Gene Classification and Pattern Analysis Using Data Mining and Machine Learning Techniques

by

Aiman Jabeen Ramisa
17101012
Ananna Hossain
17101026
Sk Md Injamul Islam
17101169
Ponuel Mollah Swadesh
17101299
Md. Toushif Islam
17101032

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Aiman Jabeen Ramisa
17101012

Ananna Hossain.

Ananna Hossain
17101026

Sk Md Injamul Islam
17101169

Ponuel Mollah Swadesh

Ponuel Mollah Swadesh
17101299

MD. Tousif Islam

Md. Toushif Islam
17101032

Approval

The thesis/project titled “Gene Classification and Pattern Analysis Using Data Mining and Machine Learning Techniques” submitted by

1. Aiman Jabeen Ramisa (17101012)

2. Ananna Hossain (17101026)

3. Sk Md Injamul Islam (17101169)

4. Ponuel Mollah Swadesh (17101299)

5. Md. Toushif Islam (17101032)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June , 2021.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez
Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University, ou=CSE, email=zavid.parvez@bracu.ac.bd, c=US
Date: 2021.06.11 11:24:27 +10'00'

Dr Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Dr Md Anisur Rahman
Academic
School of Computing and Mathematics
Charles Sturt University

Program Coordinator:
(Member)

Dr Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

We are ensuring quality and integrity of our research. We respect the confidentiality and anonymity of our research respondents. Our research is independent and impartial.

Abstract

Gene classification and pattern extraction from gene sequence data is essential in understanding different gene sequence features. The field of gene expression data analysis has grown in the past few years from being purely data-centric to integrative, aiming at complementing microarray analysis with data and knowledge from diverse available sources. Since then, it has been used for various science fields, including the discovery of new drugs, identification of protein coded genes by analyzing and separating exons from the main sequence, phenotype prediction based on gene expression. The paper presents an application of gene classification from gene sequence data using data mining and machine learning techniques. Our research's main goal is to compare different machine learning approaches based on time of execution, and overall efficiency by testing them on different microarray data sets of gene sequence and determining the best approach for gene classification. Eight different machine learning techniques have been tested on eleven different gene expression datasets, and the results are compared and improved using the feature selection method. Moreover, we perform pattern analysis on some gene expression datasets using a J48 decision tree outcome, after applying feature selection.

Keywords: Classification; Feature Selection; Accuracy; Pattern Extraction; Gene Classification; Application of Machine Learning

Dedication

Dedicated to all our beloved ones for their inspiration and support.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our Supervisor, Dr Mohammad Zavid Parvez Sir, and to our Co-Supervisor, Dr Md Anisur Rahman Sir, for their kind support and advice in our work. They helped us whenever we needed help.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

DL	Deep Learning
$J48$	C4.5 Decision Trees
KNN	K-nearest Neighbours
LMT	Logistic Model Trees
LR	Logistic Regression
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machine
U	Undetermined

Chapter 1

Introduction

Research on gene sequence data has received a great deal of attention in recent years. The principal motivation behind that attention is to recover helpful patterns and perform analysis on gene sequence data. The patterns are of utmost significance when it comes to treating ailments and anticipating illnesses. Machine learning has played a significant role in determining such patterns over the years. Gene sequence data is usually very high dimensional data requiring high computation to determine gene sequence patterns. The approach with machine learning is to train the machine so that when it is tested with similar data, it can efficiently give accurate results. Such learning can be supervised, semi-supervised, unsupervised, or reinforcement learning depending on the dataset available and the type of work that we want the machine to learn [25].

In unsupervised learning, the machine is given an unlabeled dataset, and it tries to extract some feature based on the type of pattern given. In supervised learning, labeled dataset is given and the machine is trained using that, and upon testing on similar datasets, the accuracy of the algorithm is measured for classification.

In semi-supervised learning, the algorithm is given a few labeled data and large amount of unlabeled data to first try to extract feature and then check accuracy of the same algorithm. With reinforcement learning, the artificial intelligence agent is provided feedback in every step of the way on the basis of how it works and ways to improve in order to choose the optimal path to the target.

In order to carry out machine learning algorithms smoothly, data mining is essential. Data mining has various categories. Clustering data and anomaly detection are a few, but one of the most important data mining techniques is data pre-processing. With the data pre-processing step, the dataset is reconstructed in such a way that only the features that are relevant in determining the outcome are kept, while the rest are discarded, in such a way that the computation time is reduced and the algorithm is made more accurate by discarding the noise, or redundant data [17].

1.1 Research Problem

Extracting patterns from gene sequence has a number of benefits in the field of medical science. As mentioned earlier, the different patterns generated from gene sequence can be used to treat illnesses by the creation of different drugs. It can also be used to separate healthy genes from unhealthy ones. In recent times, given the current global pandemic situation created due to Covid-19, the entire world has been drawn to a standstill. In such a crisis, people are only looking for a ray of hope in the production of a fully functional vaccine. In order to create this vaccine, pharmacists should be able to distinguish between healthy genes and covid-affected genes. The chemical and biological properties of such genes need to be properly studied and understood if a useful drug is to be created. According to Collins et al. [3], genomic medicine holds the ultimate promise of revolutionizing the diagnosis and treatment of many illnesses. Furthermore, Mattick et al. [15] stated that currently, genome sequencing is having the greatest impact in stratifying cancer, characterising genetic disease, and providing information about an individual's likely response to treatment. Here, it is discussed that advances in both computational and sequencing technologies enable an ever-increasing capacity for accurate diagnosis of existing disease, and development of effective and targeted treatment strategies and also offer opportunities to assess predisposition to disease, potentially prompting more focused clinical monitoring and lifestyle changes. Not only for virus-infected patients, but also to categorise different abnormalities in people and to detect various diseases in human beings, the study of gene sequence to classify patterns is extremely useful. To combat these problems, our paper compares different machine learning techniques using data mining based on accuracy and time to classify various gene sequence datasets and extract patterns from them.

1.2 Research Objectives

The main goal of our research is to compare 8 different machine learning algorithms before and after applying data pre-processing based on time of execution and overall accuracy by testing them on 11 different microarray datasets of gene sequence and also extract patterns from 3 datasets using the decision trees extracted after pre-processing. This research is supposed to be a helping tool to get the most time-efficient and accurate technique of gene classification from such critical gene sequence and to analyse patterns that are extracted from the datasets which can help in the creation of different drugs and vaccines, to predict if a person is healthy or unhealthy given the gene sequence, to determine abnormalities in human beings and many other applications in the field of medical science.

Chapter 2

Literature Review

The classification of gene sequences has previously been studied in many different ways. The approach with machine learning is versatile to learn valuable portrayals of hereditary ideas when given just occurrences, as opposed to express definitions, of those ideas from quality grouping information [2]. The paper mainly looked into the empirical learning approach (supervised learning) using a training set. In any case, it is hard for such a way to deal with be both exceptionally delicate (foreseeing not many bogus negatives) and profoundly explicit (anticipating not many bogus positives) via preparing. But due to the large volume of data available in open repositories, predictions can be made more reliable. Ca et al. [20] used Support Vector Machine (SVM) classifier for gene classification. Here, mutual information (MI) between the genes and the class label has been utilized for distinguishing informative genes, and the testing capacity is assessed utilizing Leave-one-Out Cross-Validation (LOOCV) technique. This process is quite time-consuming, but classification accuracy is high. As indicated by this, the fundamental issue with this quality grouping is include determination, and since measurable statistical approaches are utilized to foster the model of the classifier, these are typically incapable to characterize an example if the outflows of qualities are marginally not the same as the predefined profile and accordingly bring about an unyielding characterization framework. Gene selection improves training time, as well as classification accuracy, by reducing the dimension of gene expression data and removing a large number of irrelevant genes [20].

