Hybrid ANOVA and LASSO Methods for Feature Selection and Linear Support Vector, Multilayer Perceptron and Random Forest Classifiers Based on Spark Environment for Microarray Data Classification

Wafaa S Albaldawi1,2 and Rafah M Almuttairi1

1 Department of Software, College of Information Technology, University of Babylon, Hillah, Iraq
2 Department of Computer Science, College of Science, University of Kerbala, Karbala, Iraq

E-mail: stud.wafaa.shakir@uobabylon.edu.iq

Abstract. Microarray dataset frequently contains a countless number of insignificant and irrelevant genes that might lead to loss of valuable data. The classes with both high importance and high significance gene sets are commonly preferred for selecting the genes, which determines the sample classification into their particular classes. This property has obtained a lot of importance among the specialists and experts in microarray dataset classification. The trained classifier model is tested for cancer datasets and Huntington disease data (HD) which consists of Prostate cancer (Singh) dataset comprising 102 samples, 52 of which are tumors and 50 are normal with 12625 genes. The lung cancer (Gordon) dataset comprises 181 samples, 150 of which are normal and 31 are tumors with 12533 genes. The breast cancer (Chin) dataset comprises 118 samples, 43 of which are normal and 75 are tumors with 22215 genes. The breast cancer (Chowdary) dataset comprises 104 samples, 62 of which are normal and 42 are tumors with 22283 genes. Finally, the Huntington disease (Borovecki) dataset comprises 31 samples, 14 of which are normal and 17 are with Huntington's disease with 22283 genes. This paper uses Multilayer Perceptron Classifier (MLP), Random Forest (RF) and Linear Support Vector classifier (LSVC) classification algorithms with six different feature selection methods named as Principal Component Analysis (PCA), Extra Tree Classifier (ETC), Analysis of Variance (ANOVA), Least Absolute Shrinkage and Selection Operator (LASSO), Chi-Square and Random Forest Regressor (RFR). Further, the paper presents a comparative analysis on the obtained classification accuracy and time consumed among the models in Spark environment and in conventional system. Performance parameters such as accuracy and time consumed are applied in this comparative analysis to analyze the behavior of the classifiers in the two environments. The results indicate that the models in spark environment was extremely effective for processing large-dimension data, which cannot be processed with conventional implementation related to a some algorithms. After that, a proposed hybrid model containing embedded approach (LASSO) and the Filter (ANOVA) approach was used to select the optimized features form the high dimensional dataset. With the reduced dimension of features, classification is performed on the reduced data set to classify the samples into normal or abnormal and applied in spark in hadoop cluster (distributed manner). The proposed model achieved accuracy of 100% in case of Borovecki dataset when using all classifiers, 100% in case of Singh, Chowdary and Gordon datasets when classified
with RF and LSVC classifiers. Also, accuracy was 96% in case of Chin dataset when using RF classifier with optimal genes with respect to accuracy and time consumed.

**Keywords.** Big data; classification; gene selection; MapReduce; microarray. PySpark, Spark, machine learning, support vector classifier, feature selection, feature scaling, dimension reduction, principal component analysis, NumPy.

1. **Introduction**

One of the most important techniques for classifying, diagnosing, prognosis and treating cancer is the microarray based gene expression profiling. Huge data volume is generated due to the continuous change in the disease behavior. Also, the data obtained from the micro-array cover their veracities, while changes have been identified as the time change (i.e. the velocity). It is considered as a type related to high dimensional data that has extremely massive amounts of features instead of the number of samples. Thus, the analysis related to microarray high dimensional data-set in a short period was vital. Furthermore, the traditional techniques of data mining have no ability of processing huge data-sets with excellent performance at adequate time. For the purpose of dealing with huge data-sets, the Spark programming model was utilized [1]. The approaches of feature selection might be divided into 3 categories: embedded/hybrid, wrappers and filters. The wrappers approaches are performing excellently compared to filter approaches since the process of feature selection was enhanced for the classifier to be utilized. Yet, the wrapper approaches are costly to be utilized for large feature space due to the high computational costs and every one of the feature sets should be estimated with trained classifier which finally slows the process of feature selection. The filter approaches are faster and have reduced computational costs, yet has ineffective reliability in the classification in comparison to wrapper approaches and is better suited for high dimensional datasets. In addition, the hybrid/embedded techniques were lately designed to use the benefits of wrappers and filters techniques. Furthermore, a hybrid method applies performance evaluation function and independent test of feature subset [2]. The most simplistic and fastest selection approach was univariate filtering, as such methods rank the features on the basis of certain score. Thus, the user selects the best k features. Also, the F statistic (i.e. ANOVA), t-test generalization, was utilized as filter, as proposed in [3]. Excellent feature selection approach must have elevated learning accuracy, yet not much computational overhead (space complexity and time complexity) are available, even though there were solid reviews on the feature selection [4]. The majority focused on certain research fields in the feature selection. Thus, there is high importance in thoroughly surveying advances in feature selection as well as discussing a few future challenges. Ladha and Deepa in [5] provided the following benefits of feature selection: it reduces the feature space dimensionality, for limiting the storage requirements and increasing the speed of algorithm. It removes noisy, irrelevant or redundant data, while immediate impacts for the data analyses tasks speed the running time which is related to learning algorithms, enhancing the quality of data and increase the resulting model precision. Feature set reduction, for the purpose of saving resources in the next round which is related to data collection or throughout utilization. Performance enhancement, for gaining predictive accuracy. Furthermore, data understanding for gaining knowledge related to the procedure created the data or visualizing the data. Because of the memory utilization and high computational cost, the high-dimensional data are challenging for the algorithms of classification [6]. The remainder of the paper is organized as follows; section 2 provides a background of microarray gene classification domain and survey, section 3 presents the proposed work, section 4 describes the experimental setup, section 5 discusses the results and comparative analysis and Section 6 presents the conclusions.

