is that they have many genes compared to a relatively few numbers of data examples [29]. Another problem when classifying patient samples into classes or cancer sub-types using these data sets, the choice of gene selection method has a large effect on the performance and gene selection algorithms often select a relatively high number of genes. Thus, it is necessary to apply gene selection algorithms [17] [24] [18].

This paper considers gene selection as a feature selection problem so genes will be referred to as features and gene selection as feature selection. Feature selection algorithms aim to identify the most relevant and predictive features in a given data set while providing the best classification accuracy. Common types of feature selection algorithms include filter, wrapper, and sparsity based methods [11].

This research employs classification with an ensemble approach. It favors a classification model with multiple classifiers. Research has shown that using ensembles created from multiple base classifiers can improve results and provide greater confidence in results than when using a single classifier [5]. Each ensemble component makes a prediction. The predictions in this research are combined using majority voting [16].

Researchers have found different ways to create the individual components of an ensemble framework and the random subspace method (RSM) approach [13] is a type of ensemble classification technique. In RSM, the ensemble components are constructed by sampling features instead of instances [5]. RSM forms smaller subset of features by randomly selecting features from the original feature space. A base classifier is trained by using these feature subsets and repeating the feature subset random selection enables building an ensemble of classifiers. However, RSM assumes that the features are not highly correlated [5]. Moreover, because the features are randomly selected, the feature subsets might contain correlated or irrelevant features which affects the performance of the RSM [5]. However, it is possible to overcome the above problem. A modified version RSM [5] considers correlation between features. This method is called the concordance correlation coefficient based random subspace method (CCC_RSM) feature selector. In this approach, the authors used Relief-F filter to first rank the features based on their relevance. Then CCC_RSM selects the top ranked features. Groups of correlated features (based on the concordance correlation coefficient) were then constructed from the selected top relevant features. Finally, random feature subsets were generated by randomly selecting one feature from each correlated group.

In this research, we focus on analyzing the performance of CCC_RSM and its parameters in greater detail when using other ranking algorithms in place of ReliefF. Decision trees(J48) and SVM classifiers are used to do the evaluation of the alternative framework.

The rest of this thesis is organized as follows. In Chapter two, we describe the three feature selection algorithms: Gain Ratio, Relief-F, and Gini Index. In Chapter three, we describe the experimental design and its parameters as well as the data sets used to evaluate the alternative feature selectors. Chapter four contains the results when applying these alternative methods to select the relevant features and their performance on the classification task. Finally, Chapter five concludes with a discussion of some implications of the results for future research.

Chapter 2: Literature Review

2.1 ReliefF

ReliefF [21] is a traditional scheme-independent feature selection algorithm because the original attribute set is filtered to produce the best subset which contains the least number of features and most contribute to accuracy before the learning process starts. ReliefF adopts an "instance-based" learning method to rank features. In this algorithm, it first selects a random instance called R_i, then it picks the K nearest neighbors to the chosen example from each class - "nearest hits" (from the same class) and "nearest misses" (from other classes). Researchers in the field believe that this algorithm stems from the observation that an attribute seems to be irrelevant if a nearest hit has a different value for the attribute, but if a nearest miss has a different value, the attribute might be relevant. Assume that m data instances are randomly selected among all n instances then the feature score of a feature, f_i computed after m times is defined as follows:

$$\text{W[f\_i]} := \text{W[f\_i]} - \frac{1}{m \cdot K} \sum_{j=1}^{K} diff(f_i, R_i, H_j) + \frac{1}{m \cdot K} \sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^{K} diff(f_i, R_i, M_j(C)) \right] \tag{2.1}$$

where $P(.)$ denotes the probability. $diff(Attribute, Instance1, Instance2)$ is the similarity function which can be applied to both nominal and continuous features. This function is used for calculating the difference between the values of features for two instances. Also, to determine the distance between instances to find the nearest neighbors, we use this function. The distance between two instances is simply the sum of the attribute distances [21]. By using the term followed by the minus

sign, feature f_i is penalized when having different values for the "nearest hits" examples. Thus, to increment feature f_i's score when having distant values for the "nearest misses" examples, we add the last term. In the last term, the subscript under the outer summation means for each class C from other classes than R_i's class, class(R_i). H_j and $M_j(C)$ are the nearest instances of R_i in the same class and from other classes, respectively.

2.2 Gini Index

The Gini index [10] is also commonly used as a feature ranking algorithm. It computes and assigns a weight or merit score to each feature which indicates the feature's ability to separate instances from different classes. Given a feature f_i that takes on r different feature values, suppose A and $\overline{A}$ denote two sets of instances with the feature value smaller or equal to the jth feature value and larger than the jth feature value, respectively. In other words, the jth feature value can split the data set into A and $\overline{A}$, then the Gini index score for the feature f_i at the jth feature can be computed as follows

$$\text{gini index score}(f_i) = \min_{A}(p(A)(1 - \sum_{s=1}^{C} p(C_s|A)^2) + p(\overline{A})(1 - \sum_{s=1}^{C} p(C_s|\overline{A})^2)) \tag{2.2}$$

where p(.) denotes the probability and C_s is class s. For instance, p(C_s|A) is the conditional probability of class s given A. The subscript under the min function means we consider all the possible values r to find the split which gives the minimum Gini index score. As a result, this minimal value is the Gini score of feature f_i. For binary classification, the Gini Index score lies between [0, 0.5]. It can take a maximum value of 0.5 when A and $\overline{A}$ have equal size. It can also be used in multi-class classification problems. The lower the Gini index value, the more relevant the feature is [15].

2.3 Fisher Score

The Fisher score [9] is one of the most widely used criteria for supervised feature selection. It assigns more weight to features whose values examples from the same class are similar while values of instances from other classes are dissimilar [15].

Let denote n_j, μ_i, μ_{ij} and σ_{ij}^2 be the number of samples in class j, mean value for feature f_i, mean value of feature f_i for samples in class j, and the variance value of feature f_i for samples in class j, respectively. The Fisher score of each feature f_i is evaluated as follows:

$$\text{fisher\_score}(f_i) = \frac{\sum_{j=1}^{c} n_j(\mu_{ij} - \mu_i)^2}{\sum_{j=1}^{c} n_j \sigma_{ij}^2} \tag{2.3}$$

2.4 Gain Ratio

Gain Ratio [20] is an enhancement of the information gain feature selection algorithm [19] when some features have a large number of possible values [26]. Information gain measures attributes based on their information entropy [22]. Assume that S is a set consisting of s examples with 2 distinct classes, the process to determine the Gain Ratio score of an attribute A is expressed as below [14]. First, calculate the amount of information needed to classify a new example

$$I(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{2.4}$$

where p_i is the probability that an example belongs to class C_i and m is the number of classes.

Then we specifically denote feature A having non identical values, and let s_{ij} be the number of examples of class C_i in a subset S_j. S_j contains those examples in S that have value a_j of A. The entropy, or expected information considering the number of instances corresponding to the data split

from feature A, is given by

$$I(A) = -\sum_{a_j,i=1}^{m} \frac{s_{ij}}{s} \cdot I(S_j) \tag{2.5}$$

The informational value of creating a branch on the feature A is

$$\text{Gain (A)} = \text{I(S)} - \text{I(A)} \tag{2.6}$$

The intrinsic information value of A without considering the classes involved in the subsets is

$$\text{SplitInfo}_A(S) = -\sum_{a_j}(|S_j|/|S|)\log_2(|S_j|/|S|) \tag{2.7}$$

Lastly, we can modify the information gain by dividing by the intrinsic information value to get the gain ratio

$$\text{Gain Ratio(A)} = \text{Gain (A)}/\text{SplitInfo}_A(S) \tag{2.8}$$

The attribute with the higher Gain ratio value is, more significant the attribute is.

The original paper uses Relief-F as the first step to remove irrelevant features, so to explore more choices, we experimented with two alternative feature selectors, Gini index and Gain Ratio. However, Gain Ratio is not mentioned in the Methodology and Conclusion chapters because it does not support features with continuous values.