Singh et al. [19] gives an overview of different feature selection techniques of gene expression data for cancer classification. As per this, information mining calculations assume a significant part in quality characterization as it is the most broadly utilized way to deal with accomplish significant quality gene expression data features. Model-based methodology has been proposed to assess the entropy on the model, rather than on the actual information. They also used marker panel rank based scheme information gain to analyze, but this information gain was not accurate with large groups. Also, among the two types of feature selection techniques that have been discussed here, filter methods and wrapper methods, the latter requires extensive computation to search for the best features and takes more time [19]. Another feature selection technique has been proposed by Liu et al. [26]. This paper talks about the importance of feature selection to identify and separate disease-related information from a large amount of data and noise. Double RBF-kernels are used here for gene expression data classification for cancer. The modified method has been tested on either multi-class phenotypes or two-class phenotypes. This method has a reduced run time, as well as a higher accuracy than single RBF-kernels. This has higher exactness contrasted with some generally utilized filter-based feature ranking methods, to be specific X^2-Measurement, Maximum Relevance and Minimum Redundancy (MRMR), Relief-F, Information Gain and Fisher Score. But this approach is more suited to datasets with a small sample size.

Another paper proposed a cost-sensitive rotation forest algorithm for gene expression data classification [24]. Mainly miss-classification cost, test cost, and rejection cost are embedded into the rotation forest algorithm to make this method cost-sensitive. By reducing these classification costs, the classification of genes can be made more reliable. This approach gives a lower cost than support vector machine (SVM) and extreme learning machines (ELMs). The conventional rotation forest algorithm is improved using this to deal with linearly inseparable data. Although this might be a cost-effective approach, classification accuracy is significantly hampered as the miss-classification risk increases without considering the rejection cost and the rejection strategies. Also, the C-RoF method mentioned here alters the original way of splitting the dataset and reduces accuracy, as it prefers splitting with attributes in low-cost direction. Li et al. [10] explored the intrinsic structure of the features so as to classify the microarray data more accurately through manifold learning method by mapping the gene expression data to a low dimensional space. This can project the gene expression data into a subspace with high intra-class compactness and inter-class separability. One can map the observed high dimensional data to a low dimensional essential feature manifold space with manifold learning methods. A supervised version of an LLE, namely locally linear discriminant embedding (LLDE), is proposed here for tumor characterization. A vector interpretation and distance rescaling model is built to improve the acknowledgment exactness of the first LLE, which makes the samples with various class marks very much isolated and makes the samples having a place with a similar class nearer. This LLDE procedure altogether weakens the little sample size issue and the detached component issue contrasted with other broadly utilized feature extraction algorithms. It has been extremely powerful in extricating the discriminant highlights from the gene expression information of the tumor dataset. Yet, it depends on a suspicion that the gene expression data can likewise be tested from a manifold influenced by complicated environmental impact and internal gene factors which probably won't be altogether evident truly. Additionally the highlights with various class names will be arranged effectively, however the key qualities can't be chosen with the proposed calculation since LLDE centers around protecting the locality of manifold and improving the grouping capacity

In another paper, a Group K-Singular Value Decomposition (Group K-SVD) has been proposed which is a new sparse learning model for the classification of gene expression data [28]. This technique learns the ideal dictionary and inadequate portrayal from the preparation information, and after that appoints the out-of-test data to the class with the closest centroid. By utilizing a gathering update methodology during the dictionary update stage, it decreases the excess of over-complete word reference. To solve the optimization problem, He et al. [28] also developed a

Multivariate Orthogonal Matching Pursuit (MOMP) algorithm which updates the sparse coefficient in an adaptive group manner instead of one-by-one. This strategy has been tried on certifiable gene expression datasets, just as on a high dimensional dataset. This method is very powerful and has end up being successful and proficient contrasted and some cutting edge characterization calculations. However, this strategy isn't versatile to bidirectional sparse learning issues. García-Díaz et al. [30] proposed a Gathering Hereditary Calculation (GGA) to take care of a maximally assorted gathering issue, which has been applied for the grouping of a skewed database of 801 samples of gene expression RNA arrangement information in 5 sorts of cancer. The precision was discovered to be marginally higher in balanced databases than in unbalanced ones. The normal precision of the order is improved when a few of these classifiers are joined in a democratic framework. Yet, there is a constraint to this proposed calculation, which is the chance of inadequate investigation of the solution space. Since the pursuit space is gigantic, legitimate investigation of the hunt space isn't ensured in this.

He et al. [29] discussed how stereotypical gene expression information normally present a high dimensional imbalanced trademark, which represents an extreme test for conventional AI strategies to develop a hearty classifier performing admirably on both the minority and larger part classes. Relief is considered especially suit to deal with high-dimensional issues as it is perhaps the most successful feature weighting methods. Here, another technique, named imRelief, is proposed for productively taking care of high-dimensional imbalanced gene expression data that can fix the inclination towards the majority classes and consider the dissipated distributional trait of minority class tests in the process of estimating feature weights. [12]. This strategy beats other common Relief-based algorithms as far as different assessment metrics according to the test result. Yet, this strategy is just successful for microarray gene expression data with class imbalance distribution. Priyam et al. [14] compared the classification of several decision tree algorithms on low dimensional datasets. Much research has been carried out to figure out the best possible approach to pattern extraction from the gene sequence. Most of such research lack efficiency and accuracy [26][24][30].

Compared to the previous works, our research's main objective is to try different machine learning algorithms to test on different micro-array datasets for gene expression analysis and get the most time-efficient and accurate method for this. Also, by applying feature selection techniques, the results are compared to determine the best possible machine learning technique for such meaningful and critical data. To save time, we have used a filtering method (Information Gain) for feature selection. Our proposed method compares decision trees, along with other classification models on high dimensional gene expression data. The extracted patterns from decision trees can then be used for various medical science applications. The contributions of the paper is as follows:

- Performance evaluation of various classification techniques on high dimensional gene expression datasets,
- Performance improvement of classification techniques on high dimensional gene expression data using feature selection method,

- Pattern analysis on gene expression datasets.

Chapter 3

Proposed Method

This section presents a framework to discover an efficient machine learning classification algorithm in terms of accuracy, F1 score, and execution time from eight different classifiers using eleven gene expression datasets. A flow chart of the proposed method can be seen in Figure 3.1. The procedure is carried out in five basic steps.

- Data Collection
- Data Pre-processing
- Classification
- Evaluation and Comparison
- Pattern Analysis Based on Decision Trees

In the beginning, we have discussed the process for our data collection, then we have discussed the feature selection technique that we used for data pre-processing. Next, we apply classification algorithms to the dataset and then perform evaluation and comparison of the results. Finally, we performed pattern analysis on different gene expression datasets.