2. **Related works**

This section provides a short overview of feature selection approach and classifiers which are utilized by a variety of researchers. The rate of accuracy that has been accomplished by various authors for the classification of genes were listed in table 1. Kumar et al proposed different approaches of feature selection based upon statistical tests through the use of the Spark model. In addition, the Artificial Neural Network (ANN) which has been based on Spark framework (sf-ANN) which has been
suggested, running on a scalable cluster with several nodes. The efficiency of the sf-ANN was tested using micro-array data-sets of a variety of dimensions. A detailed comparative analysis based on the time of the execution was presented on the sf-ANN classifier based on the Spark model and the traditional system (data were maintained on a stand-alone machine), respectively to examine its efficiency [7]. The authors proposed the execution of the parallel algorithm of Chi-Square gene selection on Spark, the chosen genes were assessed with the use of the support vector machines (SVMs) and parallel logistic regression for the Binary classifications on the Spark Machine Learning library (Spark MLlib) and comparison of the precision of the prediction and classification. Results have shown that the parallel selection of the Chi-Square, which is succeeded by the parallel logistic regression and SVMs provide more sufficient accuracy in comparison with the accuracy obtained with the full set of the data of gene expression [8]. SelectKBest (from sklearn) was utilized, allowing for selecting features on the basis of k top scores. F-regression (i.e. sklearn) was utilized so as to carry out the feature selection. It was based upon the linear tests (i.e. the regression test) that are univariate. It was conducted to find the impact of each regressor. It is in fact a function of scoring. Ultimately, the Random Forest Classifier has been utilized for the prediction of classification [9]. Tabl et al presented a hierarchical system of machine learning predicting a five-year survivability of patients that have undergone a specific therapy; classes have been constructed on a combination of 2 parts which are the information of survivability and the provided therapy. For the part of information of survivability, it is responsible for the definition of whether a patient had survived the five-year interval or departed. Whereas the part of therapy indicates the fact that therapy was taken throughout that interval, including radiotherapy, surgery or hormone therapy, forming a total of 6 classes. The model performs the classification of one class versus the remaining at every node, making the tree-based model produces 5 nodes. This model has been trained with the use of a group of standard classifiers on the basis of a comprehensive study data-set, including the clinical information and genomic profiles of 347 patients. A combination of the feature selection approach and an approach of prediction have been implemented on each node for the identification of genes which have the ability of predicting class at that node. The genes that have been identified for each class could be serving as possible bio-marker to the treatment of the class for more sufficient survivability. The results have shown that this model is responsible for the identification of classes with the measurements of high-performance. An exhaustive analysis based upon the relevant literature has shown that some possible bio-markers are highly associated with the survivability of breast cancer and cancer in general [10]. Alghunaim and Al-Baity addressed the issue of the prediction of breast cancer in the context of the big data. They have considered 2 data varieties, which are, the DNA methylation (DM) and GE. The aim of this research was to scale up ML approaches utilized for the classification through the separate and joint application of each dataset. Therefore, they have chosen the Apache Spark as a platform. They have chosen the SVMs, random forest and the decision trees to create 9 models which are helpful to predict breast cancers. They have carried out a detailed comparative study with the use of 3 scenarios with DM, GE and a combination of both, to determine which one of the 3 data types would produce the optimal results in terms of the error rate and accuracy. In addition to that, they have carried out an experimental comparison between Weka and Spark platforms to determine their behaviors in the case of dealing with the large data. The results of their experimentation have shown that scaled SVMs in Spark environment have exceeded the rest of the classifiers, due to the fact that it had achieved the maximum precision and minimum error rate with gene expression data-set [11]. The parallelized hybrid feature selection (HFS) approach has been suggested for serving the purpose. This approach included parallelized correlation feature sub-set selection which is followed by the rank-based approaches of the feature selection. The chosen sub-set of genes has been assessed with the use of the algorithms of the parallel classification. The values of accuracy that were obtained have been compared to the existing rank-weight feature selection, the parallelized recursive feature selection approaches and also with values that have been obtained through executing the parallelized HFS on the Distributed Weka Spark. The accuracy of the classification that has been obtained with the suggested parallelized HFS approach was 79% & 97%, respectively for the childhood leukemia and gastric cancer. The suggested parallelized approach of the HFS approach has resulted in approximately 4%-15% enhancement in the accuracy of the classification, in comparison to the previous approaches.
The results have shown that the suggested algorithm of the parallelized feature selection has been scalable for the growing medical data and prediction of the sub-types of cancer in less time with a higher degree of accuracy [12]. Abinash and Vasudevan used RFE along with LASSO approach as a hybrid method for the selection of features and later it was applied to SVMs for easy classifications. It has made best in comparison with existing approaches by their measures of performance, regulated on 6 publicly available data-sets of cancer. Just out, it provides an adequate awareness in selecting features [13]. Venkataramana et al focused on the classification of the sub-types of the cancer with the use of the levels of Micro-array Gene Expression (MGE). The MGE data nature is multidimensional with rather a small number of samples. There is a necessity of performing the reduction of dimensionality for the selection of relevant genes and removal of redundant genes. The approach of the Recursive Feature Selection (RFS) has been suggested due to the fact that it repeatedly performs the procedure of the gene selection to the point where the optimal gene sub-set was discovered. The optimal sub-set of genes that has been obtained and additionally utilized to classify the use of various models and evaluate the use of 10-fold cross. To scale big gene expression data, parallelized model of classification has been explored on Spark model. A comparison has been made between non-parallelized classification model on the Weka and parallelized framework of classification on the Spark. The results have shown that parallelized model of classification performs better compared to the non-parallelized model of classification according to the execution of time and accuracy. In addition to that, the efficiency of the RFS and the parallelized classifier has been compared as well with the previous methods. The suggested RFS and the parallelized classifier exceeded the preceding approaches [14]. Navas-Delgado et al developed a data-base for the collection of relevant clinical data for patients who have melanoma, including the storage of patient gene levels of expression obtained from Nano-String platform. The Immune Profiling Panel has been issued in that case. That data-base was utilized by analyzing various expression profiles of patients. This analysis was carried out with Python, and a parallel version of algorithms has been available with Apache Spark for providing the scalability as required [15]. Bolon-Canedo et al explored the procedure of distribution on the micro-array data and they evaluated to what level it is possible to get similar results as these obtained with the entire data-set. They have carried out experimentations with various approaches of aggregation, the feature rankers and also evaluated the impact of distribution of feature ranking procedure on successive efficiency of the classification [16].

