Spark based Framework for Supervised Classification of Hyperspectral Images

N. Aswini[1]
Division of Computer and Information Sciences
Annamalai University, Annamalainagar, India

R. Ragupathy[2]
Department of Computer Science and Engineering
Annamalai University, Annamalainagar, India

***Abstract*—The advancement of remote sensing sensors acquired large amount of image data easily. Primary aspects of big data, such as volume, velocity, and variety, are represented in the acquired images. Furthermore, standard data processing approaches have different limits when dealing with such large amounts of data. As a result, good machine learning-based algorithms are required to process the data with higher accuracy and lower computational efficiency. Therefore, we propose ANOVA F-test based spectral feature selection method with a distributed implementation of this machine learning algorithm on Spark. Experimental results are obtained using the bench mark datasets acquired using AVIRIS and ROSIS sensors. The performance of Spark MLlib based supervised machine learning techniques are evaluated using the criteria viz., accuracy, specificity, sensitivity, F1-score and execution time. Added to that, we compared the execution time between distributed processing and processing with single processor. The results reveal that the proposed strategy significantly cuts down on analytical time while maintaining classification accuracy.**

***Keywords*—Hyperspectral images; spark; supervised classifiers; spectral features; ANOVA F-test; distributed processing**

I. Introduction

Advances in recent years, optical sensor technology have provided a wealth of data in terms of achieving necessary spectrum, temporal, and spatial resolutions. Hyperspectral imagery make up a significant portion of the spectral details (HSIs) [1].

The currently available high spectral resolution helps us to obtain small materials and mild objects with confined spectral bands for different applications such as identification, town planning, agriculture, surveillance, and quantification [2]. Though, remote sensing often relies on hyperspectral imagery (acquired from various satellites or from airborne sensors) which allows capturing simultaneously the radiance at several wavelength bands. These wavelength bands are contiguous and their range is predominantly high. Certainly, these data act as a major performer in big remote sensing data and these has at least these traditional 4V’s. The volume, the velocity, the veracity and the variety [3].

Let's begin with the letter V, which stands for Volume. The amount of data collected remotely is increasing in terms of hours and minutes. In recent the years, there has been a phenomenal increase in the data consumption that is heading from terabytes to exabytes. Velocity is the second V. It refers to the process of creating, analysing, and interpreting a large amount of remote sensing data in a short amount of time. The third V stands for Value. Multisource, multitemporal, multispectral, or hyperspectral data can be acquired remotely. The term "multisource" refers to the fact that images can be acquired from a variety of sources, including RADAR, LiDAR, optical, and so on. Images having varying resolutions are referred to as multiresolution (spatial or spectral) So, it is difficult to processing remote sensed data not only because of its large volume of data but also it pre-processing, storage and analysis. Various recent literatures, have proposed many frameworks to deal with these problems. Among these frameworks, one of the popular framework is Spark. MapReduce is a feature of Apache Spark, an open-source parallel computing platform. It gives you the flexibility, scalability, and performance you need to meet the problems of big data. Spark is a library that combines two important libraries. SQL is used to query structured data, while MLlib is used to learn about various learning algorithms and statistical methodologies [4]. Of course, MLlib is, Spark's open-source Machine Learning (ML) library, which contains a number of useful training features. It also supports a variety of languages, including Python, R, Java, and Scala, as well as a high-level API that enhances Spark's ecosystem and simplifies the building of machine learning pipelines. [5].

Supervised classification using ML is the important method to extract related information from hyperspectral images. In general, a supervised classifier learns from a training phase that contains hyperspectral data and its corresponding class labels, then generalises to identify class labels for hyperspectral data outside of the training set.

In current research, the intended architecture is composed of three stages viz., Feature extraction, Feature selection using ANOVA F-test and supervised classifier. For better classification accuracy, it is necessary to work with good number of features. So that, we include Feature extraction and Feature Selection (FS). FS is an important strategy for selecting a subset of features from a large number of characteristics and then reducing the high data dimensionality, which results in the greatest classification accuracy. The success of feature selector algorithms is generally measured by comparing the classification techniques with and without selection of features. Feature selection is predominantly used to decrease the dimension of the original feature by eliminating the redundant and irrelevant features, and also to increase the performance and effectiveness of classification. The best features identified are used to satisfy some specified criteria. When compared to using the entire feature subset, classification with feature selection lowers the learning cost by lowering the number of features used for learning, as well as it removes the unnecessary, noisy, and redundant data, and also ensures the best learning accuracy. In this study, we used one-way ANOVA F-test statistics to measure resemblances for relevant features and to reduce the data dimensions of feature space by finding the necessary features, with the objective of reduced computational complexity or enhancing classification accuracy, maybe both. A comparative study has been carried out between RDD based supervised techniques like Decision tree (DT) [6], Random Forest (RF) [7] and Logistic regression (LR) [8] using the combination of FS with the supervised classifiers on regular mode. The overall performance of hyperspectral imaging classification has been considerably enhanced as a result of the powerful feature representation learned using various ML classification approaches, according to several studies [9].