3.1 Data Collection

In this research, eleven gene expression datasets have been used as shown in Table 3.1. The datasets BC, LK, BR, TR, GT, PC, and LV, are collected from the paper of Feltes et al. [27]. These datasets come from the Curated Microarray Database (CuMiDa), a repository containing 78 handpicked cancer microarray datasets, extensively curated from several studies from the gene expression omnibus (GEO), solely for machine learning. The LP dataset was collected from the paper of Cheng et al. [11]. The DB dataset was collected from the paper of Hoshida et al. [7]. The NV and LC datasets were collected from the paper of Monti et al. [4]. Compared to the DB, NV, and LC datasets, the rest are extremely high dimensional datasets containing thousands of gene expression attributes. The BC dataset contains gene expression data for 6 types of breast cancer (including normal cell); similarly, the LK dataset contains data for 5 types of leukemia; the LP dataset contains data for 2 types of cancer; the BR dataset contains data for 5 types of brain cancer (including

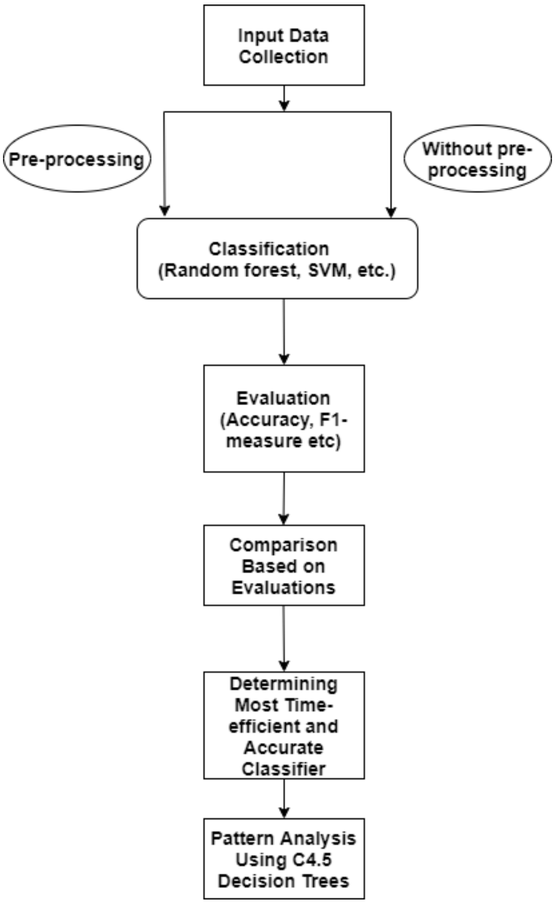
Input Data Collection
Pre-processing
Without pre-processing
Classification (Random forest, SVM, etc.)
Evaluation (Accuracy, F1-measure etc)
Comparison Based on Evaluations
Determining Most Time-efficient and Accurate Classifier
Pattern Analysis Using C4.5 Decision Trees

Figure 3.1: The flowchart of the proposed method

normal tissues); the DB dataset contains data for 3 types of diffuse large B-cell lymphoma; the NV dataset contains data for 4 types of organ tissues; the LC dataset contains data for 4 types of lung cancer; the TR dataset contains data for 2 types of throat tissues; the GT dataset contains data for 2 gastric type tissues; the PC dataset contains data for 2 types of pancreas tissues, and the LV dataset contains data for 2 types of liver tissues.

3.2 Data Pre-processing

To evaluate and compare each classifier on the datasets, the effect of data mining is to be considered. Each classifier is applied to the datasets before applying any pre-processing on the datasets. Then, again each classifier is applied to the datasets after pre-processing. To pre-process the high dimensional datasets, one of the most important method is used, which is feature selection. Singh et al. [19] already mentioned that wrapper methods require excessive computation, which requires more time, we used a filter method for feature selection. This feature selection is made using the highest 50 attributes ranked based on information gain. To calculate the information gain of each attribute, firstly, the entropy is to be calculated. The entropy is simply the amount of variance in the data. The formula to calculate entropy

Table 3.1: A brief description on gene expression datasets

Datasets	Attributes	Records	Classes
Breast Cancer GSE45827(BC)	54676	151	6
Leukemia GSE9476(LK)	22284	64	5
Lung And Prostate Cancer(LP)	12534	32	2
Brain GSE50161(BR)	54676	130	5
DLBCL(DB)	661	141	3
Novartis BPLC Multi(NV)	1000	103	4
Lung Cancer Types(LC)	1000	197	4
Throat GSE42743(TR)	54676	103	2
Gastric GSE79973(GT)	54676	20	2
Pancreatic GSE16515(PC)	54676	51	2
Liver GSE62232(LV)	54676	91	2

is given by:

$$E = -\sum_{i=1}^{c} P_i \log_2 P_i \tag{3.1}$$

Here, c is the number of classes, P$_i$ is the probability of randomly picking an element of class i (i.e., the proportion of the dataset made up of class i). The information gain tells the importance of an attribute. This gain is calculated by deducting the weighted entropies of each branch from the original entropy. The formula for information gain is given by :

$$G(T, A) = E(T) - \sum_{v \in A} \frac{|T_v|}{T} . E(T_v) \tag{3.2}$$

Here, T is the target class column, A is the attribute to be tested, and v is each value in the column of A. The feature selection method is carried out in Weka (Waikato Environment for Knowledge Analysis) software using the AttributeEvaluator method InfoGainAttributeEval and the Ranker method. Based on the highest ranks of information gain, only the first 50 attributes were selected after pre-processing, which greatly reduced the datasets' dimension. The WEKA software is open source software for data mining and the system is written using object oriented language, java [13]. Association, filtering, classification, clustering, visualization, regression, etc. can be performed by implementing state-of-the-art data mining and machine learning algorithms using WEKA.

3.3 Classification

Using the WEKA software, eight classifiers were applied on each of the datasets, where 75% of the dataset was used for training, and the remaining 25% were used for testing in a supervised learning manner. These eight classifiers are Support Vector Machine (SVM), J48 (C4.5 Decision Trees), Random Forest (RF), Logistic Regression (LR), Deep Learning (DL), Naive Bayes (NB), K-nearest Neighbours (KNN), and Logistic Model Trees (LMT).

- SVM: The SVM classifier has been chosen as it is one of the most widely used classifiers for gene expression analysis, with very good accuracy [12]. While training an SVM classifier, the process is to identify a reproducible hyperplane that maximizes the distance between the support vectors of the classes. So, the optimal hyperplane is that which maximizes the distance between class labels. The SVM classifier uses kernel function to help with classification. These kernel functions allow the SVM classifiers to handle nonlinearly separable data sets and to incorporate prior knowledge, the kernels can be defined on inputs that are not vector, and kernels provide a mathematical formalism for combining different types of data, which allow SVM to be used in critical biological applications, such as classifying DNA and protein sequences [6]. The SVM classifier is mostly suitable for handling binary classes, but we plan to see how accurately it classifies the datasets of multi-class, as well as, binary classes.

- J48: Among the classifiers used, J48 and RF use decision trees for classification [14]. Decision trees use the value of independent attributes to find the values of the dependent attributes. The possible after-effect, including chance event results, resource costs, and utility of a tree-like graph is used to make decisions using decision trees [13]. The J48 algorithm is used to see how effectively a single decision tree can do such classification. This type of decision tree (C4.5) is quite commonly used for classification purposes and is a univariate decision tree, where splitting is done by using one attribute at internal nodes [13].

- RF: The random forest classifier is used to give more accurate results than single decision trees and can be used on large datasets. This approach combines multiple decision trees for classification. This classifier takes a random subset from the features, builds a decision tree based on that subset, and repeats this process for many decision trees for several other subsets. This reduces the bias of a single subset, and this approach is also effective for missing values in data and categorical data. This algorithm overcomes the problem of over-fitting by combining or averaging the results of different decision trees [22]. This approach is aimed to give high accuracy even without pre-processing the data, but requiring high computational time than J48.

- LMT: LMT also uses tree, but these type of trees are generated based on logistic regression, and the results are mostly similar to logistic regression models but takes more time to build the trees [5]. Logistic regression models are used at the nodes to make decisions of such trees [23].