Chapter 3: Methodology

3.1 Algorithms

Algorithm 1 Algorithm for forming groups of correlated features (GrpsCorrFeat), from N features

1: Procedure GrpsCorrFeat(F_{ranked}, Concordance correlation coefficient threshold(TC))
2: Output: The correlated groups of features $CorrGrp_1, CorrGrp_2, ..., CorrGrp_{Count}$ and the number of the correlated groups, $Count \in [1, N]$.
3: $CorrGrp \leftarrow 0$; $Count \leftarrow 0$; $i \leftarrow 0$
4: $frAdded[x] \leftarrow 0$, $\forall x \in [1, N]$
5: **for** $x \leftarrow 1$ To N **do**
6: **if** $frAdded[x] = 0$ **then**
7: $frAdded[x] = 1$
8: $i \leftarrow i + 1$
9: $Count \leftarrow Count + 1$
10: $CorrGrp_i \leftarrow CorrGrp_i \bigcup x$
11: **for** $y \leftarrow x + 1$ To N **do**
12: **if** $|r_{yj}| \geq TC$ $\forall j \in CorrGrp_i$ **then**
13: $frAdded[y] \leftarrow 1$
14: $CorrGrp_i \leftarrow CorrGrp_i \bigcup y$ `#` r_{yj} `is the concordance correlation coefficient between feature y and feature j in the` i^{th} `feature subset`

Given a data set D with an s dimensional space $F_{orig} = f_0, f_1, .., f_s$, after running Gini Index to choose the best subset $N \leq s$, we applied Algorithm 1 to produce correlated groups of features. The i^{th} group has dimension q_i where $q_i \leq N$. It is important to note that the TC (correlation threshold) will determine the number of correlated groups, $Count$ where $Count \in [1, N]$. In a correlated feature group, $\{rfs_i = \hat{f}_1, \hat{f}_2, ..., \hat{f_{Count}}\}$; $\hat{f}_j$ and $\hat{f}_y, j \neq y$, have a limited correlation to each other and to all the features in rfs_i ($|r_{yj}| \geq TC$ $\forall j \in rfs_i$). The features in rfs_i are relevant and determined by the Gini Index feature selector. The $Count$ correlated group of features were then used to generate M feature subsets $\{rfs_1, rfs_2, ..., rfs_M\}$ in Algorithm 2, each with $Count$ features because we pick 1 random feature from each correlated group [5].

Algorithm 2 Algorithm for CCC_RSM with Gini Index Feature Selector

Input: Original feature set $F_{orig} = f_0, f_1, .., f_s, N, TC, M$
Output: M feature subsets $\{rfs_1, rfs_2, ..., rfs_M\}$
Rank features in F_{orig} using Gini Index
$F_{ranked} = fr_0, fr_1, .., fr_s, where fr_i$ has a higher rank than fr_{i+1}
Select first N features from F_{ranked}
$[CorrGrp, Count] = GrpsCorrFr(F_{ranked}, TC)$ `# Call the function to form groups of correlated features`
for $x \leftarrow 1$ To M **do**
    for $y \leftarrow 1$ To $Count$, for each correlated group **do**
        $rand\_feature \leftarrow GetRandFr(CorrGrp_y)$ `# Pick 1 random feature from each correlated feature group`
        $rfs_x \leftarrow rfs_x \bigcup rand\_feature$

3.2 Dataset Acquisition

We used four micro array data sets: Colon (colon cancer), Leukemia, Breast cancer and Central nervous system (CNS). These data sets all have less than 80 instances, but with a high number of features ranging from 2,000 to 7,129. Each data set comprises 2 classes. These data sets are used extensively by researchers to evaluate feature selection algorithms for cancer classification problems. Details are shown in Table 3.1. WEKA [25], a Java based open source code set for machine learning was used to conduct the experiment. The Gini Index feature selector in Java takes its inspiration from the implementation of the CART algorithm [23]. We used the LIBSVM library [4] for Support Vector Machine. It should be noted that the Colon data set and Breast cancer data set were not available in the WEKA input file format, ARFF which stands for the attribute-relation file format. For the Colon data set, the raw input was in Microsoft Excel format while for the Breast cancer, the raw input was in Matlab format. The ARFF files were prepared after a few intermediary steps. In this data conversion step, no data processing steps were conducted. Therefore, data is assured to remain intact before and after the conversion.

Table 3.1: Details Of Microarray Datasets

Dataset	No.of Genes	No.of samples	Pos/Neg	Source
Breast cancer	7129	44	21/22	Not available for download from the Internet
Central nervous system(CNS)	7129	60	39/21	http://csse.szu.edu.cn/staff/zhuzx/Datasets.html
Colon	2000	62	40/22	Not available for download from the Internet
Leukemia	7129	72	47/25	http://csse.szu.edu.cn/staff/zhuzx/Datasets.html

3.3 Experimental Setup

We followed the same approach described in [5]. The authors proposed a hybrid approach between filter based algorithms and ensemble approach to select the best feature subsets, called Concordance Correlation Coefficient Random Subspace Method (CCC_RSM). At first, the authors selected relevant features using the Relief-F algorithm. Then, they used the Concordance correlation measure [1] to form groups of correlated features from the selected features. The authors' approach to form groups of correlated features is shown in Algorithm 1. Next, they randomly selected a feature from each group to form a subset of features which is used to train the ensembles. This step has been shown in Algorithm 2. For both the original and the alternative CCC_RSM, two parameters were examined: the number of top ranked features (N) produced by the Gini Index ranging from 5 to 50 by a step of 5, and the correlation threshold values (TC) to form correlated groups ranging from 0.1 to 0.99 by a step of 0.01. The number of random feature subsets (M) was set to 100. M = 100 does not mean 100 unique feature sets. For example, if we have 3 correlated groups which are (a,b,c,d,e,i,j), (f,h), (g). M would be equal to 14, 7 x 2 x 1 = 14, so we expect to have many identical subsets among 100 feature subsets. For these four data sets with thousands of features, we might or might not account for all possible combinations of less relevant features because there are some identical feature subsets. The detailed analysis of the Gini Index results will be presented in Chapter 4: Findings. As an initial step, the data sets were normalized so that all numeric features values, except for the label attribute, would be in [0, 1]. This first step helps ensure that all features are measured based on the same scale. Next, it is easy to see that from, Table 3.1,

the Leukemia, Colon, and CNS data sets suffer from the class imbalance problem because usually there are fewer cancer patients than healthy ones available when acquiring these data sets. In the extreme case, in a binary classification problem, if one applies 0-R classifier on one of these data sets, the classifier would always predict the majority class, the 'negative' class. Therefore, the accuracy would be mostly high because the 'negative' examples dominate in these data sets. If one selected this model as the base line performance, the ability to predict the illness class would not exist. To avoid this problem, the synthetic minority over sample technique SMOTE [6] was applied to add new, minority class samples [5]. We used 5 nearest neighbors and 100% oversampling to acquire a more balanced class distribution to ensure that there is no difference in the parameter selection when comparing with the original approach. Since a separate test set is not available and the data sets'size is limited, according to the original paper, Leave-One-Out cross validation (LOOCV) was adopted to evaluate the ensemble classifiers. In LOOCV, a data set is divided into n folds where n is equal to the number of examples. In each fold, there are $(n-1)$ examples in the training set including the newly created examples and 1 hold out example in the test set. Each example is used once for testing, except the artificially created ones, and $(n-1)$ times for training. For each fold, we conducted feature selection on the $(n-1)$ training examples to select the top ranked features. However, the number of chosen features is predefined in this experiment, so another method should be studied to determine the appropriate number of features to use [5]. Since both Relief-F in the original paper and Gini Index evaluate features individually, they fail to detect correlated features. For example, two identical features would be either selected or both rejected [5]. Their combination might be influential on the classification performance. As a result, the next step is to build groups of correlated features using the concordance correlation coefficient [1]. Then random, less correlated feature subsets were formed from groups of correlated features, and these different feature subsets were fed to machine learning algorithms to develop an ensemble. Majority voting

was used to combine the predictions in the ensemble. No artificial samples generated by SMOTE were used for testing.