The purpose of this work is to implement the hyperspectral imagery with feature selection method in distributed environment. In order to test the improvement of the suggested technique, we choose to utilise four distinct widely accessible datasets based on image acquisition and image resolution. The rest of the research paper is arranged as follows: Section II discusses the feature selection based on ANOVA F-test method; Section III addresses experimental designs; Section IV reports the configuration; Section V describes the dataset related to works; Section VI reports the performance metrics; Section VII describes the results and analysis of the experiment and Section VIII draws the conclusion in brief.

II. Feature Selection based on Anova F-test

ANOVA compares the mean value between the classes and decides whether any mean value vary from each other[10]. By using the F-test value, we can calculate the difference between the mean. The F- test value can be calculated by the following equation.

$$F = \frac{MS_B}{MS_W} \qquad (1)$$

Where, MS_B characterize the group variance and defined by the equation (2).

$$\frac{\sum_i n_i(\overline{x_i} - \bar{x})^2}{m-1} \qquad (2)$$

MS_w is defied by

$$\frac{\sum_i n_i(\overline{x_{ij}} - x_i)^2}{n-m} \qquad (3)$$

Here, n is the number of samples in i^{th} group. x refers the overall mean value and x_i refers the sample value of the mean.

III. Experimental Design

An RDD based supervised hyperspectral classification with complete spectral features is proposed in this section. Generally, Spectral features contain significant information for differentiating the materials obtained on the land. In order to improve the classifiers training speed and prediction results of image classification we used distributed computing framework where its computing data is fed into the Hadoop Distributed File System (HDFS) [11] and stored there. We used noise reduced spectral bands with its corresponding labels as input. This RDD based supervised classification have training and testing phases. Here, 70 percent of the total data was used for training, with the remaining 30 percent for testing the trained model. The hyperspectral imagery, along with its matching ground truth is saved in HDFS as input during the training stage. Following that, feature selection based on ANOVA, selected spectral features and their values are loaded into a supervised classifier using Distributed Spark ML. Spark MLlib provides many API's with supervised classifiers. It uses Spark's strong distributed engine to scale out classification on huge datasets. The classifier was then trained using the supplied ground truth. The learned model is then built. The residual 30% of samples in the testing are used to create the feature vector. Following that, the feature vector collected from the dataset is provided to the predictive model, which is a trained data from the supervised classifier. For each pixel, this prediction model generates the right class label. Fig. 1 shows how these training and testing procedures are carried out. The classifiers' evaluation is conducted using various elements like overall accuracy, specificity, sensitivity, and F1-score founded with the help of confusion matrix, as well as the predictive model's outcome and each classifier's execution duration.