- NB: The NB and KNN classifiers do not use any model for classification; instead, it depends on memory [8]. The NB classifier is one of the most widely used Bayesian learning methods, and it tries to simplify the learning by assuming that attributes are independent given class. It is a simple algorithm which is easy to build and particularly useful for large datasets. In order to see how this simplification impacts the classification of gene expression data, this classifier is chosen.

- KNN: The KNN classifier classifies the result of a new instance query based on the majority of the k-nearest neighbor category. It is easy to interpret classification results using KNN, and it has low calcultaion time [1]. We have used KNN with the value of k=1.

- DL: The DL classifier uses neural networks to classify the data [21] and is used to determine how neural networks behave in gene expression classification. Deep learning is just very big neural networks requiring a lot of data, and depicts the way a human brain works. This calculation permits computational models that are made out of various preparing layers to learn portrayals of information with numerous degrees of reflection and finds many-sided structure in huge informational indexes by utilizing the backpropagation calculation to demonstrate how a machine should change its inner boundaries that are utilized to register the portrayal in each layer from the portrayal in the past layer[18].

- LR: Finally, the LR model has not been used much previously for gene sequence classification, so we plan to use this on our datasets. LR models mainly work with binary class and uses the logistic function for classification [9]. But we intend to use this to see how effectively this model classifies both multi-class and binary class datasets of gene expression.

3.4 Evaluation and Comparison

In order to compare the classifiers to determine the most time-efficient and accurate classifier, these classifiers were evaluated in terms of accuracy, F1-score, and execution time. The definition of accuracy, F1-score, and execution time is given below.

3.4.1 Accuracy

This determines how correctly the labels have been classified from the total dataset [31].

$$Accuracy = \frac{CorrectlyClassifiedSamples}{TotalSamples} \tag{3.3}$$

3.4.2 F1-measure

This score is based on two other measuring techniques, namely, recall and precision. The recall estimates the extent of real positives that were distinguished accurately dependent on the confusion matrix of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [16].

3.4.3 Time

This time is the summation of the time taken to build a model and the time taken to test the model on test split generated in WEKA.

Based on these three measures, the classifier with the best score for gene sequence classification is determined both before and after feature selection.

3.5 Pattern Analysis Using Decision Tree Outcome

In this section, we perform some pattern analysis. The decision trees produced by the J48 classifier (C4.5 Decision Trees) on three datasets after applying feature selection are used to generate patterns.

Chapter 4

Experimental Result and Discussion

This section discusses the experimental results of the classification techniques on the gene expression datasets in terms of accuracy, F1-score, and execution time. We also perform pattern analysis using the J48 decision tree on three datasets. We discuss the results before feature selection and after feature selection.

4.1 Evaluation Before Feature Selection

Table 4.1 gives each classifier's classification accuracy results on each dataset before feature selection. Before using information gain on the features, for the BC dataset, LR and LMT classifiers gave highest accuracy of 94.74%, but LR took less time than LMT. The confusion matrix generated by LR that among the 25% test set, the model correctly classified 11 out of 11 basal types, 4 out of 5 of HER type, 1 out of 1 of cell_line type, 3 out of 4 of the normal type, 9 out of 9 of luminal_A type and 8 out of 8 of luminal_B type. For the LK dataset, the highest accuracy of 100% was generated by the RF classifier. The confusion matrix revealed no data of type Bone_Marrow_CD34 in the test set, the 5 data for Bone_Marrow and AML type each were classified correctly, so were the 4 PB types and 2 PBSC_CD34 types in the test set. SVM, LR, DL, NB, and LMT produced 100% accuracy, and SVM produced the least time for the LP dataset. The model correctly classified 5 lung cancer type data and 3 prostate cancer type data in the test set.

For the BR dataset, SVM produced the highest accuracy of 100%. This classifier correctly classified 11 ependymoma types, 10 glioblastoma types, 4 medulloblastoma types, 3 normal types, and 4 pilocytic_astrocytoma type of brain cancer data in the test set. For the DB dataset, RF produced the highest accuracy of 100%. The model correctly classified 14 OxPhos types, 12 BCR types, and 9 HR type of DLBCL in the test data. For the NV dataset, SVM, LR, DL, KNN, and LMT produced the highest accuracy of 100%, with KNN taking the lowest time. These classifiers correctly classified 8 breast types, 7 prostate types, 10 lung types, and 1 colon type tissue in the test data. For the LC dataset, LR, DL, NB, and LMT produced the highest accuracy of 93.88%, with NB taking the lowest time. The confusion matrix revealed that the model correctly classified 30 out of 32 AD type, 4 out of 5 NL type, 7 out of 7 SQ type, and 5 out of 5 COID type of lung cancer data. For the TR dataset, SVM, RF, LR, and LMT gave the highest accuracy of 100%, with SVM taking the least time among them. These models correctly classified 24 oral_cavity_cancer type and 2 normal type data in the test set. For the GT dataset, RF, LR, KNN, and LMT gave the highest accuracy of 80%, with KNN taking the least time. According to the confusion matrix, the model correctly classified 2 out of 3 adenocarcinoma types and 2 out of 2 normal type data in the test set. For the PC dataset, LR and DL gave the highest accuracy of 92.31%, with DL taking less time. The model correctly classified 9 out of 10 tumoral types and 3 out of 3 normal type tissue for the test data as per the confusion matrix. SVM, LR, KNN, and LMT gave 100% accuracy, with SVM taking the lowest time for the LV dataset. The model correctly classified 4 normal types and 19 HCC type data in the test set.

The average accuracies of the classifiers before pre-processing is given in Figure 4.1. Before the datasets were pre-processed, the classifier which generated the highest average accuracy was LR, with an average accuracy of 94.51%, with LMT and SVM being second and third highest (93.81% and 91.79%). The classifier which performed the worst was NB, with an average accuracy score of 74.72%, with J48 and DL being the second and third lowest (76.85% and 85.73%). The average accuracy of RF was 91.14%, and the average accuracy of KNN was 86.27%.

Table 4.1: Accuracy of the Techniques Before Feature Selection

Dataset	**SVM** (%)	**J48** (%)	**RF** (%)	**LR** (%)	**DL** (%)	**NB** (%)	**KNN** (%)	**LMT** (%)
BC	92.11	78.95	89.47	94.74	81.58	63.16	84.21	94.74
LK	93.75	68.75	100.00	87.50	87.50	62.50	87.50	87.50
LP	100.00	62.50	87.50	100.00	100.00	100.00	75.00	100.00
BR	100.00	78.13	93.75	96.88	87.50	90.63	81.25	96.88
DB	97.14	80.00	100.00	94.29	94.29	88.57	91.43	94.29
NV	100.00	88.46	96.15	100.00	100.00	96.15	100.00	100.00
LC	89.80	83.67	91.84	93.88	93.88	93.88	91.84	93.88
TR	100.00	92.31	100.00	100.00	80.77	65.38	88.46	100.00
GT	60.00	40.00	80.00	80.00	60.00	60.00	80.00	80.00
PC	76.92	76.92	76.92	92.31	92.31	53.85	69.23	84.62
LV	100.00	95.65	86.96	100.00	65.22	47.83	100.00	100.00

Table 4.2 gives the weighted F1-measure results of each classifier on each dataset before feature selection. Figure 4.2 gives the average F1-score of the classifiers before pre-processing. Before the datasets were pre-processed using information gain, the classifier which generated the highest average F1-measure was LR, with an average weighted F1-measure of 0.944, with LMT and RF being second and third highest (0.937 and 0.919). The classifier which performed the worst was J48, with an average