Table 1. The different practitioners and researchers for classification with the use of the micro-array (Singh, Chowdary, Chin, Borovecki and Gordon) data sets. The table provides feature selection and classification methods adopted and their equivalent accuracy values.

| Authors | Feature Selection Methods | Classification Method | Classification Accuracy |
|----------------|--------------------------|-----------------------|------------------------|
| Kumar M, Singh S and Rath S 2015 [17] | t-test | Functional link neural network (FLNN) | 0.9722 (Leukemia dataset) |
| Kumar M, Rath N, Swain A and Rath S 2015[19] | Information theory + statistical approach | Statistical test, ANOVA based upon the Map Reduce | 93.31 |
| Das K, Ray J and Mishra D 2015[18] | Information theory + statistical approach | MapReduce based K-Nearest Neighbor (K-NN) | 88.99 |
| Shukla A, Vardhan M and Singh P 2018 [20] | CMIM and AGA called | SVM, Extreme Learning Machine (ELM), and K-NN. | 87.06 |
| Pashaei E and Aydin N 2017 [21] | Binary version of the Black Hole Algorithm (BBHA-RF) | | 85.17 |
Zhang G, Hou J, Wang J, Yan C and Luo J 2020[22] Information gain (IG) and an enhanced binary krill herd (MBKH) KNN, SVM, and NB KNN 96.12 SVM 95.98 NB 73.58 on Lung Cancer Dataset

Han F, Tang D, Sun Y, Cheng Z, Jiang J and Li Q 2019 [23] Hybrid gene selection approach, which is based upon the strategy of gene scoring and enhanced particle swarm optimization * *

Xiong Y, Ling Q, Han F and Liu Q 2019[24] LASSO and BPSO * *

Shukla A, Singh P and Yardhan M 2020 [27] minimum redundancy maximum relevance (mRMR) then, teaching learning-based algorithm (TLBO) and gravitational search algorithm (GSA) * 99.62% in DLBCL dataset

Alanni R, Hou J, Azzawi H and Xiang Y 2019 [26] Deep gene selection (DGS) * 1 gene average of chromosomes AC avg.: 83.18 2 genes average of chromosomes AC avg.: 85.77 3 genes average of chromosomes AC avg.: 87.43 above 98% in six datasets and

Shukla A, Singh P and Yardhan M 2020 [27] minimum redundancy maximum relevance (mRMR) then, teaching learning-based algorithm (TLBO) and gravitational search algorithm (GSA) * 99.62% in DLBCL dataset

Alanni R, Hou J, Azzawi H and Xiang Y 2019 [28] An innovative Gene Selection Programming (GSP) is based on Gene Expression Programming (GEP) SVM a linear kernel serves GSP achieved 100% LOOCV accuracy with < 5 chosen genes on Lung Cancer, Leukemia 1, Leukemia 2, SRBCT, as well as the DLBCL data-sets. Over 98% --- other datasets

Dashtban M and Balafar M 2017 [29] novel evolutionary algorithm, which is referred to as the intelligent dynamic genetic algorithm (IDGA), based upon concepts of GA and ANN Integer-coded genetic algorithm with the variable-length genotype, adaptive parameters and modified genetic operators *

Sserwadda A and Sarac O 2017 [30] LASSO RF , SVM LASSOMLP, ReliefMLP, LASSOSVM, ReliefSVM, LASSORF, ReliefRF 100%, 85.6%, 100%, 85.6%, 90.7%, 89.0% (Chin)

Kumar M, Rath N and Rath S 2016 [32] (ANOVA, Friedman, and Kruskal–Wallis) based on Map-Reduce MapReduce-based K-nearest neighbour (mRKN) 77.41%, K= 21 82.24%, K=17 79.83%, K=21

Ray R, Kumar M, Tirkey A and Rath S 2016 [33] Mutual Information based on spark model (sf-MIFS) Logistic Regression (sf-LoR) and Naïve Bayesian (sf-NB) based on Spark model Train Accuracy, Test Accuracy Naive Bayes 80.83%, 81.57% LoR 63.86%, 67.28% GSE1315 Dataset

Kumar M 2017 [34] Kruskal-Wallis, ANOVA, and Friedman tests have been implemented with the use of the Spark and MapReduce Proximal SVM 100%, Leukemia 72%, Breast cancer 100%, Ovarian 81.7%, GSE24080 84.14%, GSE15061 81.11%, GSE13159
Based on the advanced strategies introduced in the literature review, most studies have announced comparable outcomes in terms of accuracy and reliability. Nonetheless, not all studies are based on the similar datasets. Also, not all studies are implemented in the same environment. Subsequently, they cannot be benchmarked consistently and thought about on a common ground. The presented work used minimum of single technique from each of the 2 categories mentioned earlier (embedded and filter). Each technique enabled choosing certain number of features. There are no requirements inducing the relation between feature sets from the same algorithm. With regard to all algorithms of feature selection, Scikit-learn Python package was utilized. Also, the current researchers decided to choose apache spark framework to perform classification of microarray data because it can be considered as large data because the numbers of features are larger than microarray samples to reduce consumed time and enhance the performance.

3. Proposed work

The existence of a massive amount of the irrelevant and insignificant features results in degrading the analysis quality of diseases such as ‘cancer’. A gene selection method using feature importance measures on microarray gene cancer data and HD is presented. Six feature selection approaches namely PCA, ETC, ANOVA, LASSO), Chi Square and RFR are applied to select the genes and afterward the high important genes are considered to contribute in disease diagnosis. Five well-known binary microarray data are considered for assessment and furthermore the applied classifiers are Multilayer Perceptron Classifier (MLP), Random Forest (RF) and Linear Support Vector classifier (LSVC) in Spark environment as well as in convention system with regard to binary classifications of the microarray samples through evaluating the time consumed on microarray samples and the classification accuracies. In addition, to enhance quality, it is highly important to analyze the dataset in a suitable point of view. This section includes the suggested method for classifying the microarray data, which consists of three stages:

**Stage 1: Pre-Processing**
Rescale each feature to certain range (commonly [0, 1]).

**Stage 2: Feature Selection**
The feature selection methodology was applied by utilizing hybrid method as t-statistic and LASSO which has high value of relevance; therefore, the dimensionality issue curse was decreased.

**Stage 3: Classification**
Random Division of obtained Data-set: The dataset was divided into 2 classes: training and test sets.