3.3.1 Gini Index

Gini Index is used as a measure to find the best features to split the data sets for building Classification and Regression Trees (CART) [3]. It works well with both nominal and numerical features. Intuitively, the Gini index measures the impurity of successive nodes after a partition. A node is maximally impure if it has equal distribution across all available classes. We would like to choose the relevant features which can separate the examples into discrete groups as much as possible. The best scenario occurs when we have a pure node, containing examples form one group only. In detail, a Gini score is in the range [0, 1]. A feature is more relevant when its Gini score is close to 0. For example, when a data set has only 2 classes, a feature with its Gini score of 0 means, after branching, there are two subsets; each includes examples from one class only. Therefore, we want to keep the features with lower, close to 0 index scores.

Though the Gini index score discovers all relevant features, it fails to remove correlated features. In the second step, we used the concordance correlation coefficient (CCC) [1] (r_c) to find the correlated features and group them together. The concordance (r_c) gives a measure of the agreement relationship between two continuous variables X and Y. r_c lies in the interval [-1, 1] where -1 implies negative agreement, and 1 indicates positive agreement, while a zero value indicates no value [5] [8]. The value r_c is determined by the following formula:

$$r_c = \frac{2cov(X,Y)}{var(X) + var(Y) + (\overline{X} - \overline{Y})^2} \tag{3.1}$$

where *cov* is the covariance, *var* is the variance, $\overline{X}$ is the mean of X and $\overline{Y}$ is the mean for Y.

Once r_c is generated, a pair X and Y are said to be correlated if $\mid r_c \mid \geq TC$. TC means a threshold value. In my opinion, other correlation coefficients which give a measure of the linearity relationship between 2 variables such as Pearson and Spearman can also be considered for the same purpose.

Two ensembles which incorporate decision trees (DT) and support vector machines (SVM) were created with the top ranked Gini index features.

3.3.2 J48 Decision Tree

A decision tree is a way of representing knowledge in machine learning for which each internal node involves a choice between attributes and each terminal (leaf) node gives a classification [27]. In practice, decision tree learning is popular because it produces models with acceptable performance [23] and with good interpretation, quickly. In this thesis, we used WEKA's J48 Java implementation of the C4.5 release 8 algorithm [20] and error based pruning method [28] to avoid over fitting and be resilient with noise. We used a confidence factor of of 0.25 for this pruning technique. Witten (2017) et al. suggest that the error based pruning method while building J48 decision tree is preferred in practice, rather than reduced error pruning. The reduced error pruning method holds back training instances and uses them as the pruning set. It suffers from the disadvantage that the classification tree is built on less data and we experimented on data sets having only a few examples. According to the C4.5 release 8 algorithm, a decision tree is built in depth-first order. First, it selects an attribute for the root node and creates one branch for each possible attribute value for nominal attributes. This breaks the instance set into subsets, one for every value of the attribute. For continuous attributes, it chooses one feature to break the instance set into two subsets. Then this process can be reapplied recursively for each branch, using only those examples that reaching the branch. If at any time, all instances at a node belongs to the same class, we consider stop developing

that part of the tree [27]. Decision trees recursively test an attribute and pick the attribute which after splitting gains the most information to classify the data examples. Some challenges of decision trees include space complexity, the size of trees grow linearly with the number of training instances without properly pruning [12], and difficult to visualize the learning output from large and complex trees etc.

3.3.3 Support Vector Machine

SVM is a supervised, linear classifier. SVM can also be extended to be nonlinear by using kernel functions such as polynomial kernel, radial basis function (RBF), and sigmoid. SVM can provide good performance through building an optimal separating hyperplane. This maximal margin hyperplane is defined by the training examples from both classes nearest to the separating hyperplane which are known as support vectors. In soft margin SVM, support vectors can be on the wrong side of the margin of the hyperplane while they must be on the margin and on the correct side for hard margin SVM. Soft margin SVM is used widely in practice because it can cope with noise, allowing support vectors on the wrong side of the margin. However, there are some disadvantages in that the choice of a suitable kernel function must be considered and CPU time is high for tuning the kernel function parameters especially for non-linear kernel functions (RBF), etc [5]. We used C-Support Vector Classification [2][7] supported in LIBSVM , a soft margin hyperplane based SVM with the RBF kernel. We followed the default parameters suggested by LIBSVM for our experiments [4].

We used the following metrics to measure the effectiveness of the alternative method:

- True Positive (TP): Number of positive samples correctly classified as positive.
- True Negative (TN): Number of negative samples correctly classified as negative.
- False Positive (FP): Number of negatives samples misclassified as positive.

- False Negatives (FN): Number of positive samples misclassified as negative.

- Overall accuracy is the probability of correctly identifying both positive and negative samples and given as

$$\text{Acc} = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.2}$$

- Sensitivity (Sens) also known as Recall: the probability of correctly identifying positive samples

$$\text{Sens} = \frac{TP}{TP + FN} \tag{3.3}$$

- Specificity (Spec): Probability of correctly identifying negative samples

$$\text{Spec} = \frac{TN}{TN + FP} \tag{3.4}$$

- Average accuracy (for 2 class problem):

$$\text{AvgAcc} = 0.5 \times (\text{Accuracy Class1} + \text{Accuracy Class2}) \tag{3.5}$$

- Precision is the probability of correctly identifying positive samples from all the samples which the classifier returned as positive and given as

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.6}$$

- The F-measure is the harmonic mean of precision and recall which is the same as sensitivity and given as

$$\text{F measure} = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{3.7}$$

Chapter 4: Findings

4.1 Overall And Average Accuracy

The highest overall accuracy obtained for the alternative method (Gini Index CCC RSM) when evaluated with the J48 and SVM ensembles for each of the four data sets, without its accompanying parameter design and its sensitivity and specificity, can be seen detailed in Figure 4.1. Tables 4.1, 4.2, 4.3, 4.4 list the number of top Gini index ranked features and the statistics of the correlated thresholds to produce the highest average accuracy, together with its sensitivity and specificity. Those table cells highlighted in bold red are the parameter values that are required to generate the accuracy of the ensembles, including the original method for comparison purpose. For example, in Table 4.1, with Gini Index CCC_RSM_SVM method, in red highlighted cells, shows how much each parameter needs to be changed individually to reach a 1.00 in both the average accuracy, sensitivity and specificity. The first parameter to set is the number of top ranked features produced by the Gini Index, ($N = 10$). The next parameter is the boundary of the concordance correlation coefficient threshold values (TC), defining the concordance correlation coefficient, r_C, which is we used to find the correlated features and group them together if $r_C \leq TC$ ($Min_{TC} = 0.35$, $Average_{TC} = 0.38$ and $Max_{TC} = 0.41$).

- Leukemia data set: Figure 4.1 represents a bar chart. The bar chart depicts the best overall accuracy for the Leukemia data set with both classifiers having greater than 99% overall accuracy. Note, we will later report average accuracy in a Table which will often be different. The alternative method when evaluated with SVM classifier is more accurate than the J48

classifier. The accuracy difference is 1%. It is also interesting to note that the highest accuracy occurs at 100% for the SVM classifier.