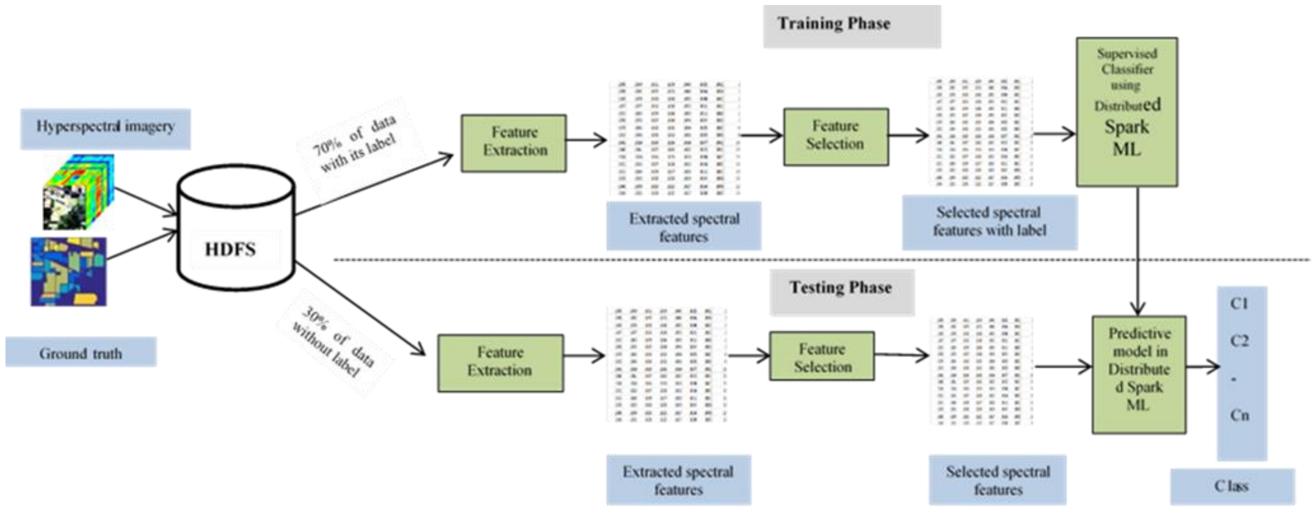
Training Phase
Hyperspectral imagery
70% of data with its label
Feature Extraction
Feature Selection
Supervised Classifier using Distributed Spark ML
Extracted spectral features
Selected spectral features with label
HDFS
Ground truth
Testing Phase
30% of data without label
Feature Extraction
Feature Selection
Predictive model in Distribute d Spark ML
C1
C2
.
Cn
Extracted spectral features
Selected spectral features
Class

Fig. 1. General Block Diagram of Hyperspectral Image Classification with Feature Selection.

IV. Experimental Configuration

To assess the proposed work described in Section III, we carried out the experiments on single node computer for Hyperspectral image classification focusing on ANOVA based feature selection. The configuration of single node processor intel core i7 7th generation, 16 GB RAM, Apache Spark – 1.6.0, Hadoop 3.2.2 and Linux 18.04.

V. Dataset

Experiments are conducted using commonly available hyperspectral dataset along with its reference ground data are available publicly on the website [12].

A. Indian Pines Dataset

This dataset was gathered throughout a vast region of Indian pines in north-western Indiana. The Airborne visible/infrared imaging spectrometer (AVIRIS) sensor takes images in 224 spectral bands with a spatial resolution of 20nm in the spectral range 0.43 to 0.86nm. Twenty water absorbed bands were removed and remaining 200 were processed for the experiment. Two-thirds of the image is cultivated land, while one-third is woodland. It is 145 x 145 pixels with 16 different categories.

B. Salinas

The second dataset Salinas, was gathered in California's Salinas Valley and has a high spatial resolution. The image, which covers the Salinas area, is 512 x 217 pixels in size and has 224 bands. This scene's ground truth has 16 classes of interest.

C. Salinas-A

Third dataset is Salinas-A, a minimal sub-scene of the Salinas image. It has a resolution of 86 × 83 pixels, 224 bands in the same area, and six classes.

D. University of Pavia

ROSIS sensor absorbed this dataset and it is generated over Pavia, northern Italy. It collects the images in 103 spectral bands with 610 x 610 pixels, and some of the samples contain no information. So, it is removed before analysis. There are nine different types of samples in this scenario ground truth.

VI. Performance Measures

The experimental results of each dataset were assessed using the evaluation metrics such as Accuracy, specificity, sensitivity and F1-score and the result of each classifier is compared with each other [13].

Accuracy: It is the ratio between number of correctly predicted data and total number of input values.

$$\boldsymbol{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4}$$

Specificity: Specificity refers the proportion between the number of true positive attributes and the number of positive classification results is known as specificity.

$$Specificity = \frac{True\ positives}{True\ positives + False\ positives} \tag{5}$$

Sensitivity: sensitivity is defined by the proportion between the number of true positive results and the total number of relevant samples.

$$Sensitivity = \frac{True\ positives}{True\ positives + False\ neagtives} \tag{6}$$

F1-score: The average of specificity and sensitivity is F1-score. This score varies between [0, 1]. We may determine the number of cases it properly classifies as well as the classifier's robustness from this.

$$F1 - score = 2 * \frac{(sensitivity\ X\ specificity)}{(sensitivity + specificity)} \tag{7}$$

VII. Experimental Analysis

This Section compares the performance classification results obtained using ANOVA feature selection method on different dataset. In order to get the efficiency of ANOVA method[13], classification is carried out using different number of features. Different feature combination was obtained using ANOVA. We let the first trial consist of 5 spectral features; second contains 10; subsequently the trials had number of features 50,100,150,180 and 200 for Indian pines test data. The reason for selecting different sets of features is to ensure that, fewer features could also obtain the comparable classification accuracy.