Table 4.2: F1 Score of the Techniques Before Feature Selection

Dataset	SVM	J48	RF	LR	DL	NB	KNN	LMT
BC	0.92	0.77	0.89	0.95	0.82	U	0.84	0.95
LK	0.94	0.70	1.00	0.87	0.87	0.76	0.88	0.87
LP	1.00	0.63	0.88	1.00	1.00	1.00	0.75	1.00
BR	1.00	0.77	0.93	0.97	0.87	0.88	0.81	0.97
DB	0.97	0.80	1.00	0.94	0.94	0.88	0.92	0.94
NV	1.00	0.89	0.96	1.00	1.00	0.96	1.00	1.00
LC	0.90	0.83	0.92	0.94	0.94	0.94	0.92	0.94
TR	1.00	0.93	1.00	1.00	0.85	0.94	0.91	1.00
GT	0.60	0.40	0.78	0.80	0.60	U	0.80	0.80
PC	0.75	0.75	U	0.93	0.93	U	0.63	0.85
LV	1.00	0.95	0.84	1.00	0.69	0.91	1.00	1.00

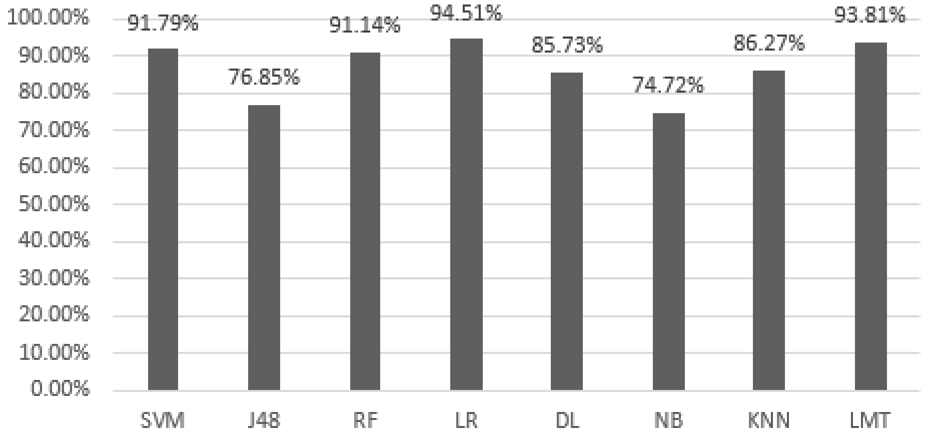

Figure 4.1: Average Accuracies of the Classifiers Before Pre-processing

weighted F1-measure of 0.766, with KNN and DL being the second and third lowest (0.859 and 0.864). The average weighted F1-measure of SVM was 0.916, and the average weighted F1-measure of NB was 0.911. The highest result is similar to that of accuracy. The main reason for the difference in lowest value results between accuracy and F1-measure was that WEKA could not determine weighted F1-measure for BC, GT, and PC datasets using NB classifier due to the inconsistent distribution of classes in train and test set. Similarly, the F1-measure for PC dataset was also undetermined (U) using RF classifier.

As for computational time, LMT took the most time with an average computation time of 39.75s, and KNN took the least time with an average computation time of 0.49s. the average computational time for SVM was 0.77s, J48 was 1.34s, RF was 2.72s, LR was 16.17s, DL was 8.44s, and NB was 1.01s. Table 4.3 gives the computational time taken by each classifier before pre-processing for each dataset. Figure 4.3 gives the average time taken by the classifiers before pre-processing.

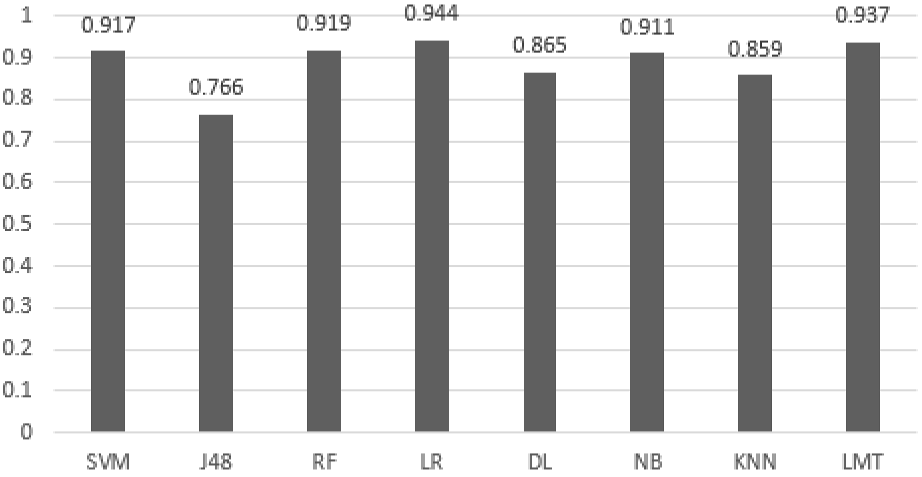

Figure 4.2: Average F1-score of the Classifiers Before Pre-processing

Table 4.3: Computational Time of the Techniques Before Feature Selection

Dataset	**SVM** (s)	**J48** (s)	**RF** (s)	**LR** (s)	**DL** (s)	**NB** (s)	**KNN** (s)	**LMT** (s)
BC	3.62	7.68	9.56	75.47	16.64	4.30	2.79	182.27
LK	1.00	0.92	3.28	12.48	27.93	0.81	0.31	22.11
LP	0.09	0.05	1.04	0.87	0.92	0.24	0.05	1.58
BR	2.12	3.8	8.09	74.69	15.87	3.59	1.09	203.00
DB	0.22	0.12	0.44	0.70	2.12	0.23	0.24	1.21
NV	0.08	0.05	0.15	0.32	1.31	0.06	0.02	1.12
LC	0.09	0.15	0.24	0.79	2.38	0.10	0.07	2.25
TR	0.71	1.24	2.26	5.49	19.91	0.81	0.38	11.69
GT	0.07	0.06	1.68	1.18	0.78	0.12	0.04	1.70
PC	0.21	0.31	1.69	2.25	1.68	0.31	0.08	3.84
LV	0.21	0.33	1.52	3.64	3.25	0.57	0.29	6.48

4.2 Evaluation After Feature Selection

Table 4.4 gives the classification accuracy results of each classifier on each dataset after feature selection. After using information gain on the features, for the BC dataset, the SVM classifier gave the highest accuracy of 97.37%. The confusion matrix generated by LR that among the 25% test set, the model correctly classified 11 out of 11 basal types, 4 out of 5 of HER type, 1 out of 1 of cell_line type, 4 out of 4 of the normal type, 9 out of 9 of luminal_A type and 8 out of 8 of luminal_B type. For the LK dataset, except DL, all the other classifiers produced 100% accuracy, and among them, J48 took the least time. This classifier successfully classified the test data as per the type of leukemia. For the LP dataset, all the classifiers produced 100% accuracy, and J48 and KNN took the least time. All the classes of the test set were classified correctly.