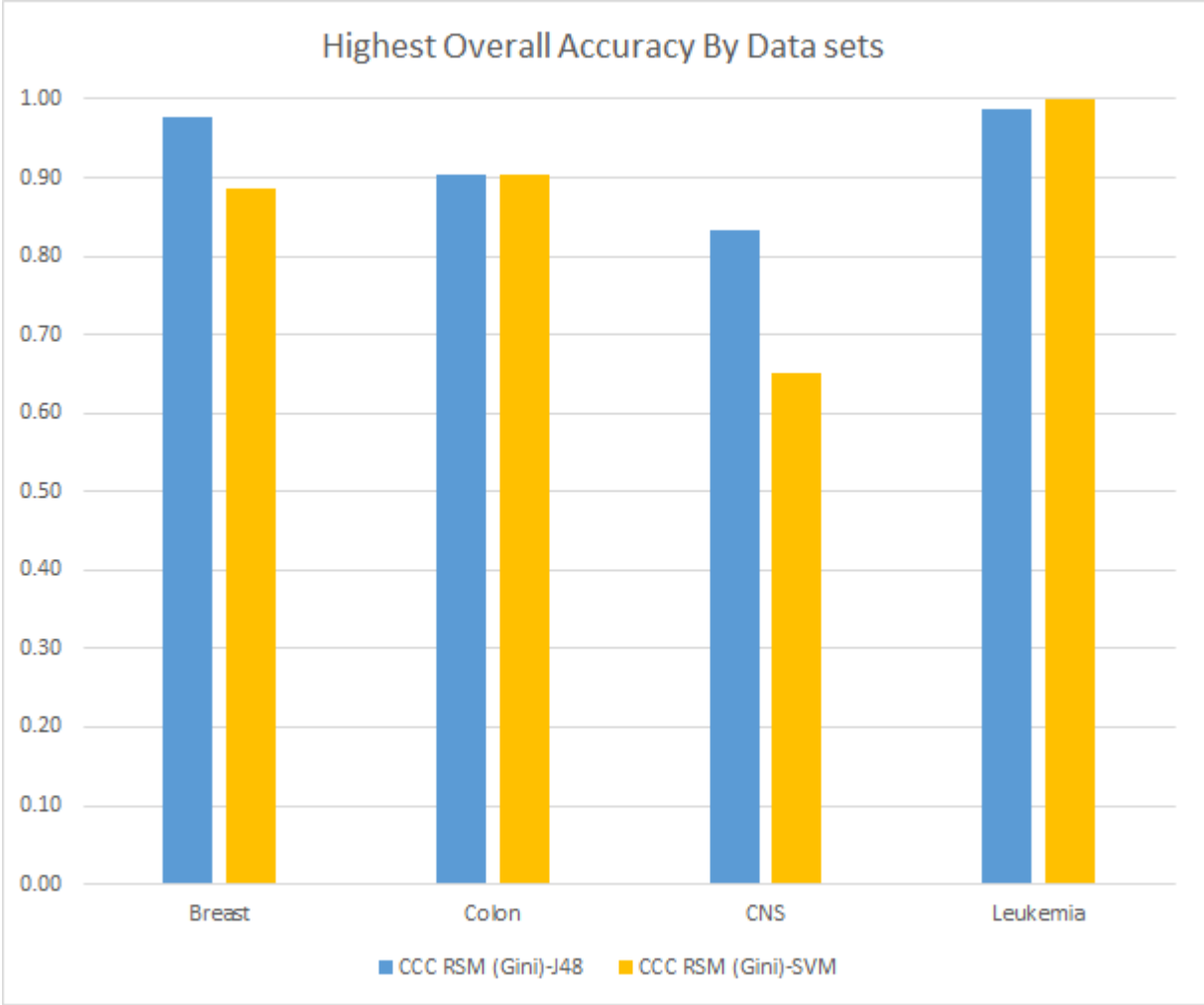

Figure 4.1: Highest Overall Accuracy By Data Sets

- Colon data set: the bar chart presents a decrease in overall accuracy when comparing to the Leukemia data set. Both these two methods achieve the same accuracy of 90%.

- CNS data set: Figure 4.1 shows that with overall accuracy, when evaluated with SVM, the CCC_RSM gave the worst result among all four data sets at 65%. Though it gave the worst result when evaluated with the SVM classifier, it gave a better result when evaluated with J48

classifier, at 83.33%. However, the CNS data set still has the lowest accuracy for all measures. This result was also reported in another experiment which uses fuzzy set similarity measures to group uncorrelated features [8].

- Breast data set: The bar chart shows that the method when evaluated with the J48 classifier produces the highest accuracy at 98%. When evaluated with SVM classifier, the method is less accurate with an 89% accuracy. Another important point to note was obtained when evaluated with the SVM classifier followed by the Breast data set with the J48 classifier. The lowest accuracy was obtained when analyzing the CNS data set for both classifiers.

- Summary over the two classifiers: The alternative method with the J48 classifier has the highest or tied accuracy in three of the four data sets, i.e., the exception being for the Leukemia data sets where the SVM classifier has the highest accuracy of 100%. The tie for the J48 classifier happens with the SVM classifier for the Colon data set at 90%.

From Tables 4.1, 4.2, 4.3, and 4.4, for the Breast data set, the best result of the alternative method when evaluated with SVM classifier is achieved with the top 10 ranked features. The CNS and Colon data sets have a small range of the top ranked features, i.e.: N = 20 and N = 50 and N = 20 and N = 30 respectively. The Leukemia data set allows for a high accuracy with a broader range of top ranked features, from N = 10 to N = 40.

4.2 Sensitivity

The bar chart in Figure 4.2 shows the highest sensitivity for each parameter set for each data set. From this bar chart, the greatest overall sensitivity values occur for both the Leukemia and Breast data set with all measures achieving a value of 100 %. However, as seen in Table 3.1, the Leukemia data set is imbalanced on the number of positive and negative cases while the Breast data

set is balanced. Chaudhury (2015) states that, for the Leukemia data set, without SMOTE, the average accuracy of the original CCC_RSM reduced from 98% to 96%, so the original algorithm does not cope well with handling an imbalanced data set. This also might be the case for the alternative algorithm. Furthermore, the smallest sensitivity values of 80% are produced for the alternative method when evaluating with the J48 ensemble on the CNS data set and 93% for the SVM on the Colon data set. It must be noted that this algorithm generates the sensitivity value of 100% with different N values which vary from 10 to 50. In detail, for the SVM ensemble, this result happens at N=10, 30, 40 and 50. For the J48 ensemble, it occurs when N=10, 20, 30, 40, and 50.

4.3 Specificity

The bar chart in Figure 4.3 provides the highest specificity performance of the ensembles for each data set. With the information that this bar chart provides, the smallest specificity values occur for both ensembles for the CNS data set. The worst results occur for the CNS data set is the J48 ensemble with 82% specificity while the SVM ensemble is much less accurate with a value of only 56%. Only the Leukemia data set has both ensembles producing a specificity of 100%. None of the other three data sets achieve the specificity of 100%. However, the Breast and Colon data sets all have specificity values more than 78%.

To summarize, the J48 ensemble has the highest or tied for the highest specificity value for three of the four data sets. The SVM ensemble is only more accurate with a specificity value of 93% for the Colon data set. The SVM ensemble has the lowest specificity for the Breast and CNS data sets. Again, for Leukemia data set, both ensembles have equal specificity of 100%.

4.4 F-measure

To incorporate sensitivity and specificity in a single metric quantity, Figure 4.4 plots the F-score performance for each data set. With the information that it provides, we can see that the least F-score values occur for both ensembles with the CNS data set. The best performing ensemble for the CNS data set is the SVM ensemble with an 80% value while J-48 ensemble is less accurate with a value of 77%. Only the Leukemia data set has both ensembles producing an F-score of 100% because both specificity and sensitivity values are 100%. None of the other three data sets achieve the specificity of 100%. Results with the Colon data set follow the same pattern with CNS data set, only 1% difference (91% vs 90%). However, the Breast data set shows an opposite result. The J48 ensemble (F-measure = 98%) outperformed SVM (F-measure = 95%) with 3% difference.

Sometimes the product of sensitivity and specificity can be used as an overall measure, Figure 4.5. Comparing the results from both charts, they give similar results for Leukemia and Breast data sets, i.e.: J48 ensemble is the winner. However, there is a big difference between them for Breast data set. In the combined Sensitivity and Specificity, the difference is 17% while only 3% for the F-score. For the combined metric, the J48 ensemble performs better than the SVM ensemble for both Colon and CNS data set.

4.5 Average Accuracy Sensitivity

From Figures 4.6, 4.7, 4.8, and 4.9, we consider the effect of the correlation threshold and the number of top ranked features on the average accuracy. From these figures, we can see that there is a transition from increase to decrease in the average accuracy at a specific threshold value. For example, in Figure 4.6, for N = 40, the transitional point was manually found at the correlation threshold value of 0.12 where the best accuracy was achieved. When the correlation threshold rises to 0.12, the average accuracy generally increases to the best accuracy. Passing this point, the accuracy

starts declining. However, there is not the same effect on average accuracy with a change in the value of N. For example, in Figure 4.7 for the CNS data set, the accuracy increases from N = 10 to N = 50. On the other hand, for the Leukemia data set, the accuracy seems to drop with a rise in the number of top ranked features.

