International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

On Appraisal of Spectral Features Based Supervised Classifications for Hyperspectral Images

N. Aswini, R. Ragupathy

Abstract*: The classification of hyperspectral images is a challenging task since it contains unbalanced ratio between the training and testing samples, and number of spectral bands. The detailed spectral data of hyperspectral images increases the ability to individualize the different classes and achieving accurate classification maps. Hence, in this paper, we use spectral data for classification and we address the performance of different supervised classification techniques like logic-based, ensemble-based, statistical-based, non-probabilistic-based and instance-based techniques on spectral features. Experiments are carried out using hyperspectral imagery captured by AVIRIS sensor such as Indian Pines, Salinas and Salinas-A. The appraisal of these supervised classification methods are held with each other in terms of performance metrics such as overall accuracy, precision, recall, F1-score and execution time.

Keywords: Hyperspectral images, Logic based classifier, Ensemble classifier, Non-Probabilistic classifier, Statistical classifier, Instance based classifier.

I. INTRODUCTION

Hyperspectral data are becoming a vital tool for observing the Earth's surface; the high spectral resolution of an acquired image provides the ability to detect, identify and classify the objects more accurately than multi spectral imagery. A 3-Dimensional hyperspectral image cube consists of hundreds of co-registered images for the same image scene taken in very narrow spectral bands. But, efficiently handling such a large amount of data, by taking optimum spectral information is a difficult task. So, it leads the researchers to concentrate on hyperspectral remote sensing. The purpose of hyperspectral image classification is to classify all the pixels in hyperspectral imagery into individual classes automatically. Usually, we encounter two main challenges during classifying the images. First one is a curse of dimensionality which is also termed as Hughes phenomenon [1]. Another one is performing with inadequate training samples.

In recent years, in order to overcome these problems and to achieve better results, many machine learning methods have been proposed to deal with hyperspectral classification. In 2005, Sveinn R. Toselsson et al. proposed different random forest method for the classification of hyperspectral imagery [2]. In 2006, Shuji Kawaguchi et al. proposed Adaboost by decision stump method for classification of remotely sensed images [3]. T.A. Moughal suggested support vector machine to deal with multi class problem in hyperspectral image [4]. Richa sharmal et al. developed Decision tree classification method for Remotely sensed hyperspectral image [5] . In 2008, Kaiguang zhao et al. introduces Gaussian for Bayesian learning in remote sensing [6]. In 2013, Haoliang yuan, et al. proposed spectral-spatial linear discriminant analysis for Hyperspectral images [7]. In 2014, Rig Das et al. worked on quadratic discriminant analysis [8]. In 2016, Weiwei song proposed K-nearest neighbour based classification method [9]. In 2018, Kyle Loggenberg applied Extreme Gradient boosting method for classification of Hyperspectral images [10] . Therefore, this paper reviews the performance of matured supervised machine-learning classifiers such as SVMs, Decision Tree (DT), Random Forest, Ada boost, and k-nearest neighbour (K-NN) [11], Gradient boosting [12], Gaussian Naive Bayes (NB) [13], Linear Discriminant Analysis (LDA), Quadratic discriminant Analysis (QDA) [10] based on spectral features. Spectral features based hyperspectral data classification has been considering in recent works [11]. L.Fiumi proposed an approach based on spectral signature with standard [14]. S. Roessner et.al, proposed an advanced method, automated differentiation method of urban hyperspectral data [15] and identification of spectral signatures on mixed pixels [16]. This paper describes the following: 1) we tested the performance of different classification methods with spectral features 2) performance of classifiers are compared with each other. Rest of the paper is sequenced in following manner. Section II reviews different classification methods. Section III reports the Experimental design. Section IV reports the Experimental results and discussion and Section V draws the conclusion of the overall discussion in brief.

II. CLASSIFICATION METHODS

A brief introduction to the supervised machine learning techniques such as logic based, Ensemble, Non-Probabilistic, Statistical and Instance based techniques is given this Section. These supervised machine learning techniques are classified as shown in Fig. 1.