For the BR dataset, LR and LMT produced the highest accuracy of 100%, with LR taking the lower time. All the classes of the test set were classified correctly for this

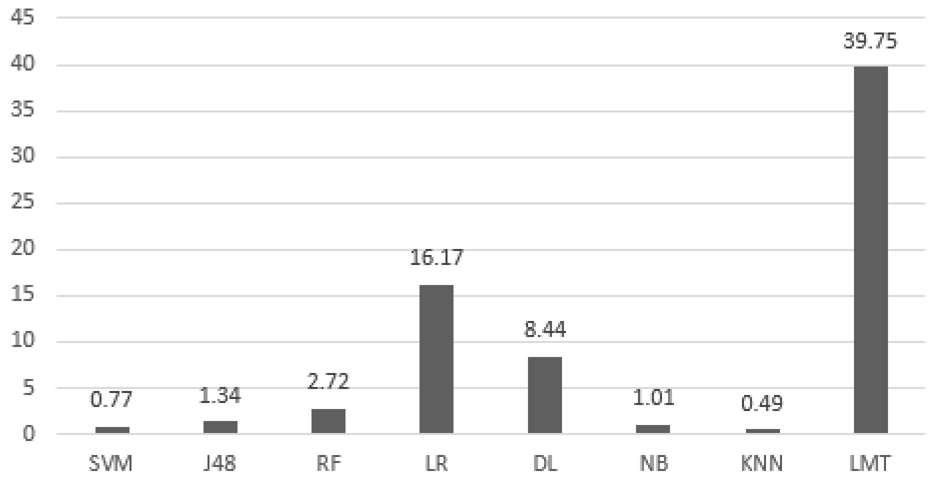

Figure 4.3: Average Computational Time (in seconds) of the Classifiers Before Pre-processing

dataset, as well. For the DB dataset, LR and LMT gave the highest accuracy of 94.29%, with LR taking the lower time. This classifier correctly classified 14 out of 14 OxPhos types, 10 out of 12 BCR types, and 9 out of 9 HR type data. The highest accuracy slightly decreased before pre-processing due to this dataset's smaller dimension (also LC dataset). For the NV dataset, RF, LR, DL, and LMT gave the highest accuracy of 100%, with LR taking the least time among them. These models correctly classified all the types of tissues in the test set. For the LC dataset, LR, NB, and LMT models gave the highest accuracy of 89.80%, with NB taking the lowest time. The models correctly classified 29 out of 32 AD type, 4 out of 5 NL type, 6 out of 7 SQ type, and 5 out of 5 COID type of lung cancer data. For the TR dataset, SVM, J48, RF, LR, and LMT gave 100% accuracy, with SVM and J48 taking the lowest time to classify all the test data correctly.

For the GT dataset, except NB, all the classifiers gave 100% accuracy, with J48 and KNN taking the lowest time to classify the entire test set correctly. For the PC dataset, J48, RF, and NB gave 100% accuracy, with J48 and NB taking the lowest time to correctly classify all the test data. For the LV dataset, all except DL gave 100% accuracy, with J48 taking the least time to classify the entire test set correctly.

Figure 4.4 gives the average accuracies of the classifiers after pre-processing. After the datasets were pre-processed, the classifier which generated the highest average accuracy was similar to that of before feature selection, which is LR, as well as LMT, with an average accuracy of 97.38%, with RF and SVM being close to second and third highest (96.92% and 96.54%). The classifier which performed the worst after feature selection was DL, with an average accuracy score of 91.14%, with J48 and NB being the second and third lowest (91.19% and 92.68%). The average accuracy of KNN was 95.12%. Overall, the average accuracies for each classifier improved after feature selection.

Table 4.5 gives the weighted F1-measure results of each classifier on each dataset after feature selection. Figure 4.5 gives the average F1-score of the classifiers after

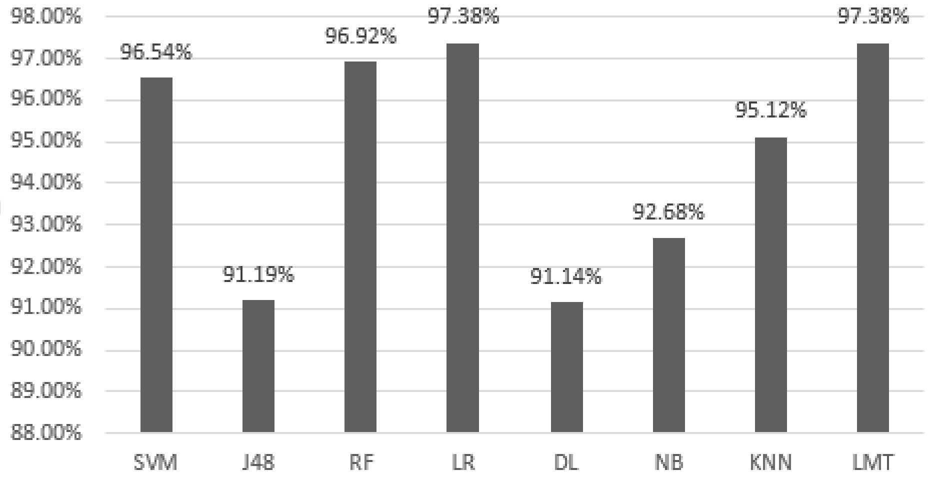

Figure 4.4: Average Accuracies of the Classifiers After Pre-processing

pre-processing. After the datasets were pre-processed using information gain, the classifier which generated the highest average F1-measure was similar to that of before feature selection, which is LR, as well as LMT, with an average weighted F1-measure of 0.974, with RF and SVM being close second and third highest (0.968 and 0.966). The classifier that performed the worst was J48, with an average weighted F1-measure of 0.913, with DL and NB being the second and third lowest (0.915 and 0.925). The average weighted F1-measure of KNN was 0.953. The highest result is similar to that of accuracy.The lowest score is similar to that of before feature selection.

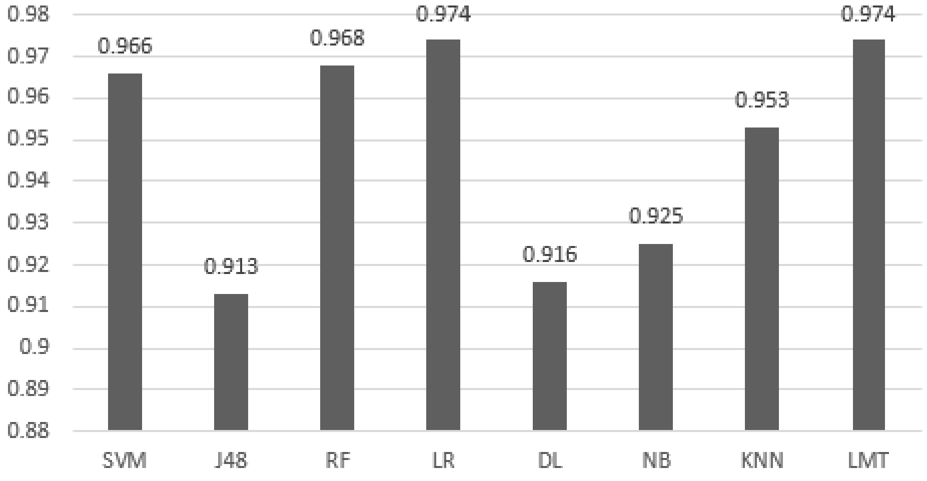

Figure 4.5: Average F1-score of the Classifiers After Pre-processing

As for computational time, DL took the most time with an average computation time of 2.39s, and KNN and J48 took the least time with an average computation time of