Table 4.1: Leukemia: Highest Average Accuracy with N (best features selected) and Correlated Threshold Values

Method	N	MIN[1]	AVG[2]	MAX[3]	Avg no. of features[4]	Avg acc	Sens	Spec
ReliefF - CCC-RSM-J48					7	0.98	1.00	0.96
ReliefF - CCC-RSM-SVM					5	0.869	0.978	0.76
Gini-CCC-RSM-J48	10	0.10	0.27	0.45	1.4	0.98	1.00	0.96
Gini-CCC-RSM-J48	20	0.33	0.37	0.39	4.9	0.98	1.00	0.96
Gini-CCC-RSM-J48	30	0.34	0.42	0.48	6.8	0.98	1.00	0.96
Gini-CCC-RSM-J48	40	0.18	0.41	0.63	10.6	0.98	1.00	0.96
Gini-CCC-RSM-SVM	**10**	0.35	0.38	**0.41**	**2.1**	**1.00**	1.00	1.00

[1] denotes the minimum threshold for the correlation accompanied by the two conditions. First, all the features within a group must achieve this correlated condition to each other. Second, our experiment showed that using this threshold value in Leukemia data set with our alternative approach generated an average accuracy of 0.98.
[2] is defined as the average of all the correlated threshold values which produced the same average accuracy.
[3] is equivalent to the maximum correlated threshold
[4] indicates the average number of correlated groups per fold which is equal to the number of features used for building each ensemble component. For example, in 4.1, when running CCC_RSM evaluating with J48 and N = 10, there were total 2530 folds with 3577 correlated groups in total so the average number of correlated groups per fold is 1.4.

Table 4.2: Colon: Highest Average Accuracy with N (best features selected) and Correlated Threshold Values

Method	N	MIN	AVG	MAX	Avg no. of features	Avg acc	Sens	Spec
ReliefF - CCC-RSM-J48					2	**0.916**	0.875	0.955
ReliefF - CCC-RSM-SVM					2	0.8815	0.90	0.863
Gini-CCC-RSM-J48	20	0.3	0.3	0.3	4.6	0.90	0.90	0.91
Gini-CCC-RSM-SVM	20	0.65	0.67	0.69	11.1	0.89	0.93	0.86
Gini-CCC-RSM-SVM	30	0.47	0.63	0.68	12.8	0.89	0.93	0.86

Table 4.3: CNS: Highest Average Accuracy with N (best features selected) and Correlated Threshold Values

Method	N	MIN	AVG	MAX	Avg no. of features	Avg acc	Sens	Spec
ReliefF - CCC-RSM-J48					5	**0.8864**	0.9523	0.8205
ReliefF - CCC-RSM-SVM					5	0.739	0.758	0.72
Gini-CCC-RSM-J48	50	0.37	0.37	0.37	24.9	0.83	0.81	0.85
Gini-CCC-RSM-SVM	20	0.39	0.41	0.43	12.5	0.70	0.86	0.54

Table 4.4: Breast: Highest Average Accuracy with N (best features selected) and Correlated Threshold Values

Method	N	MIN	AVG	MAX	Avg no. of features	Avg acc	Sens	Spec
ReliefF - CCC-RSM-J48					4	0.9306	0.9047	0.9565
ReliefF - CCC-RSM-SVM					4	0.863	0.857	0.87
Gini-CCC-RSM-J48	**40**	0.12	0.12	0.12	**4.9**	**0.98**	1.00	0.96
Gini-CCC-RSM-SVM	30	0.40	0.42	0.43	13.6	0.89	1	0.78
Gini-CCC-RSM-SVM	40	0.34	0.48	0.68	20	0.89	1	0.78
Gini-CCC-RSM-SVM	50	0.46	0.54	0.62	28.3	0.89	1	0.78

Table 4.5: Top 5 Most Selected Features From Gini CCC_RSM

Dataset	Features
Colon	X1671, X66, X765, X377, X1042
Breast Cancer	5812^{th} , 4473^{th}, 2565^{th} , 5433^{th} , 6484^{th}
Leukemia	M63379_at, M96326_rna1_at, M84526_at, U46499_at, X17042_at
Central nervous system	AF001787_s_at, U08998_at,L33243_at, S71824_at, S76475_at