Table 2. Datasets List.

| Dataset  | Sample Size | Number of Features | Number of Classes | Disease          | Year   |
|----------|-------------|--------------------|-------------------|------------------|--------|
| Singh    | 102         | 12625              | 2                 | Prostate Cancer  | 2002   |
| Borovecki| 31          | 22283              | 2                 | Huntington's Disease | 2005   |
| Gordon   | 181         | 12533              | 2                 | Lung Cancer      | 2002   |
| Chin     | 118         | 22215              | 2                 | Breast Cancer    | 2006   |
| Chowdary | 104         | 22283              | 2                 | Breast Cancer    | 2006   |
Figure 1. The proposed system for binary microarray classification.
4. Experimental setup
The suggested algorithms were carried out on Spark cluster that consists of 1 master node and 5 slave nodes. Six commodity computers that are connected to a WI-FI have been utilized in this experimentation, and the configuration has been set up as follows:

4.1. Hardware configuration
- The Master node: Name Node1, CPU intel core i7, 3.4GHz × 4, RAM 16GB, Hard disk 1 TB
- The Slave node1: Data Node1, CPU intel core i5, 3.2GHz × 4, RAM 8GB, Hard disk 500GB
- The Slave node2: Data Node1, CPU intel core i5, 3.2GHz × 4, RAM 8GB, Hard disk 500GB
- The Slave node3: Data Node1, CPU intel core i5, 3.2GHz × 4, RAM 8GB, Hard disk 500GB
- The Slave node4: Data Node1, CPU intel core i5, 3.2GHz × 4, RAM 8GB, Hard disk 500GB
- The Slave node5: Data Node1, CPU intel core i5, 3.2GHz × 4, RAM 8GB, Hard disk 500GB

4.2. Software requirements
- Ubuntu 16.04 master node
- Ubuntu 19.10 on all slave nodes
- Java version "1.8.0_181"
- Hadoop 2.7.3
- Python 3.7.6
- Spark 2.4.3

There are two libraries of machine learning (ML) in the Spark framework. Spark Mllib and Spark ML with similar algorithms and different APIs. The two ML libraries receive several performance considerations of RDD and DataFrame and DataSet APIs, respectively. The new active development happens in Spark ML machine learning library. ML Spark is scikit-learn inspired. The reason for choosing ML Spark instead of Mllib Spark is that:

- ML Spark works on different data type (DataSet) from the data type (RDDs) that Mllib Spark works on. However, both of them deal with the RDDs and Data-sets of vectors which are easily represented and converted between RDD format and Data-set format.
- Spaks’ MLLib focuses on the provision of a core group of the algorithms for the programmers to utilize, whereas largely leaving data pipeline, preparation, cleaning, and feature selection tasks up to users. On the other hand, the Spark ML focuses on the exposure of a scikit-learn inspired pipeline API for everything from the preparation of data to the training of the model. In addition, MLLib RDD-based API is presently in the maintenance mode in Spark2.0+. Therefore, nowadays DataFrame is the primary Machine Learning API in spark.ml package. With fixing the problems, Mllib Spark will continue supporting the RDD-based API in spark.mllib package without adding new features to RDD-based APIs. Mllib Spark will add features to DataFrame-based API for reaching the feature parity with RDD-based API in the releases of the Spark2.0+.

The advantages of the DataFrames are the Spark Data sources, queries of the SQL and DataFrame, Catalyst and Tungsten optimizations, and uniform APIs across the languages. This is why, DataFrames provide a more user-friendly API compared to the RDDs. DataFrame-based APIs for the MLLib provide a consistent API over the algorithms of machine learning in multiple languages. They facilitate functional Pipelines of machine learning, especially the feature transformations. However, DataFrame-based API and MLLib are not deprecated as a whole.

4.3. Security
Spark protection is off by default. It supports multiple types of deployments. Security issues are faced in a wide range of the forms. Spark does not always protect from everything [35]. Currently Spark supports the channel authentication of remote procedure call (RPC) with the use of a shared secret. Authentication can be switched on through setting spark.authenticate. The exact deployment-specific
method is used to produce and distribute the shared secret [35]. Spark can automatically manage producing and transmitting the mutual secret for Spark on the YARN and on the local deployments. Each application utilizes a shared secret which is distinctive to it. In YARN’s cases, this function is based on permitting the encryption of YARN RPC to protect the distribution of the secrets. For other resource managers, it is important to configure the spark.authenticate.secret on every node. All daemons and applications have to share this secret [35].

5. Results
In this section, the results of the suggested method are discussed. Five case studies Chin, Chowdary, Borovecki, Gordon and Singh microarray datasets were considered to find the accuracy of the classification. N fold cross-validation (CV) has been implemented for assessing the classifier’s efficiency, due to the fact that it presents more realistic evaluation of the classifiers, considerably generalizing to the unseen data.

Case study: Chin dataset
The chn dataset includes 22215 features (i.e. genes), and it was classified to two classes. These two in combination have 118 samples as shown in Tables 3 to 6. The dataset includes 43 normal and 75 abnormal samples.

Table 3. Performance analysis of Multilayer Perceptron (a) accuracy (b) time consumed. Using Chin Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy | Chin | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|----------|------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 0.954545454545455 | 0.909090909090909 | 0.772727272727273 | 0.909090909090909 | 0.727272727272727 | 0.909090909090909 | 0.772727272727273 |
| Multilayer Perceptron (without Spark) | 0.745672711864407 | 0.779661016949153 | 0.73728813559322 | 0.73728813559322 | 0.73728813559322 | 0.73728813559322 | 0.73728813559322 |
| Time | | | | | | | |
| Chin | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
| Multilayer Perceptron (Spark) | 7.9682263168335 | 7.8222287012024 | 8.936895942688 | 71.6801040172577 | 4.28378295898438 | 13.740439414978 |
| Multilayer Perceptron (without Spark) | 7661.42957782745 | 5889.61997127533 | 456.782869815826 | 431.16827750206 | 2561.99800419807 |

Table 4. Performance analysis of LinearSVC (a) accuracy (b) time consumed. Classifiers using Chin Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy | Chin | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|----------|------|-------|-------|------------|-----|-----|-----|
| Linear Support Vector Machine (Spark) | 0.904761904761905 | 0.754237288135593 | 0.745672711864407 | 0.73728813559322 | 0.73728813559322 | 0.73728813559322 | 0.73728813559322 |
| Linear Support Vector Machine (without Spark) | 2.56412744522095 | 3.36847734451294 | 3.22765445709229 | 5.39573097229004 | 3.5589394569397 | 3.27751064300537 |

9
Table 5. Performance analysis of Random Forest (a) accuracy (b) time consumed. Classifiers using Chin Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Classifier (Spark) | Accuracy | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|--------------------|----------|-------|-------|------------|-----|-----|-----|
| Random Forest      | 0.8077   | 0.8   | 0.8947| 0.9444     | 0.8421|     |     |
| Time Chin          | 0.7881   | 0.7796| 0.8050| 0.7881     | 0.7881|     |     |

Case study: Gordon Dataset
The Gordon dataset includes 12533 features (i.e. genes), which have been classified into normal and cancer classes, with 181 samples as presented in Tables 7 to 10. Out of the 181 samples, the data-set includes 150 normal and 31 cancer samples.