From Table I, it is evident that the overall classification accuracy is increases with number of features and beyond certain number of features, accuracy is not increasing. Hence, the features with highest accuracy are selected and compared with full features. For Indian Pines dataset we consider 150 number of features. Like that, for Salinas and Salinas-A we considered 150 features and for University of Pavia dataset we considered 100 selected features. The experimentations were carried out on multi-core machine in pseudo-distributed mode to perform classification on a single machine by creating a pseudo cluster (considering each core as a computer node) in spark [14]. To measure the performance, metrics such as execution time, accuracy, specificity, sensitivity and F1-score were computed. The results obtained from Indian pines dataset are tabulated in Table II. The experimentations were carried out on multi-core machine in pseudo-distributed mode to perform classification on a single machine by creating a pseudo cluster (considering each core as a computer node) in spark. To measure the performance, metrics such as execution time, accuracy, specificity, sensitivity and F1-score were computed. The results obtained from Indian pines dataset are tabulated in Table II. From Table II, it can be inferred that RF & DT discloses better accuracy compared with remaining two classifiers. LR and GNB obtains similar accuracy score of 51%. However, LR requires very short execution time than other 3 classifiers. As RF is an ensemble-based method, it produces better results than single classifier DT but RF requires more execution time than DT.

Table III compares the experimental results of Salinas dataset. It is observed that GNB performs poorly and achieved only 32% accuracy. Like Indian pines dataset, DT and RF perform equally good and but LR takes very less execution time.

TABLE I. OVERALL ACCURACY PERFORMANCE OF DIFFERENT NUMBER OF FEATURES ON INDIAN PINES

Supervised classifiers	**Number of selected features**						
	5 Features	***10 Features***	***50 Features***	***100 Features***	***150 Features***	***180 Features***	***200 Features***
Decision tree	59.16	59.93	59.41	61.69	63.24	61.40	61.40
Random Forest	58.38	59.62	60.68	61.74	64.98	63.69	63.00
Logistic Regression	51.24	51.42	51.05	51.02	51.63	51.65	51.69
Gaussian Naïve Bayes	19.18	21.55	23.74	24.63	30.91	51.41	50.78

TABLE II. EXPERIMENTAL RESULTS OF INDIAN PINES DATASET WITH 180 FEATURES

Metrics	**Decision Tree**	**Random Forest**	**LR**	**Gaussian NB**
Accuracy	61.40	63.69	51.41	51.41
Sensitivity	67.49	72.98	51.00	47.00
Specificity	67.49	72.98	51.41	47.00
F1-score	67.00	72.90	51.41	47.00
Time	0.0942	0.1111	0.0286	0.1036

TABLE III. EXPERIMENTAL RESULTS OF SALINAS DATASET WITH 180 FEATURES

Metrics	**Decision Tree**	**Random Forest**	**LR**	**Gaussian NB**
Accuracy	76.98	79.25	51.52	32.00
Sensitivity	62.78	59.51	51.52	30.00
Specificity	62.78	59.51	51.52	30.00
F1-score	62.78	59.51	56.41	47.00
Time	0.1086	0.0837	0.0311	0.0790

Experimental results of Salinas-A dataset are tabulated in Table IV from that we infer that RF achieves 63% of accuracy value and DT scores 61.40% LR and GNB produces equivalent result of 51% LR performs quickly than other classifiers it requires only 0.02 s.

Table V compares the results produced from urban based dataset called Pavia University from that we infer that GNB out performs RF accuracy of GNB depends on feature value but RF executes faster than GNB. LR performs lower than all classifiers. By comparing other metrics like specificity, sensitivity and F1-score value acquired from various classifiers, it is evident that RF performs well than other classifiers.

Fig. 2 illustrates the performance of different classifiers based on their accuracy value using proposed method. As shown in Fig. 2, it is clearly shown that RF performed better than all other classifiers.

Fig. 3 describes the performance of different classifiers based on their specificity value using proposed method. As shown in Fig. 3, it is clearly shown that RF performed better on large datasets like Indian pines, Salinas. DT classifier performed better on smaller dataset like Salinas-A. Traditional classifier GNB performed better than RF on Pavia University dataset.