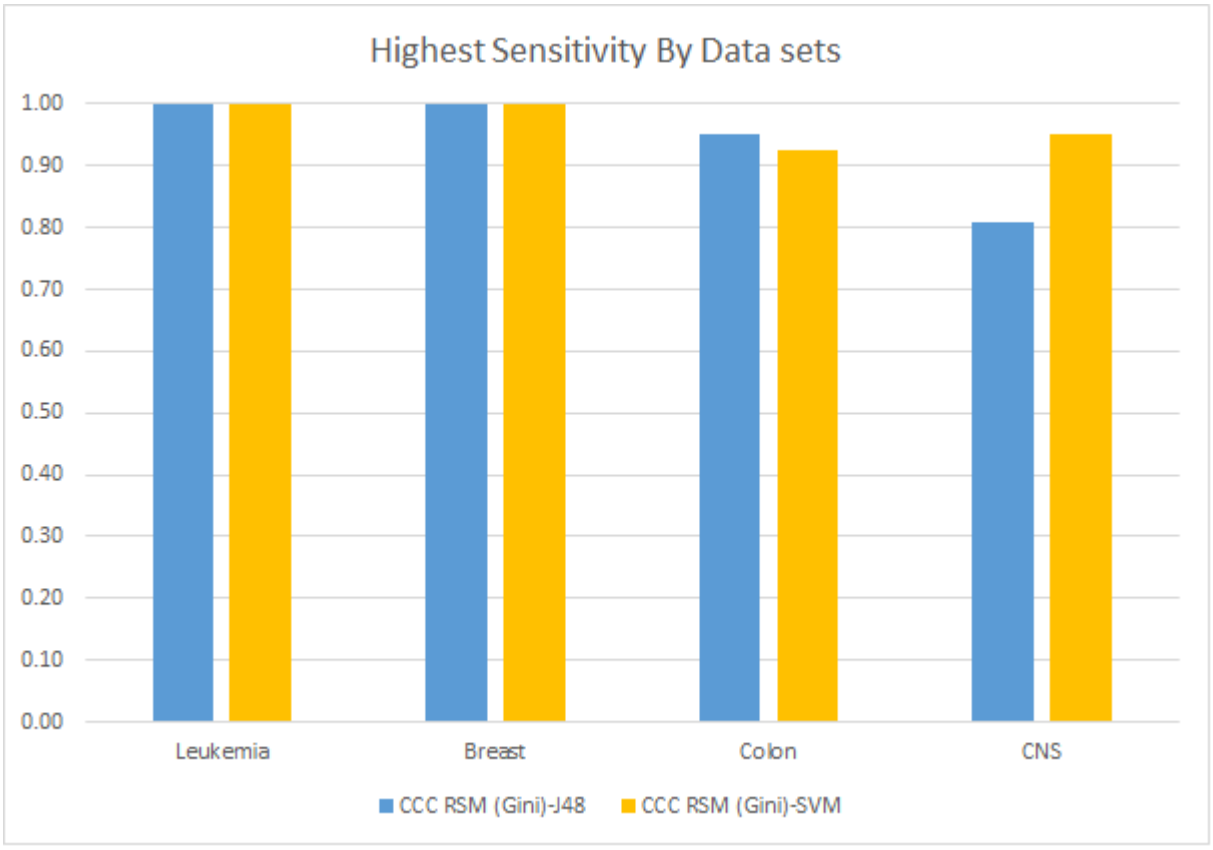

Figure 4.2: Overall Sensitivity By Data Sets

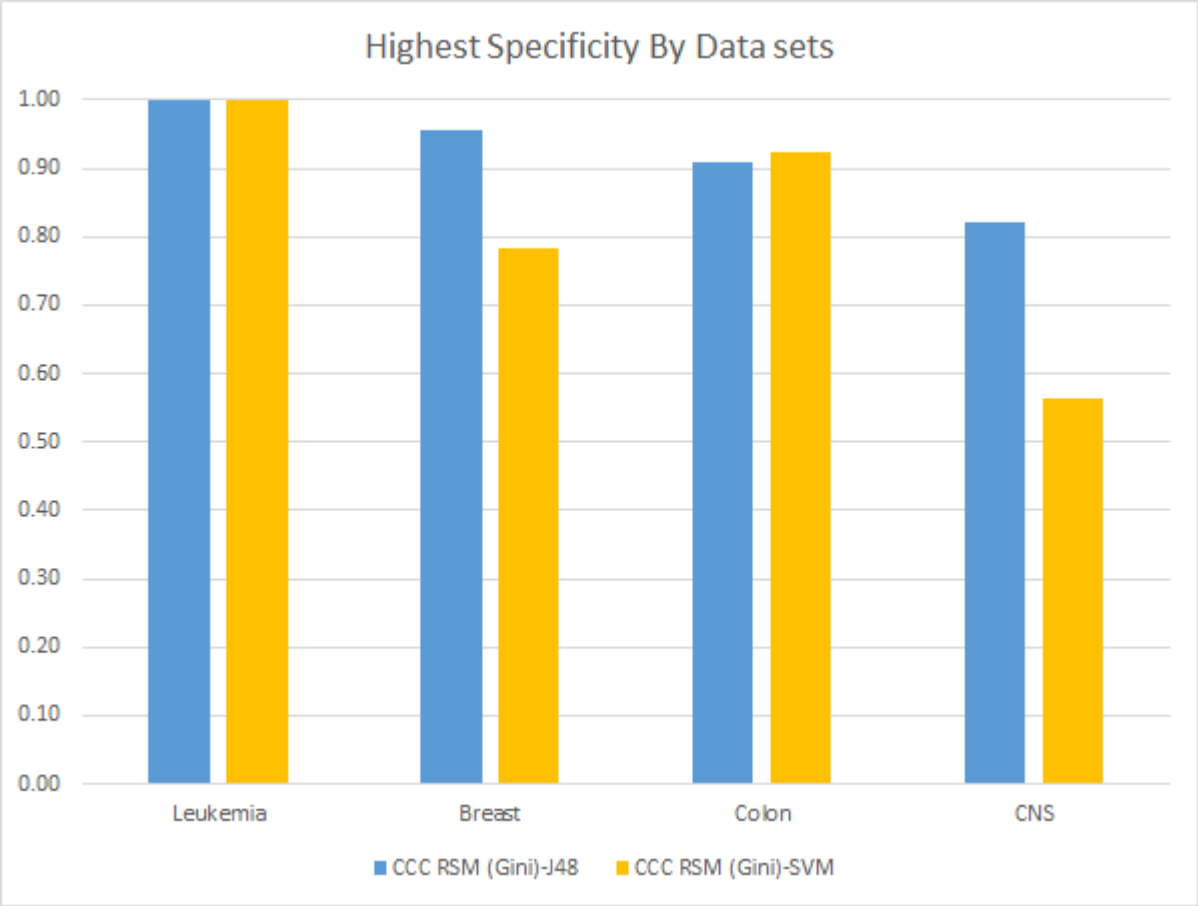

Figure 4.3: Overall Specificity By Data Sets

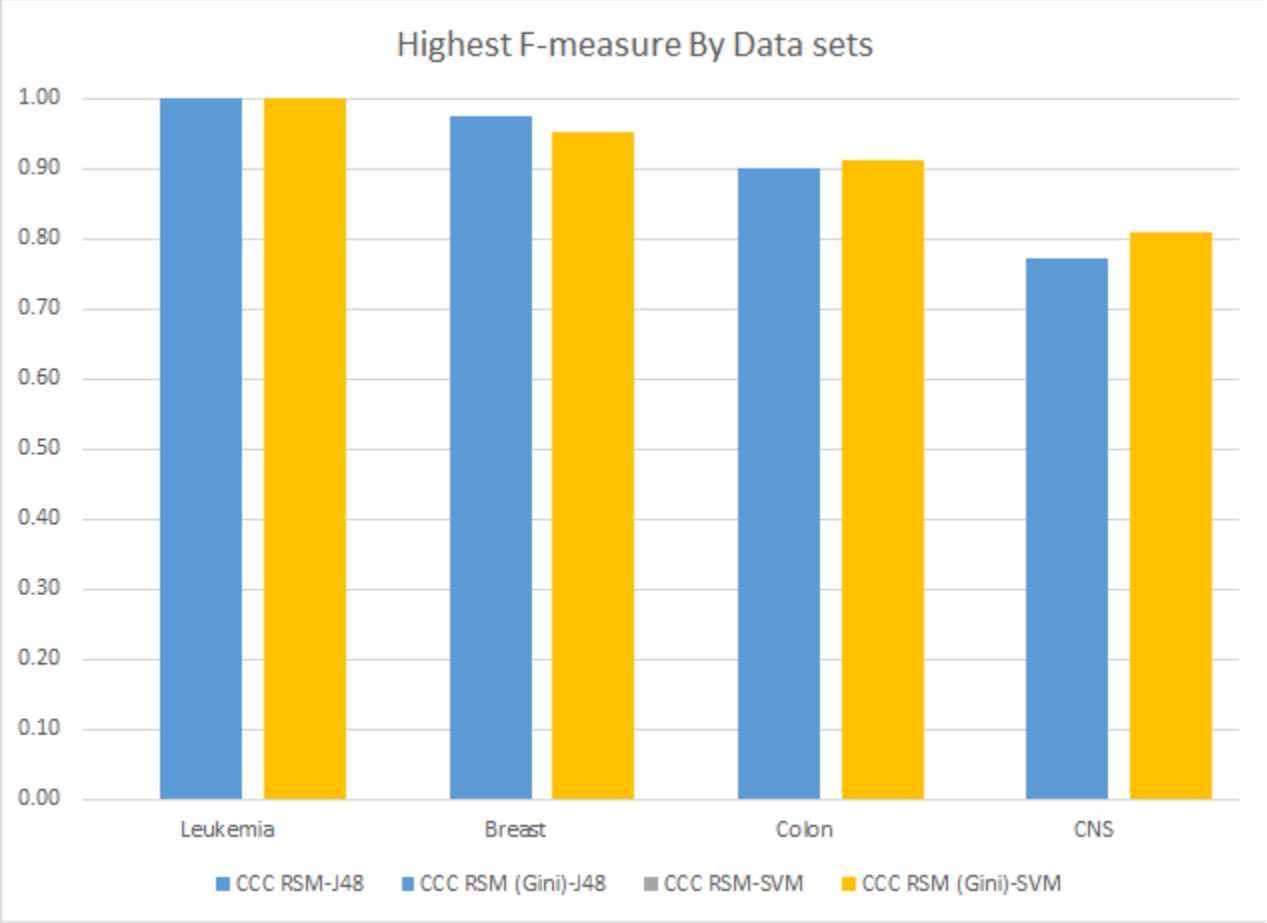

Figure 4.4: Overall F-measure By Data sets

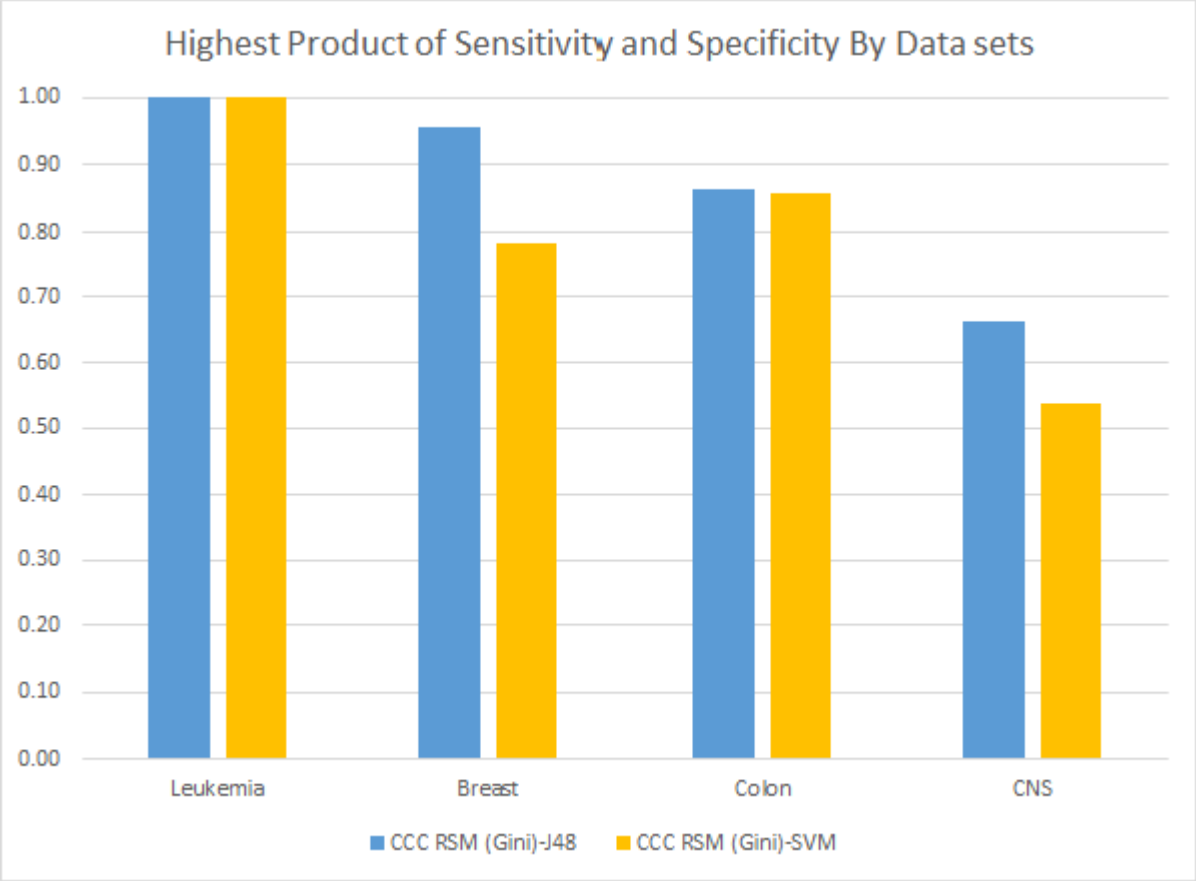