Table 6. Performance analysis of Multilayer Perceptron, LinearSVC, Random Forest classifiers using Chin Dataset with proposed hybrid feature selection in Apache Spark Framework (Hadoop cluster) by the accuracy and time.

| Classifier             | Accuracy | Time            | numFold |
|------------------------|----------|-----------------|---------|
| LASSO+ANOVA Chin Dataset |          |                 |         |
| Multilayer Perceptron  | 0.8929   | 4.6372610474    | 2       |
| Linear Support Vector Machine | 0.8421  | 9.8988270756    | 12      |
| Random Forest          | 0.9615   | 4.8354220390    | 15      |

Table 7. Performance analysis of Multilayer Perceptron (a) accuracy (b) time consumed. Using Gordon Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Classifier (Spark) | Accuracy | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|--------------------|----------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron | 1       | 0.9677 | 1     | 1          | 1   | 1   | 1   |
| Time Gordon        | 0.988950276243094 | 1587.0660532951 | 3932.62175154686 |
Table 8. Performance analysis of LinearSVC (a) accuracy (b) time consumed. Classifiers using Gordon Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Gordon | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-----------------|-------|-------|------------|-----|-----|-----|
| Linear Support Vector Machine (Spark) | 1 | 1 | 0.972222222222222 | 0.84375 | 0.972972972972973 | 1 |
| Linear Support Vector Machine (without Spark) | 0.988950276243094 | 0.988950276243094 | 0.988950276243094 | 0.988950276243094 | 0.988950276243094 |
| Time Gordon | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
| Linear Support Vector Machine (Spark) | 2.69476771354675 | 3.87859678268433 | 3.22416877746582 | 9.05194759368897 | 3.3305448532104 | 2.91451478004456 |
| Linear Support Vector Machine (without Spark) | 4.00.53237080574 | 6.02916502952576 | 5.29101753234863 | 2698.8727517128 | 30.5987060070038 | 60.9946784973145 |

Table 9. Performance analysis of Random Forest (a) accuracy (b) time consumed. Classifiers using Gordon Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Gordon | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-----------------|-------|-------|------------|-----|-----|-----|
| Random Forest (Spark) | 1 | 0.973 | 0.9697 | 1 | 1 | 0.9706 |
| Random Forest (without Spark) | 0.961325966850829 | 0.883977900552486 | 0.87845036874033 | 0.961325966850829 | 0.939226519337017 | 0.939226519337017 |
| Time Gordon | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
| Random Forest (Spark) | 4.1542809009552 | 3.379840367399292 | 3.9933621833923 | 4.7840695504761 | 4.45235061645508 | 3.70792746543884 |
| Random Forest (without Spark) | 63.440184818268 | 26.6270852088928 | 26.8976616859436 | 121.06323579254 | 36.9190378189087 | 33.2902998924255 |

Table 10. Performance analysis of Multilayer Perceptron, LinearSVC, Random Forest classifiers using Gordon Dataset with proposed hybrid feature selection in Apache Spark Framework (Hadoop cluster) by measuring the accuracy and time.

| LASSO+ANOVA Gordon Dataset | Accuracy | Time | numFold |
|----------------------------|----------|------|--------|
| Multilayer Perceptron | 0.9778 | 9.61810612679 | 12 |
| Linear Support Vector Machine | 1.0 | 9.95545601845 | 12 |
| Random Forest | 1.0 | 4.79531884193 | 4 |

Case study: Borovecki Data-set
The Borovecki data-set includes 22283 features (i.e. genes), which have been classified as person with Huntington's Disease and without classes, which have 31 samples as shown in Tables 11 to 14. Out of 31 samples, the data-set includes 14 without and 17 with Huntington's Disease samples.
Table 11. Performance analysis of Multilayer Perceptron (a) accuracy (b) time consumed. Using Borovecki Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy          | Borovecki (Spark) | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-------------------|--------------------|-------|-------|------------|-----|-----|-----|
| Random Forest     | 0.903225806451613  | 0.870967741935484 | 0.870967741935484 | 0.870967741935484 | 0.903225806451613 | 0.838709677419355 | 0.903225806451613 |
| Random Forest     |                    |       |       |            |     |     |     |
| (without Spark)   |                    |       |       |            |     |     |     |

Table 12. Performance analysis of LinearSVC (a) accuracy (b) time consumed. Classifiers using Borovecki Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy          | Borovecki (Spark) | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-------------------|--------------------|-------|-------|------------|-----|-----|-----|
| Linear Support    | 0.903225806451613  | 0.835483870967742 | 0.835483870967742 | 0.903225806451613 | 0.903225806451613 | 0.9316453870967742 |
| Vector Machine    |                    |       |       |            |     |     |     |
| (Spark)           |                    |       |       |            |     |     |     |
| Linear Support    |                    |       |       |            |     |     |     |
| Vector Machine    |                    |       |       |            |     |     |     |
| (without Spark)   |                    |       |       |            |     |     |     |

Table 13. Performance analysis of Random Forest (a) accuracy (b) time consumed. Classifiers using Borovecki Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy          | Borovecki (Spark) | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-------------------|--------------------|-------|-------|------------|-----|-----|-----|
| Random Forest     | 0.870967741935484  | 0.838709677419355 | 0.838709677419355 | 0.870967741935484 | 0.870967741935484 | 0.870967741935484 | 0.870967741935484 |
| Random Forest     |                    |       |       |            |     |     |     |
| (without Spark)   |                    |       |       |            |     |     |     |
| Random Forest     |                    |       |       |            |     |     |     |
| (Spark)           | 3.17668461799622    | 3.09092235565186 | 3.545868982727051 | 3.6043976823420105 | 4.03150844573975 | 3.13382077217102 |
| Random Forest     |                    |       |       |            |     |     |     |
| (without Spark)   |                    |       |       |            |     |     |     |