Table 4.4: Accuracy of the Techniques After Feature Selection

Dataset	**SVM** (%)	**J48** (%)	**RF** (%)	**LR** (%)	**DL** (%)	**NB** (%)	**KNN** (%)	**LMT** (%)
BC	97.37	78.95	92.11	94.74	84.21	78.95	89.47	94.74
LK	100.00	100.00	100.00	100.00	81.25	100.00	100.00	100.00
LP	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
BR	96.88	90.63	96.88	100.00	90.63	93.75	96.88	100.00
DB	91.43	82.86	91.43	94.29	91.43	88.57	91.43	94.29
NV	96.15	73.08	100.00	100.00	100.00	92.31	96.15	100.00
LC	87.76	77.55	85.71	89.80	87.76	89.80	87.76	89.80
TR	100.00	100.00	100.00	100.00	92.31	96.15	92.31	100.00
GT	100.00	100.00	100.00	100.00	100.00	80.00	100.00	100.00
PC	92.31	100.00	100.00	92.31	92.31	100.00	92.31	92.31
LV	100.00	100.00	100.00	100.00	82.61	100.00	100.00	100.00

Table 4.5: F1 Score of the Techniques After Feature Selection

Dataset	**SVM**	**J48**	**RF**	**LR**	**DL**	**NB**	**KNN**	**LMT**
BC	0.97	0.79	0.92	0.95	0.84	0.79	0.90	0.95
LK	1.00	1.00	1.00	1.00	0.83	1.00	1.00	1.00
LP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
BR	0.97	0.91	0.97	1.00	0.91	0.93	0.97	1.00
DB	0.91	0.83	0.91	0.94	0.91	0.89	0.92	0.94
NV	0.97	0.74	1.00	1.00	1.00	0.93	0.96	1.00
LC	0.88	0.78	0.85	0.90	0.88	0.90	0.88	0.90
TR	1.00	1.00	1.00	1.00	0.93	0.97	0.93	1.00
GT	1.00	1.00	1.00	1.00	1.00	0.78	1.00	1.00
PC	0.93	1.00	1.00	0.93	0.93	1.00	0.93	0.93
LV	1.00	1.00	1.00	1.00	0.85	1.00	1.00	1.00

0.01s. The average computational time for SVM was 0.20s, RF was 0.11s, LR was 0.12s, NB was 0.02s, and LMT was 0.16s. Table 4.6 gives the computational time taken by the classifiers after pre-processing for each dataset. After feature selection, the weighted F1-measure and computation time for each classifier improved on an average. Figure 4.6 gives the average computational time taken by the classifiers after pre-processing. As we had predicted that RF gives better results than J48, our results show a similar pattern. In RF, multiple decision trees are used, which gives a better result than a single decision tree. Also, LR and LMT have similar accuracy and F1-measure results, but LMT has a much higher computation time. DL, NB, and J48 are among the worst-performing classifiers compared to the rest on an average in the majority of cases. KNN has not too great nor too poor results. LR, SVM, and RF are among the best performing classifiers. LR gave the highest average accuracy and weighted F1-measure in all cases and improved time after pre-processing, which can be considered the best performing method for gene expression classification based on the results.

Table 4.6: Computational Time of the Techniques After Feature Selection

Dataset	**SVM** (s)	**J48** (s)	**RF** (s)	**LR** (s)	**DL** (s)	**NB** (s)	**KNN** (s)	**LMT** (s)
BC	0.08	0.02	0.06	0.12	1.96	0.01	0.00	0.19
LK	1.64	0.00	0.2	0.36	14.3	0.07	0.04	0.27
LP	0.03	0.01	0.09	0.07	0.88	0.02	0.01	0.10
BR	0.09	0.04	0.18	0.23	1.81	0.03	0.02	0.34
DB	0.04	0.02	0.15	0.16	1.57	0.02	0.01	0.26
NV	0.08	0.01	0.12	0.07	1.00	0.01	0.00	0.16
LC	0.10	0.04	0.18	0.16	1.88	0.03	0.02	0.27
TR	0.01	0.01	0.08	0.05	0.96	0.04	0.00	0.10
GT	0.03	0.00	0.04	0.01	0.22	0.00	0.00	0.03
PC	0.01	0.00	0.02	0.03	0.64	0.00	0.00	0.04
LV	0.08	0.01	0.05	0.04	1.05	0.03	0.05	0.05

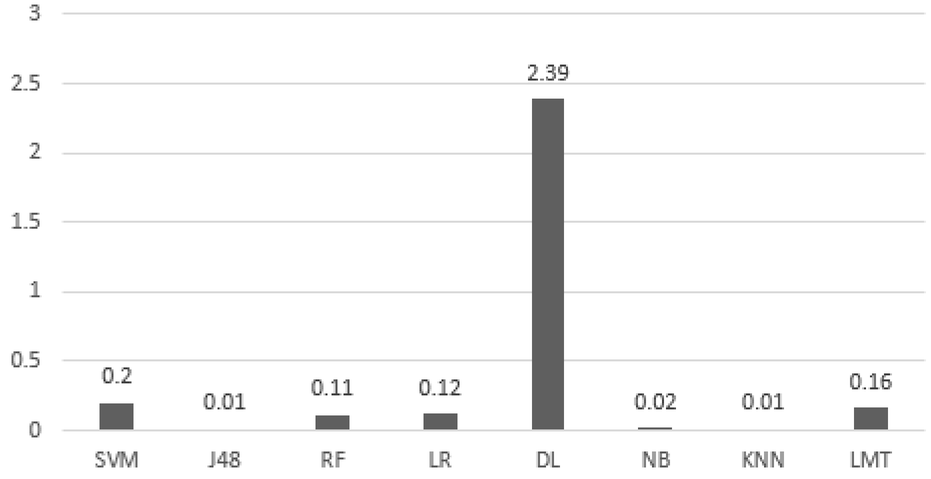

Figure 4.6: Average Computational Time (in seconds) of the Classifiers After Preprocessing

4.3 Pattern Analysis on Gene Expression Dataset

In this section, we carry out some pattern analysis with the J48 decision tree's outcome on TR, BC, and BR datasets. Note that, the number of records for TR, BC and BR is 103, 151, and 130, respectively. The number of attributes for each dataset is 54676. For pattern analysis, we considered 50 attributes for each dataset based on information gain that we discussed previously. The decision tree is built for each dataset using 10-fold cross-validation.

4.3.1 Pattern Analysis on TR dataset

From the decision tree on the TR dataset, we can see that when the value of attribute 205596_s_at is greater than 8.673253 and the value of attribute 204915_s_at is greater than 5.971706, then 65 records satisfy this pattern, and all of the 65 records have class value "oral cavity cancer". However, when the value of attribute 205596_s_at is less or equal to 8.673253 and the value of attribute 204915_s_at is less than or equal to 6.164087, then 27 records satisfy this pattern and 26 out of 27 records have class value "normal". But when the value of attribute 205596_s_at is less or equal to 8.673253, and the value of attribute 204915_s_at is greater than 6.164087, then 7 records satisfy this pattern, and 6 of them have the class value "oral cavity cancer".

4.3.2 Pattern Analysis on BC dataset

From the decision tree on BC dataset, we can see that when the value of attribute 221811_at is greater than 8.103477 and the value of attribute AFFX-CreX-3_at is greater than 13.57468, then 26 records satisfy this pattern, and all of them have class value "luminal_B". However, when the value of attribute 221811_at is greater than 8.103477, and the value of attribute AFFX-CreX-3_at is less than or equal to 13.57468, then 29 records satisfy this pattern, and all of them have class value "HER". But when the value of attribute 221811_at is less than or equal to 8.103477 and the value of attribute 209642_at is less than or equal to 6.941912 and the value of attribute 228241_at is less than or equal to 9.186783, then 8 records satisfy this pattern, and seven of them have class value "normal". When the value of attribute 221811_at is less than or equal to 8.103477 and the value of attribute 209642_at is less than or equal to 6.941912 and the value of attribute 228241_at is greater than 9.186783 and the value of attribute 236641_at is less than or equal to 4.176825, then 28 records satisfy this pattern and all of them have class value "luminal_A". When the value of attribute 221811_at is less than or equal to 8.103477 and the value of attribute 209642_at is greater than 6.941912, and the value of attribute 200795_at is less than or equal to 4.258904, then 14 records satisfy this pattern and all of them have class value "cell_line". But when the value of attribute 221811_at is less than or equal to 8.103477 and the value of attribute 209642_at is greater than 6.941912 and the value of attribute 200795_at is greater than 4.258904, then 41 records satisfy this pattern, and all of them have class value "basal".