Figure 4.5: Overall Product of Sensitivity and Specificity By Data Sets

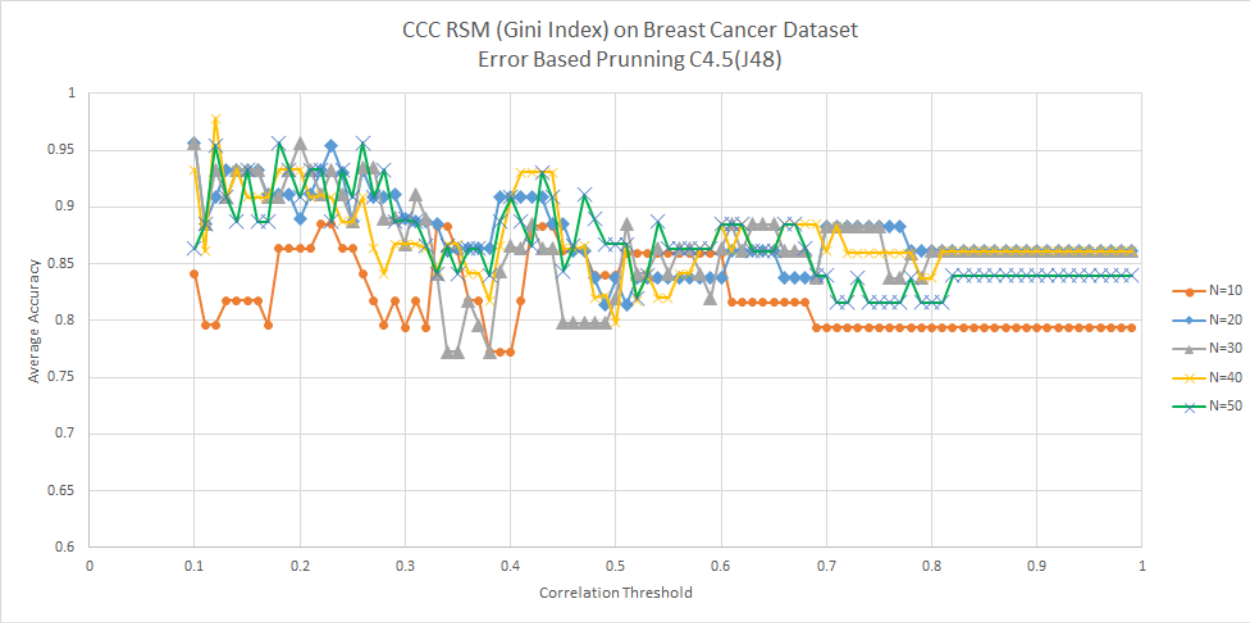

Figure 4.6: Breast: Average Accuracy vs Correlation Threshold For Different Values Of Top Ranked Features, N

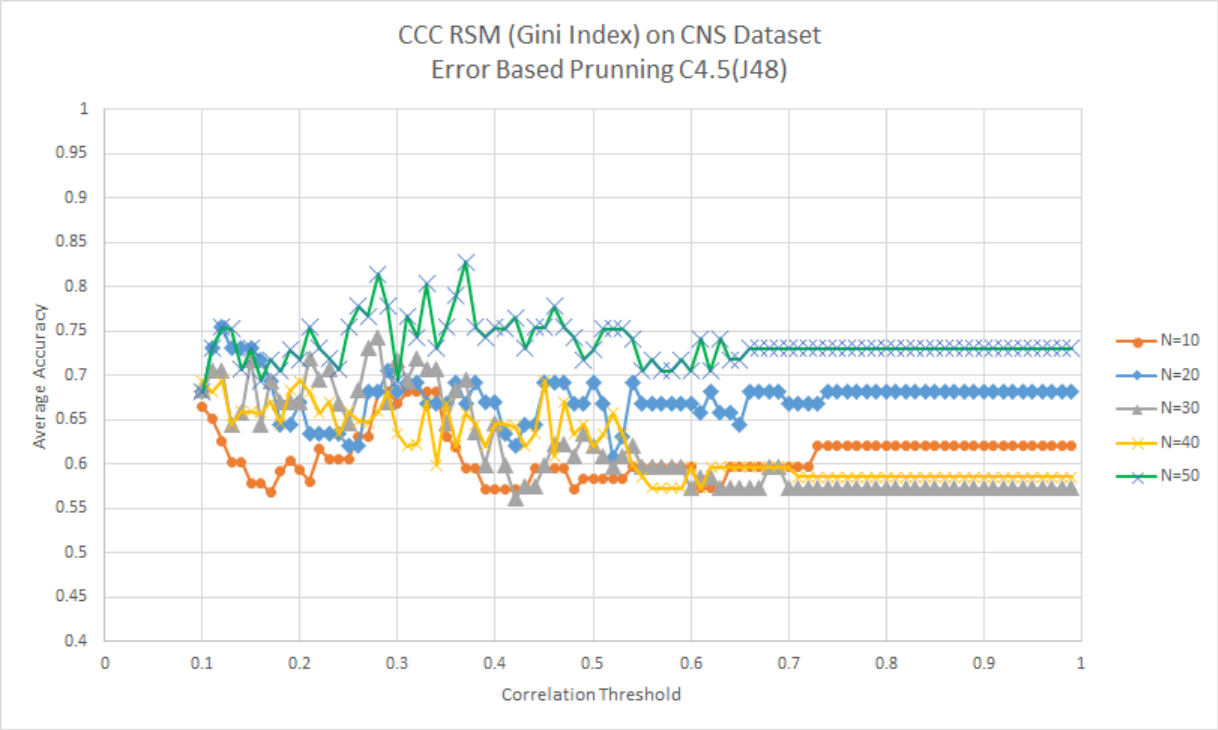

Figure 4.7: CNS: Average Accuracy vs Correlation Threshold For Different Values Of Top Ranked Features, N