TABLE IV. EXPERIMENTAL RESULTS OF SALINAS-A DATASET WITH 180 FEATURES

Metrics	**Decision Tree**	**Random Forest**	**LR**	**Gaussian NB**
Accuracy	61.40	63.69	51.41	51.41
Sensitivity	67.49	72.98	51.00	47.00
Specificity	67.49	72.98	51.41	47.00
F1-score	67.00	72.90	51.41	47.00
Time	0.0984	0.092	0.0225	0.802

TABLE V. EXPERIMENTAL RESULTS OF UNIVERSITY OF PAVIA WITH 100 FEATURES

Metrics	**Decision Tree**	**Random Forest**	**LR**	**Gaussian NB**
Accuracy	63.00	64.05	43.61	70.04
Sensitivity	62.57	64.00	43.28	70.00
Specificity	62.57	64.00	43.28	69.09
F1-score	62.57	64.00	43.28	69.09
Time	0.0982	0.1003	0.2620	0.7689

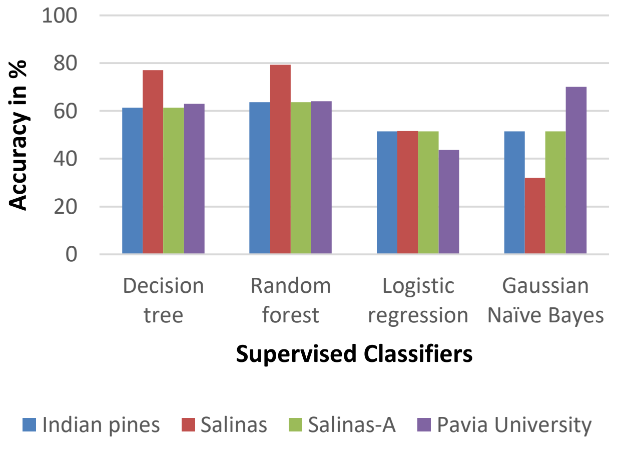

Fig. 2. Performance of Classifiers based on Accuracy Value.

Fig. 4 illustrates the performance of different classifiers based on their sensitivity value using proposed method. As shown in Fig. 4, it is clearly shown that RF performed better than other classifiers on all agricultural data set. GNB performed well on Pavia University dataset.

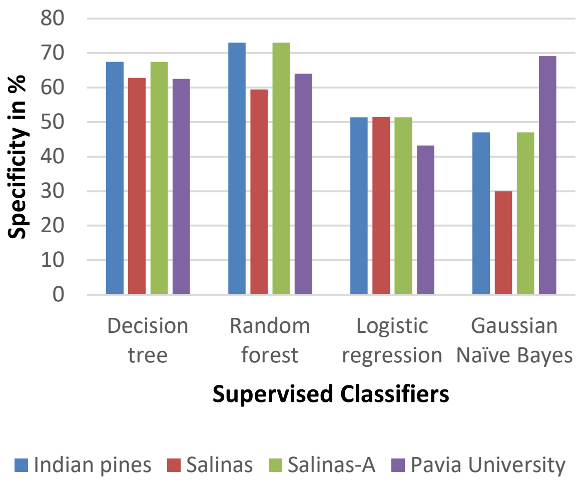

Fig. 3. Performance of Classifiers based on Specificity Value.

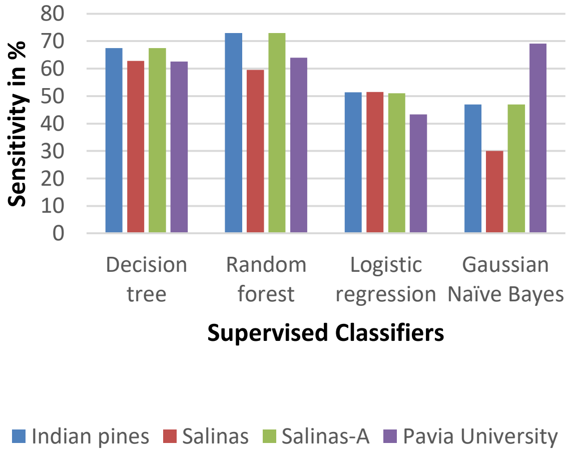

Fig. 4. Performance of Classifiers based on Sensitivity Value.