4.3.3 Pattern Analysis on BR dataset

From the decision tree on the BR dataset, we can see that when the value of attribute 226872_at is greater than 9.468941 then 41 records satisfy this pattern, and all of the 41 records have class value "ependymoma". However, when the value of attribute 226872_at is less or equal to 9.468941 and the value of attribute 209109_s_at is less than or equal to 8.538174, then 13 records satisfy this pattern and all of 13 records have class value "normal". But when the value of attribute 226872_at is less or equal to 9.468941 and the value of attribute 209109_s_at is greater than 8.538174 and 217762_s_at is less than or equal to 10.512088 then 21 records fall in this pattern and all of 21 records have class value "medulloblastoma".

Chapter 5

Conclusion

In this paper, we have compared the performance of classification techniques in terms of accuracy, F1-score, and computation time on eleven different gene expression datasets. These results show that the logistic regression model has better results than the rest in maximum cases and is better suited for gene expression classification, despite not being used much for this purpose. We also performed pattern analysis using J48 decision tree on three datasets after applying feature selection. The patterns from the decision trees explain why a particular outcome (i.e. class value) has occurred which may help health professionals to understand the reasons for a particular situation. Note that, for pattern analysis purposes for the J48 decision tree, we used only 50 selected attributes, but only a few of them appeared in the tree. In the future, we would like to investigate why other attributes were not used for model building. In the future, we would like to further improve the results for classification by incorporating various feature selection methods, including PCA and tSNE. We also aim to understand the genes which dominate the classification results.

Bibliography

[1] J. C. Bezdek, S. K. Chuah, and D. Leep, "Generalized k-nearest neighbor rules," *Fuzzy Sets and Systems*, vol. 18, no. 3, pp. 237–256, 1986.

[2] M. W. Craven and J. W. Shavlik, "Machine learning approaches to gene recognition," *IEEE Expert*, vol. 9, no. 2, pp. 2–10, 1994.

[3] F. S. Collins and V. A. McKusick, "Implications of the human genome project for medical science," *Jama*, vol. 285, no. 5, pp. 540–544, 2001.

[4] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Machine learning*, vol. 52, no. 1-2, pp. 91–118, 2003.

[5] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[6] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[7] Y. Hoshida, J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Subclass mapping: Identifying common subtypes in independent disease data sets," *PloS one*, vol. 2, no. 11, e1195, 2007.

[8] M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers," in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, IEEE, 2007, pp. 1541–1546.

[9] I. Kurt, M. Ture, and A. T. Kurum, "Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease," *Expert systems with applications*, vol. 34, no. 1, pp. 366–374, 2008.

[10] B. Li, C.-H. Zheng, D.-S. Huang, L. Zhang, and K. Han, "Gene expression data classification using locally linear discriminant embedding," *Computers in Biology and Medicine*, vol. 40, no. 10, pp. 802–810, 2010.

[11] Q. Cheng, H. Zhou, and J. Cheng, "Fast selecting optimal feature subset for multi-class classification with applications to high dimensional data," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 33, no. 6, pp. 1217–1233, 2011.

[12] G. George and V. C. Raj, "Review on feature selection techniques and the impact of svm for cancer classification using gene expression profile," *arXiv preprint arXiv:1109.1062*, 2011.

[13] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, “Decision tree analysis on j48 algorithm for data mining,” *Proceedings of international journal of advanced research in computer science and software engineering*, vol. 3, no. 6, 2013.

[14] A. Priyam, G. Abhijeeta, A. Rathee, and S. Srivastava, “Comparative analysis of decision tree classification algorithms,” *International Journal of current engineering and technology*, vol. 3, no. 2, pp. 334–337, 2013.

[15] J. S. Mattick, M. A. Dziadek, B. N. Terrill, W. Kaplan, A. D. Spigelman, F. G. Bowling, and M. E. Dinger, “The impact of genomics on the future of medicine and health,” *Medical Journal of Australia*, vol. 201, no. 1, pp. 17–20, 2014.

[16] M. A. Rahman, “Automatic selection of high quality initial seeds for generating high quality clusters without requiring any user inputs,” *PhD dissertation*, 2014.

[17] S. Garcıa, J. Luengo, and F. Herrera, *Data preprocessing in data mining.* Springer, 2015, vol. 72.

[18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] R. K. Singh and M. Sivabalakrishnan, “Feature selection of gene expression data for cancer classification: A review,” *Procedia Computer Science*, vol. 50, pp. 52–57, 2015.

[20] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, “Gene expression data classification using support vector machine and mutual information-based gene selection,” *procedia computer science*, vol. 47, pp. 13–21, 2015.

[21] R. Asaoka, H. Murata, A. Iwase, and M. Araie, “Detecting preperimetric glaucoma with standard automated perimetry using a deep learning classifier,” *Ophthalmology*, vol. 123, no. 9, pp. 1974–1980, 2016.

[22] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[23] W. Chen, X. Xie, J. Wang, B. Pradhan, H. Hong, D. T. Bui, Z. Duan, and J. Ma, “A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility,” *Catena*, vol. 151, pp. 147–160, 2017.

[24] H. Lu, L. Yang, K. Yan, Y. Xue, and Z. Gao, “A cost-sensitive rotation forest algorithm for gene expression data classification,” *Neurocomputing*, vol. 228, pp. 270–276, 2017.

[25] M. Kang and N. J. Jameson, “Machine learning: Fundamentals,” *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, pp. 85–109, 2018.

[26] S. Liu, C. Xu, Y. Zhang, J. Liu, B. Yu, X. Liu, and M. Dehmer, “Feature selection of gene expression data for cancer classification using double rbf-kernels,” *BMC bioinformatics*, vol. 19, no. 1, pp. 1–14, 2018.

[27] B. C. Feltes, E. B. Chandelier, B. I. Grisci, and M. Dorn, “Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research,” *Journal of Computational Biology*, vol. 26, no. 4, pp. 376–386, 2019.

[28] P. He, B. Fan, X. Xu, J. Ding, Y. Liang, Y. Lou, Z. Zhang, and X. Chang, “Group k-svd for the classification of gene expression data,” *Computers & Electrical Engineering*, vol. 76, pp. 143–153, 2019.

[29] Y. He, J. Zhou, Y. Lin, and T. Zhu, “A class imbalance-aware relief algorithm for the classification of tumors using microarray gene expression data,” *Computational biology and chemistry*, vol. 80, pp. 121–127, 2019.

[30] P. Garcıa-Dıaz, I. Sánchez-Berriel, J. A. Martınez-Rojas, and A. M. Diez-Pascual, “Unsupervised feature selection algorithm for multiclass cancer classification of gene expression rna-seq data,” *Genomics*, vol. 112, no. 2, pp. 1916–1925, 2020.

[31] M. A. Rahman, B. Honan, T. Glanville, P. Hough, and K. Walker, “Using data mining to predict emergency department length of stay greater than 4 hours: Derivation and single-site validation of a decision tree algorithm,” *Emergency Medicine Australasia*, vol. 32, no. 3, pp. 416–421, 2020.