12
Table 14. Performance analysis of Multilayer Perceptron, LinearSVC, Random Forest classifiers using Borovecki Dataset with proposed hybrid feature selection in Apache Spark Framework (Hadoop cluster) by measuring the accuracy and time.

| LASSO+ANOVA Borovecki Dataset | Accuracy | Time       | numFold |
|-------------------------------|----------|------------|---------|
| Multilayer Perceptron         | 1.0      | 5.60887598991 | 2       |
| Linear Support Vector Machine | 1.0      | 6.73601007462 | 12      |
| Random Forest                 | 1.0      | 4.19274616241 | 4       |

Case study: Chowdary Dataset
The Chowdary dataset includes 22283 features (genes), which have been classified into normal and cancer classes, which have 104 samples as shown in Tables 15 to 18. Out of 104 samples, the data-set includes 62 normal and 42 cancer samples.

Table 15. Performance analysis of Multilayer Perceptron (a) accuracy (b) time consumed. Using Chowdary Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|------------------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 1.0 | 0.9 | 0.952380952380952 | 1 | 0.904761904761905 |
| Multilayer Perceptron (without Spark) | 0.951923076923077 | 0.971153846153846 | 0.961538461538462 | 0.961538461538462 | 0.942307692307692 | 0.961538461538462 |

| Time Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|---------------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 4.41435670852661 | 8.17480039596558 | 9.78351378440857 | 11.2930853366852 | 3.08392810821533 | 4.93559505555298 |
| Multilayer Perceptron (without Spark) | 5983.21925926209 | 2678.38493156433 | 5962.6507525444 | 47162.6010899544 | 318.18338561058 | 1203.59094333649 |

Table 16. Performance analysis of LinearSVC (a) accuracy (b) time consumed. Classifiers using Chowdary Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|------------------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 1.0 | 1.0 | 0.875 | 1 | 1 | 0.9 |
| Multilayer Perceptron (without Spark) | 0.971153846153846 | 0.923076923076923 | 0.923076923076923 | 0.971153846153846 | 0.942307692307692 | 0.942307692307692 |

| Time Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|---------------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 2.5987630429993 | 2.6297565721405 | 3.2249388947632 | 3.4455089569018 | 3.35117411613464 | 3.1840808391571 |
| Multilayer Perceptron (without Spark) | 52.092871427536 | 1.06287741661072 | 1.04184770584106 | 661.07497528989 | 2.85964727401733 | 5.7695791721344 |
Table 17. Performance analysis of Random Forest (a) accuracy (b) time consumed. Classifiers using Chowdary Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|-------------------|-------|-------|------------|-----|-----|-----|
| Random Forest (Spark) | 0.9412 | 0.9062 | 1 | 1 | 1 | 1 |
| Random Forest (without Spark) | 0.884615384615385 | 0.826923076923077 | 0.826923076923077 | 0.875 | 0.865384615384615 | 0.865384615384615 |
| Time Chowdary | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
| Random Forest (Spark) | 3.72608494758606 | 3.3051381111145 | 4.43893241882324 | 4.74723505973816 | 4.50769543647766 | 3.36265897750854 |
| Random Forest (without Spark) | 21.4761786460876 | 13.8484935760498 | 13.6304321289063 | 36.8311598300934 | 15.4488880634308 | 14.1782126426697 |

Table 18. Performance analysis of Multilayer Perceptron, LinearSVC, Random Forest classifiers using Chowdary Dataset with proposed hybrid feature selection in Apache Spark Framework (Hadoop cluster) by measuring the accuracy and time.

| LASSO+ANOVA | Chowdary Dataset | Accuracy | Time | numFold |
|-------------|-----------------|----------|------|---------|
| Multilayer Perceptron | 0.9545 | 18.609910965 | 5 |
| Linear Support Vector Machine | 1 | 6.38929700851 | 12 |
| Random Forest | 1 | 4.80585408211 | 10 |

Case study: Singh Dataset
The Singh data-set includes 12625 features (i.e. genes), which have been classified into normal and cancer classes, which have 102 samples as presented in Tables 19 to 22. Out of 102 samples, the data-set includes 50 normal samples and 52 cancer ones.

Table 19. Performance analysis of Multilayer Perceptron (a) accuracy (b) time consumed. Using Singh Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

| Accuracy Singh | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
|----------------|-------|-------|------------|-----|-----|-----|
| Multilayer Perceptron (Spark) | 1 | 0.8 | 0.8 | 0.95 | 0.6 | 0.9 |
| Multilayer Perceptron (without Spark) | 0.705882352941176 | 0.764705882352941 | 0.8 | killed | 0.647058823529412 | 0.696078431372549 |
| Time Singh | LASSO | ANOVA | Chi-Square | ETC | PCA | RFR |
| Multilayer Perceptron (Spark) | 5.0800130367291 | 22.315198893154 | 5.61544442176819 | 30.5479934215546 | 4.09667602759399 | 4.81678175926209 |
| Multilayer Perceptron (without Spark) | 3770.30138683319 | 2538.83329534531 | 5.5660274535706 | killed | 308.155163049698 | 1233.580950737 |
Table 20. Performance analysis of LinearSVC (a) for accuracy (b) for time consumed. Classifiers using Singh Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

|                        | Accuracy | Time        |
|------------------------|----------|-------------|
| LASSO                  | 0.894736 | 2.845682   |
| ANOVA                  | 0.96     | 2.682762   |
| Chi-Square             | 0.857143 | 3.194722   |
| ETC                    | 0.809524 | 5.751634   |
| PCA                    | 0.818182 | 3.498464   |
| RFR                    |          | 2.802773   |

Table 21. Performance analysis of Random Forest (a) accuracy (b) time consumed. Classifiers using Singh Dataset with LASSO, ANOVA, Chi-Square, ETC, PCA and RFR feature selections in Scikit Learn. The comparison is between the classification in the two environments (Scikit Learn and Apache Spark Framework).