Fig. 5 illustrates the performance of different classifiers based on their F1-score value using proposed method. As shown in Fig. 5, it is clearly shown that like specificity and sensitivity RF performed better on all agricultural data set. GNB performed well on Pavia University dataset.

Fig. 6 compares the execution time of Indian pines data set on distributed mode using spark MLlib and normal classification method. It clearly shows that, classification using distributed processing reduces the computational time [15].

Fig. 7 compares the execution time of Salinas data set on distributed mode using spark MLlib and normal classification method. As shown in Fig. 7, it clearly shows that, classification using distributed processing reduces the computational time [15].

Fig. 8 compares the execution time of Salinas-A data set on distributed mode using spark MLlib and normal classification method. As shown in Fig. 8, it shows that classification using distributed processing reduces the computational time [15].

Fig. 9 compares the execution time of University of Pavia data set on distributed mode using spark MLlib and normal classification method on different classifiers. As shown in Fig. 9, it clearly shows that, classification using distributed processing reduces the computational time[15].

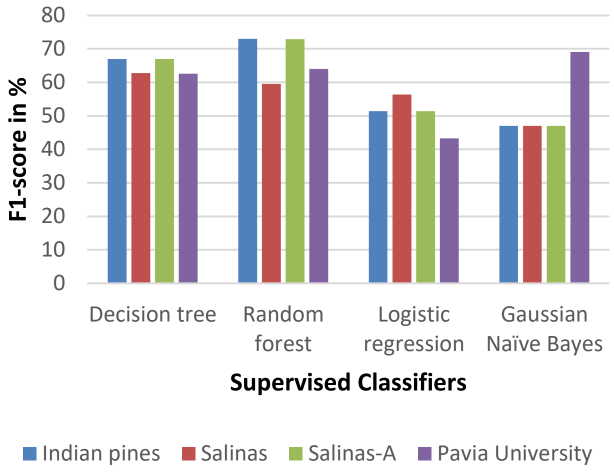

Fig. 5. Performance of Classifiers based on F1-score Value.

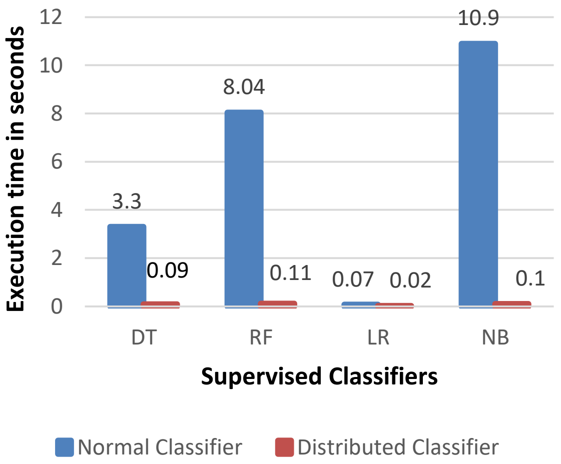

Fig. 6. Execution Time of Various Classifiers on Indian Pines Dataset.

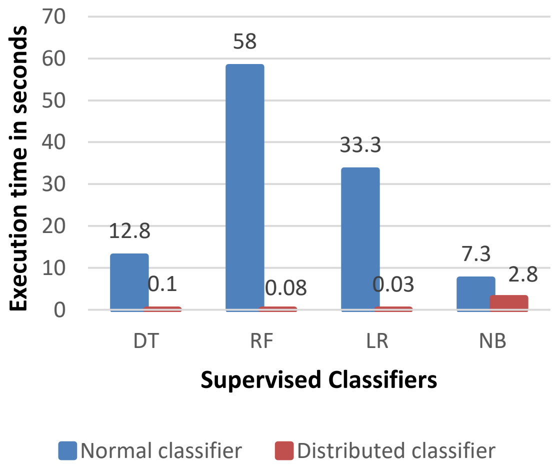

Fig. 7. Execution Time of Various Classifiers on Salinas Dataset.