Manuscript received on February 10, 2020.
Revised Manuscript received on February 20, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author
N. Aswini*, Division of Computer and Information Science, Faculty of Science, Annamalai University, Annamalainagar, Tamil nadu 608002, India. Email: aswini.it03@gmail.com
R. Ragupathy, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalainagar, Tamil nadu 608002, India. Email: cse_ragu@yahoo.com

A. Logic based technique

Logic based learning technique aims at learning rule based on creating hypotheses, from the observations.

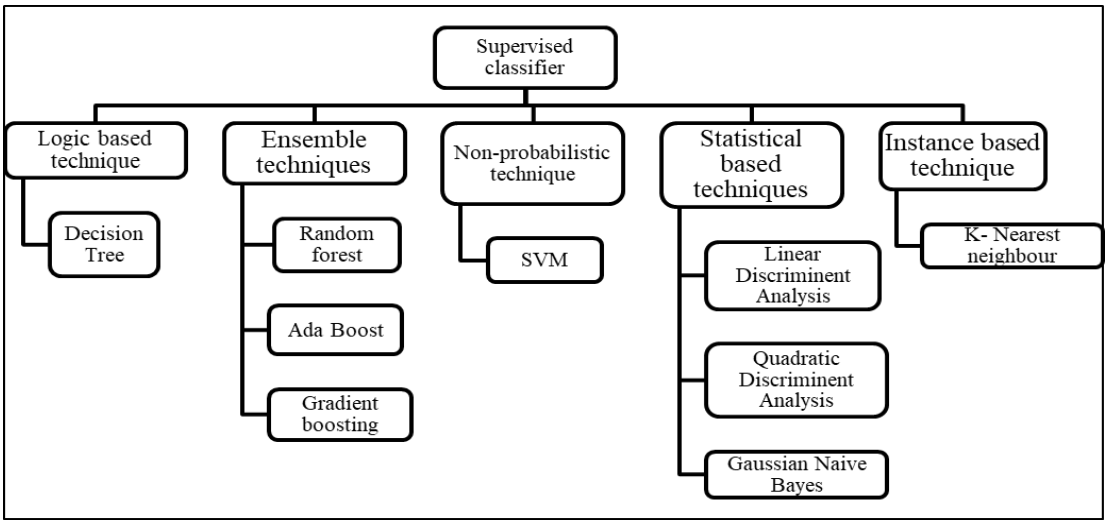

Fig. 1. Supervised classifier techniques

Decision Tree

Decision tree divides the given data into more than one related group. This group is made up of most relevant attributes/ independent variables to create as many distinct groups as possible [5].

For classification purpose, Decision tree uses Gini's default also it uses Entropy as an alternative.

$$Gini = \sum_{i=1}^{c} f_i \ (1 - f_i) \quad (1)$$

$$Entropy = \sum_{i=1}^{c} -f_i \log (f_i) \quad (2)$$

Where, f denotes the frequency of level i at a node and C denotes the number of unique labels.

The following Fig. 2 shows an example of simple decision tree.

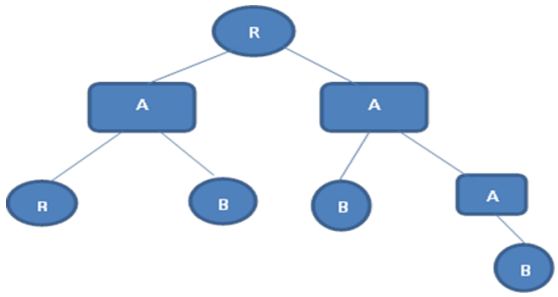

Fig. 2. Simple example for decision tree

In the image R represents root node; A denotes the internal node and B represents leaf node.

B. Ensemble technique

Ensemble algorithm is combination of one or more algorithms of same or different kind to classify the objects. Random forest, Ada Boost and Gradient boosting are some of the ensemble based algorithms

Random forest

To overcome the limitations of single decision tree like over fitting the data, Random forest classifier creates a set of decision trees. To classify a new object based on its type, each tree returns a report and we call it as tree "votes" for that class. The forest selects the classification result, which is having maximum number of votes [2]. Ensemble of classifiers is given as h1(x), h2(x),...,hK (x), and with the training set drawn from the random vector X,Y define the margin function as

$$mg(X,Y) = avk\ I(hk\ (X) = Y) - max\ j \neq Y\ avk\ I(hk\ (X) = j) \quad (3)$$

Where, $I()$ is the indicator function. Fig. 3 illustrates the techniques of Random forest classifier.