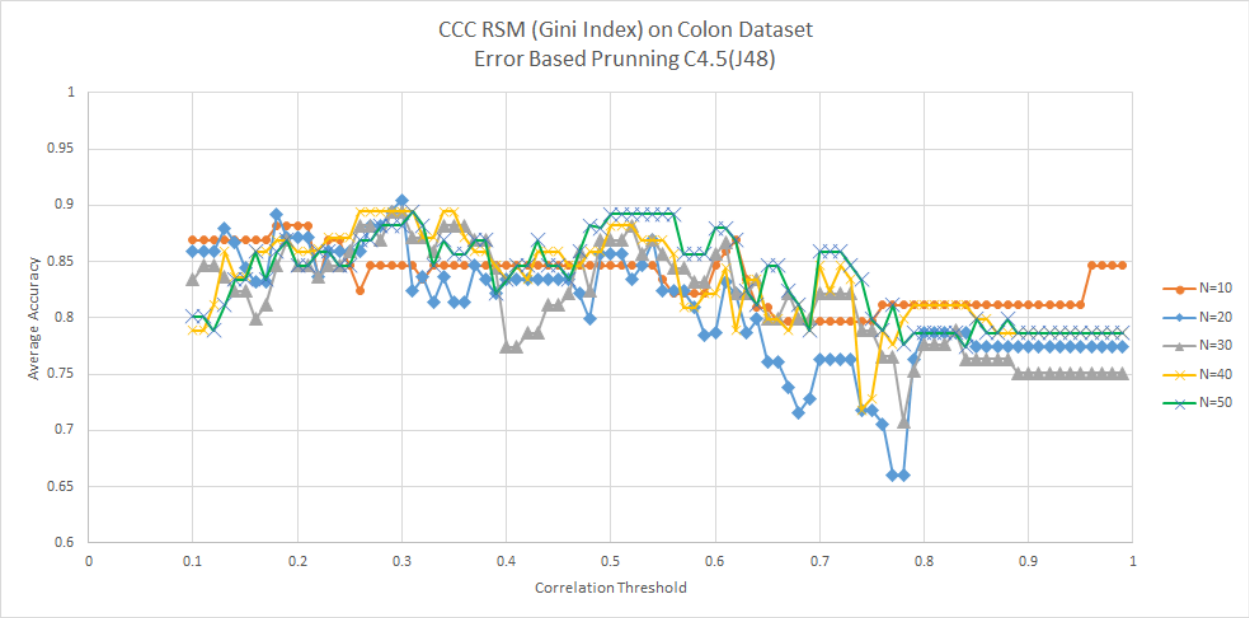

Figure 4.8: Colon: Average Accuracy vs Correlation Threshold For Different Values Of Top Ranked Features, N

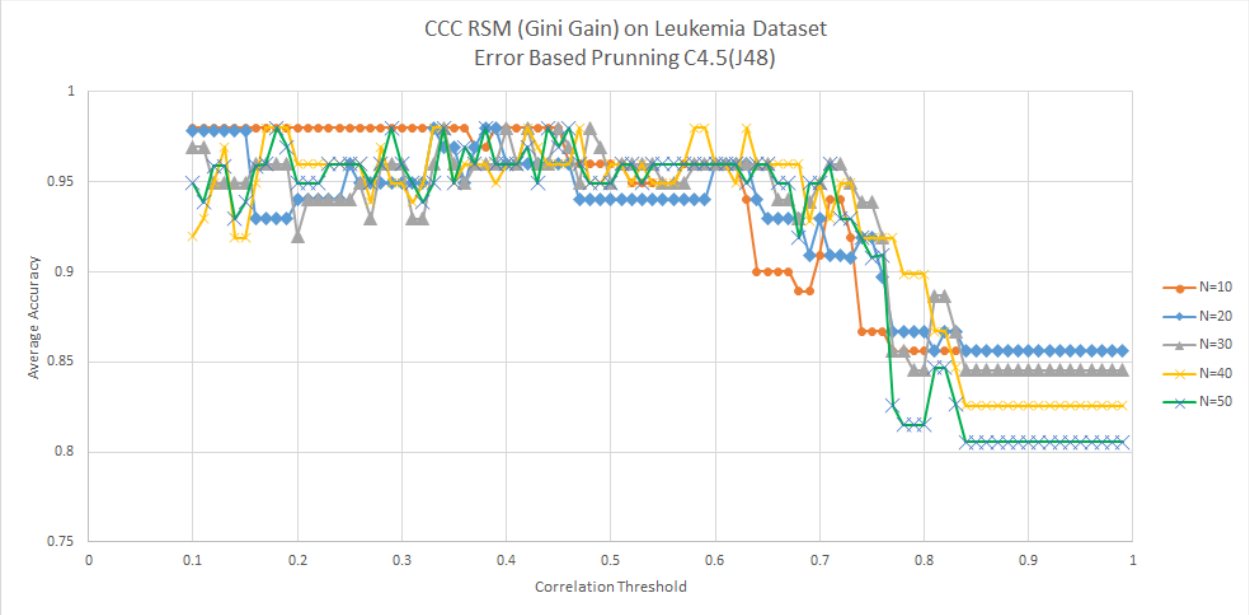

Figure 4.9: Leukemia: Average Accuracy vs Correlation Threshold For Different Values Of Top Ranked Features, N

References

[1] J Martin Bland, Altman, and Douglas G. Statistical methods for assessing agreement between two methods of clinical measurement. *The lancet*, 327(8476):307–310, 1986.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[5] Baishali Chaudhury, Dmitry B Goldgof, Lawrence O Hall, Robert A Gatenby, Robert J Gillies, and Jennifer S Drukteinis. Correlation based random subspace ensembles for predicting number of axillary lymph node metastases in breast dce-mri tumors. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2164–2169. IEEE, 2015.

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] Valerie Cross, Michael Zmuda, Rahul Paul, and Lawrence Hall. Fuzzy set similarity for feature selection in classification. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2020.

[9] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[10] Corrado W Gini. Variability and mutability, contribution to the study of statistical distributions and relations. studi cconomico-giuridici della r. universita de cagliari (1912). reviewed in: Light, rj, margolin, bh: An analysis of variance for categorical data. *J. American Statistical Association*, 66:534–544, 1971.

[11] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[12] Lawrence O Hall, Kevin W Bowyer, Robert E Banfield, Steven Eschrich, and Richard Collins. Is error-based pruning redeemable? *International Journal on Artificial Intelligence Tools*, 12(03):249–264, 2003.

[13] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[14] Asha Gowda Karegowda, AS Manjunath, and MA Jayaram. Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 2(2):271–277, 2010.

[15] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.

[16] Huawen Liu, Lei Liu, and Huijie Zhang. Ensemble gene selection by grouping for microarray data classification. *Journal of biomedical informatics*, 43(1):81–87, 2010.

[17] Jenny Önskog, Eva Freyhult, Mattias Landfors, Patrik Rydén, and Torgeir R Hvidsten. Classification of microarrays; synergistic effects between normalization, gene selection and machine learning. *BMC bioinformatics*, 12(1):390, 2011.

[18] Mehdi Pirooznia, Jack Y Yang, Mary Qu Yang, and Youping Deng. A comparative study of different machine learning methods on microarray gene expression data. *BMC genomics*, 9(S1):S13, 2008.

[19] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[20] J Ross Quinlan. C4.5: Programs for machine learning. 1993.

[21] Marko Robnik-Šikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, volume 5, pages 296–304, 1997.

[22] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[23] Haijian Shi. *Best-first decision tree learning*. PhD thesis, The University of Waikato, 2007.

[24] Stephane Wenric and Ruhollah Shemirani. Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, 9:297, 2018.

[25] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. Data mining: practical machine learning tools and techniques with java implementations. pages 553 – 571, 2017.

[26] Ian H. Witten, Eibe Frank, and and Christopher J. Pal Mark A. Hall. Data mining: practical machine learning tools and techniques with java implementations. pages 110 – 113, 2017.

[27] Ian H. Witten, Eibe Frank, and and Christopher J. Pal Mark A. Hall. Data mining: practical machine learning tools and techniques with java implementations. pages 105 – 110, 2017.

[28] Ian H. Witten, Eibe Frank, and and Christopher J. Pal Mark A. Hall. Data mining: practical machine learning tools and techniques with java implementations. pages 215 – 217, 2017.

[29] Jiucheng Xu, Lin Sun, Yunpeng Gao, and Tianhe Xu. An ensemble feature selection technique for cancer recognition. *Bio-medical materials and engineering*, 24(1):1001–1008, 2014.