|                        | Accuracy | Time        |
|------------------------|----------|-------------|
| LASSO                  | 0.9545   | 4.214425   |
| ANOVA                  | 0.8636   | 3.586763   |
| Chi-Square             | 0.8182   | 4.188703   |
| ETC                    | 0.9       | 5.559513   |
| PCA                    | 0.8125   | 4.612807   |
| RFR                    |          | 3.816942   |

Table 22. Performance analysis of Multilayer Perceptron, LinearSVC, Random Forest classifiers using Singh Dataset with proposed hybrid feature selection in Apache Spark Framework (Hadoop cluster) by the accuracy and time with different number of folds.

|                        | Accuracy | Time        | numFold |
|------------------------|----------|-------------|---------|
| Multilayer Perceptron  | 0.9048   | 8.154597   | 2       |
| Linear Support Vector Machine | 1 | 8.785873 | 12       |
| Random Forest          | 1        | 4.566454   | 12      |

Based on the analysis above, it has been noticed that there are variations in the rate of the classification. The interpretation which has been obtained from the results above can be summarized as: In case of Chin dataset as shown in table 6, hybrid selection method with Random Forest classifiers show more sufficient accuracy value than Multilayer Perceptron classifier and LSVC. While in case of Borovecki dataset as listed in table 14, hybrid selection method with all classifiers shows the best accuracy at 100%. In case of Gordon, Chowdary and Singh datasets (table 10, table 18 and table 22 respectively), hybrid selection method with LSV and RF classifiers show better accuracy than the MLP classifier. Table 10, table 14, table 18 and table 22 for four datasets except Chin dataset show that hybrid selection method with Random Forest has taken less time than the other two classifiers. In addition, in all cases presented in table 3, table 4, table 5, table 7, table 8, table 9, table 11, table 12,
table 13, table 15, table 16, table 17, table 19, table 20, table 21, all feature selection methods with all classification algorithms in apache spark have taken less time with higher accuracy than the same classifiers in the traditional system (in which the data were not distributed, maintained on a standalone machine and processed in the conventional way manner).

6. Conclusion and future work
In this paper, an experiment has been made to design classification models for classifying the samples of microarray data into their particular classes. Thus, a classification model was designed using MLP, LSVC and RF classifiers on selected gene subsets. Feature selection was carried out using six methods and the hybrid technique using embedded approach and Filter based method. K-fold CV technique was applied to enhance the performance of the classifiers. The performance of the classifiers for all five data sets were evaluated using performance parameters. From the computed result, it is observed that RF classifier with hybrid feature selection produce better outcomes when compared with MLP and LSVC. Further, this hybridization may help in reducing the complexity of the classification model. The results indicate that the models in spark environment was extremely effective for processing large-dimension data, which cannot be processed with conventional implementation related to some algorithms. In future work, a combination of other methods of feature selection can be used and other algorithms of data mining in a suitable environment. The goal is to increase the classification accuracy with less time consumed and reduce the number of genes for another analysis.

Nomenclature

| Acronym | Description |
|---------|-------------|
| HD      | Huntington disease |
| PCA     | Principal Component Analysis |
| ETC     | Extra Tree Classifier |
| ANOVA   | Analysis of Variance |
| LASSO   | Least Absolute Shrinkage and Selection Operator |
| RFR     | Random Forest Regressor |
| MLP     | Multilayer Perceptron |
| RF      | Random Forest |
| LSVC    | Linear Support Vector Classifier |
| ML      | Machine Learning |
| KNN     | K-Nearest Neighbors |
| NB      | Naive Bayes |
| SVM     | Support Vector Machine |
| DM      | DNA methylation |
| GE      | Gene Expression |
| HFS     | Hybrid Feature Selection |
| MGE     | Microarray Gene Expression |
| RFS     | Recursive Feature Selection |
| FLNN    | Functional link neural network |
| ELM     | Extreme Learning Machine |
| IG      | Information Gain |
| MBKH    | Binary Krill Herd |
| BPSO    | Binary Particle Swarm Optimization |
| GSP     | Gene Selection Programming |
| GSA     | Gravitational Search Algorithm |
| GEP     | Gene Expression Programming |
| IDGA    | Intelligent Dynamic Genetic Algorithm |
| RDD     | Resilient Distributed Dataset |
| Mlib    | Machine learning library |
| API     | Application Programming Interface |
| RPC     | Remote Procedure Call |
| YARN    | Yet Another Resource Negotiator |
| CV      | Cross-Validation |
| FS      | Feature Selection |
| SQL     | Structured Query Language |
| sf-NB   | Naïve bayesian based on spark model |
| numFold | number of folds cross validation |
| sf-MIFS | Mutual Information based on spark model |
| sf-LoR  | Logistic Regression based on spark model |
| sf-ANN  | sampling function Artificial Neural Network |
| BBHA-RF | Binary Version of the Black Hole Algorithm |
| CMIM-AGA| conditional mutual information maximization - adaptive genetic algorithm |
| DGS     | Deep gene selection |
| mRMR    | minimum redundancy maximum relevance |
| TLBO    | teaching learning-based algorithm |