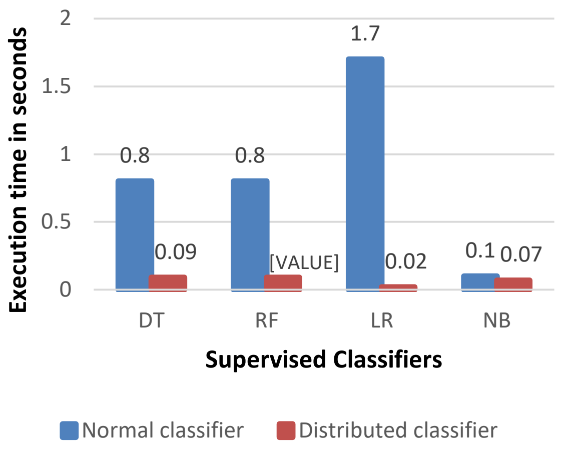

Fig. 8. Execution Time of Various Classifiers on Salinas-A Dataset.

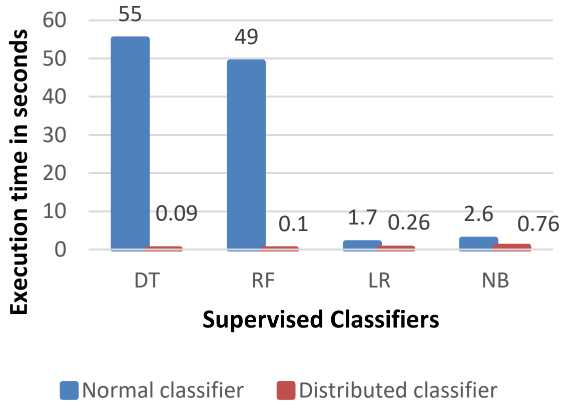

Fig. 9. Execution Time of Various Classifiers on Pavia University Dataset.

VIII. Conclusion

The proposed method uses, Spark based distributed environment for classification of Hyperspectral images with ANOVA feature selection. By comparing it with performance of normal classification methods, the proposed method leads very less computational time and produces good accuracy. Also, we found that distributed method reduces the computational time. As a conclusion remark, Random Forest and Decision tree method of classification produces better accuracy for given hyperspectral dataset. This work uses spectral data for classification of high dimensional hyperspectral image. As a future work, spatial related feature and the fusion of spatial-spectral features can be considered to achieve better classification results with reduced computational time.

References

[1] R. O. Green et al., “Imaging spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS),” Remote Sensing of Environment, vol. 65, no. 3, pp. 227–248, 1998.

[2] R. Ragupathy and N. Aswini, “Performance Comparison of Filter-Based Approaches for Display of High Dynamic Range Hyperspectral Images,” in Advances in Intelligent Systems and Computing, 2020, vol. 1079, pp. 79–89.

[3] S. Misra and S. Bera, “Introduction to Big Data Analytics,” Smart Grid Technology, pp. 38–48, 2018.

[4] J. A. Richards, “Analysis of remotely sensed data: The formative decades and the future,” IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pp. 422–432, 2005.

[5] X. Meng et al., “MLlib: Machine learning in Apache Spark,” Journal of Machine Learning Research, vol. 17, pp. 1–7, 2016.

[6] R. Sharma, A. Ghosh, and P. K. Joshi, “Decision tree approach for classification of remotely sensed satellite data using open source support,” Journal of Earth System Science, vol. 122, no. 5, pp. 1237–1247, 2013.

[7] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, “Random Forest Classifiers for Hyperspectral Data,” IEEE, pp. 160–163, 2005.

[8] N. Aswini and R. Ragupathy, “On Appraisal of Spectral Features Based Supervised Classifications for Hyperspectral Images,” International Journal of Recent Technology and Engineering, vol. 8, no. 6, pp. 593–600, 2020.

[9] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification, second edition. 2001.

[10] B. Venkatesh and J. Anuradha, “A review of Feature Selection and its methods,” Cybernetics and Information Technologies, vol. 19, no. 1, pp. 3–26, 2019.

[11] “HDFS Architecture Guide.” [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

[12] “Hyperspectral Remote Sensing Scenes - Grupo de Inteligencia Computacional (GIC).” [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[13] N. O. F. Elssied, O. Ibrahim, and A. H. Osman, “A novel feature selection based on one-way ANOVA F-test for e-mail spam classification,” Research Journal of Applied Sciences, Engineering and Technology, vol. 7, no. 3, pp. 625–638, 2014.

[14] “Hadoop - Different Modes of Operation - GeeksforGeeks.” [Online]. Available: https://www.geeksforgeeks.org/hadoop-different-modes-of-operation/.

[15] N. Aswini and R. Ragupathy, “ANOVA F-test based Framework for Supervised Classifiers on Classification of Hyperspectral Images,” vol. 26, no. 12, pp. 394–403, 2020.