1. References
[1] Jenifer X and Lawrance R 2017 Classification of Microarray Data using SVM Mapreduce (2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), doi:10.1109/itcosp.2017.8303083)
[2] Veerabhadrapa M and Rangarajan L 2010 Bi-Level Dimensionality Reduction Methods Using Feature Selection and Feature Extraction (Int. J. Comput. Appl.) vol 4 pp 33–8
[3] Haury A, Gestraud P and Vert J 2011 The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures (PLoS ONE 6 e28210,
do:10.1371/journal.pone.0028210)
[4] Sheikhpour R, Sarram M, Gharaghani S and Chahooki M 2017 A Survey on Semi-Supervised Feature Selection Methods (Pattern Recogn) vol 64 pp 141–58
[5] Ladha L and Deepa T 2011 Feature Selection Methods And Algorithms (Int. j. Eng.) vol 3 pp 1787–97
[6] Janecek A, Gansterer W, Demel M and Ecker G 2008 On the Relationship Between Feature Selection and Classification Accuracy (Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD 2008 PMLR) vol 4 pp 90–105
[7] Kumar M, Ray R and Rath S 2017 Spark Based Classification of Microarray Data using Scalable Artificial Neural Network (Int. J. Data Min. Bioin.) vol 19 pp 312–39
[8] Lokeswari Y and Jacob S 2017 Prediction of Child Tumours from Microarray Gene Expression Data Through Parallel Gene Selection and Classification on Spark (Adv. Intell. Syst.) pp 651–61
[9] Bagga S, Goyal A, Gupta N and Goyal A 2020 Credit Card Fraud Detection using Pipeling and Ensemble Learning (Procedia Comput. Sci.) vol 173 pp 104–112
[10] Tabl A, Alkhateeb A, ElMaraghy W, Rueda L and Ngom A 2019 A Machine Learning Approach for Identifying Gene Biomarkers Guiding the Treatment of Breast Cancer (Front. Genet, doi: 10.3389/fgene.2019.00256) vol 10
[11] Alghunaim S and Al-Baity H 2019 On the Scalability of Machine-Learning Algorithms for Breast Cancer Prediction in Big Data Context (IEEE Access) vol 7 pp 91535–46
[12] Venkataramana L, Jacob S, Ramadoss R, Saisuma D, Haritha D and Manoja K 2019 Improving Classification Accuracy of Cancer Types using Parallel Hybrid Feature Selection on Microarray Gene Expression Data (Genet. Genom, doi:10.1007/s13258-019-00859-x)
[13] Abinash M and Vasudevan V 2019 Gene Selection using a Hybrid RFE Along with LASSO for Cancer Classification (Int. j. Eng.) vol 9 pp 83–96
[14] Venkataramana L, Jacob S and Ramadoss R 2018 Parallelized Classification of Cancer Sub-Types From Gene Expression Profiles using Recursive Gene Selection (Stud. Inform. Control) vol 27 pp 215–24
[15] Navas-Delgado I, Garcia-Nieto J, López-Camacho E, Rybinski M, Lavado R, Berciano Guerrero M and Aldana-Montes J 2019 VIGLA-M: Visual Gene Expression Data Analytics (BM Bioinformatics) vol 20
[16] Bolon-Canedo V, Sechidis K, Sanchez-Marono N, Alonso-Betanzos A and Brown G 2017 Exploring The Consequences of Distributed Feature Selection in DNA Microarray Data (Int Jt Conf Neural Netw)
[17] Kumar M, Singh S and Rath S 2015 Classification of Microarray Data using Functional Link Neural Network (Procedia Comput. Sci.) vol 57 pp 727–37
[18] Das K, Ray J and Mishra D 2015 Gene Selection using Information Theory and Statistical Approach (Indian J Sci Technol) vol 8 p 695
[19] Kumar M, Rath N, Swain A and Rath S 2015 Feature Selection and Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor (Procedia Comput. Sci.) vol 54 pp 301–10
[20] Shukla A, Vardhan M and Singh P 2018 A Two-Stage Gene Selection Method for Biomarker Discovery From Microarray Data For Cancer Classification (Chemometr. Intell. Lab, doi:10.1016/j.chemolab.2018.10.009)
[21] Pashaei E and Aydin N 2017 Binary Black Hole Algorithm for Feature Selection and Classification On Biological Data (Appl. Soft Comput.) vol 56 pp 94–106
[22] Zhang G, Hou J, Wang J, Yan C and Luo J 2020 Feature Selection for Microarray Data Classification Using Hybrid Information Gain and a Modified Binary Krill Herd Algorithm (Interdiscip. Sci.) vol 12 pp 288–301
[23] Han F, Tang D, Sun Y, Cheng Z, Jiang J and Li Q 2019 A Hybrid Gene Selection Method Based On Gene Scoring Strategy And Improved Particle Swarm Optimization (BM Bioinformatics) vol 20, doi: 10.1186/s12859-019-2773-x)
[24] Xiong Y, Ling Q, Han F and Liu Q 2019 *An Efficient Gene Selection Method for Microarray Data Based on LASSO And BPSO* (BMC Bioinformatics) vol 20, doi:10.1186/s12859-019-3228-0

[25] Zhongxin W, Gang S, Jing Z and Jia Z 2016 *Feature Selection Algorithm Based on Mutual Information and Lasso for Microarray Data* (Open Biotechnol J) vol 10 pp 278–86

[26] Alanni R, Hou J, Azzawi H and Xiang Y 2019 *Deep Gene Selection Method to Select Genes from Microarray Datasets for Cancer Classification* (BMC Bioinformatics 20, doi:10.1186/s12859-019-3228-0)

[27] Shukla A, Singh P and Vardhan M 2020 *Gene Selection for Cancer Types Classification using Novel Hybrid Metaheuristics Approach Swarm* (Evol. Comput.) vol 54 p 100661

[28] Alanni R, Hou J, Azzawi H and Xiang Y 2019 *A Novel Gene Selection Algorithm for Cancer Classification using Microarray Datasets* (BMC Med. Genomics) vol 12, doi:10.1186/s12920-018-0447-6

[29] Dashtban M and Balafar M 2017 *Gene Selection for Microarray Cancer Classification using a New Evolutionary Method Employing Artificial Intelligence Concepts* (Genomics) vol 109 pp 91–107

[30] Sserwadda A and Sarac O 2017 *Gene Selection and Classification of Pancreatic Microarray Datasets* (2017 25th Sig. Process Commun., doi:10.1109/siu.2017.7960251)

[31] Güçkıran K, Cantürk I and Ozyılmaz L 2019 *DNA Microarray Gene Expression Data Classification Using SVM, MLP, and RF with Feature Selection Methods Relief and LASSO* (Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi) vol 23 pp 126–132

[32] Kumar M, Rath N and Rath S 2016 *Analysis of Microarray Leukemia Data using an Efficient Mapreduce-Based K-Nearest-Neighbor Classifier* (J. Biomed. Inform.) vol 60 pp 395–409

[33] Ray R, Kumar M, Tirkey A and Rath S 2016 *Scalable Information Gain Variant on Spark Cluster for Rapid Quantification of Microarray* (Procedia Comput. Sci.) vol 93 pp 292–298

[34] Kumar M 2017 *Analysis of Microarray Data using Machine Learning Techniques on Scalable Platforms PhD thesis*

[35] Security-Spark 2.4.3 Documentation 2020 Retrieved 5 November 2020 (https://spark.apache.org/docs/2.4.3/security.html#standalone-mode-only)