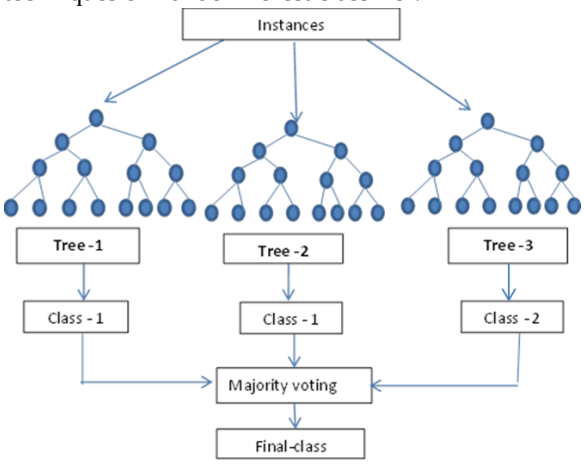

Fig. 3. Random forest technique

Ada Boost

Adaboost or adaptive boosting is also an ensemble classifier using decision tree like Random forest. It is important to choose an appropriate set of weak classifiers. Binary decision stumps are always candidates for weak classifiers. Moreover, they are superior in extraction of useful features for classification [3].

$$H(x) = sign\ (\sum_{t=1}^{T} \propto_t h_t\ (x)) \quad (4)$$

Where, H (combined classifier) is computed as a weighted majority vote of the h_t denotes weak classifier where each is assigned weight $\propto_t$. Fig. 4 shows the formation of simple boosted classifier.

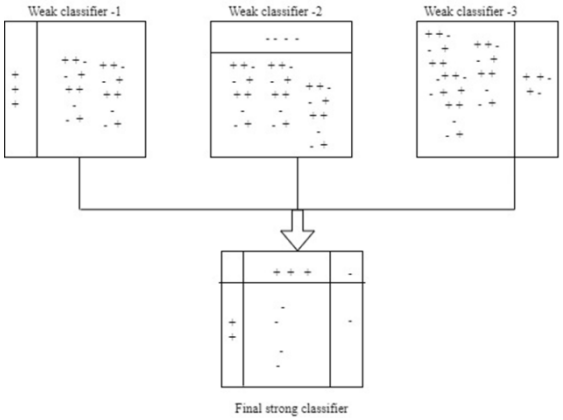

Fig. 4. Boosted classifier

Gradient Boosting

Like Random Forests, Gradient Boosting is also an ensemble method. This means, it creates a final result based on a group of single models. The prediction of this single model will increase its performance by combing two or more weak models. This ensemble method will lead to maximised accuracy rate [12].

$$F_m\ (x) = F_{m-1}(x) + r_m\, h_m(x) \quad (5)$$

Where, F_m the final strong classifier, r is is the pseudo residuals.

C. Non-Probabilistic technique SVM

SVM is a non-probabilistic classification method which creates a decision boundary in multi-dimensional space using a subset of the training vectors. The elements of this subset are known as support vectors. Support vectors are the training models that are closer to the decision boundary. Generally, support vectors achieve maximum accuracy rate compared to any other classification techniques like maximum likelihood. SVM provides increased accuracy rate especially when the input contains multiple class [4]. Fig. 5 shows the example for good marginal separation between two classes.

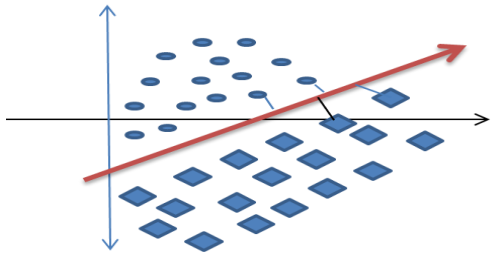

Fig. 5. Support vector machine algorithm

The hyperplane is defined by the below (6)

$$w.x + b = 0 \quad (6)$$

where, x is the point on hyperplane

From (7) we can find the optimal hyperplane of the data

$$\min_{w,b} \frac{1}{2}\,|w|^2$$

$$y_i\,(w.x_i + b) - 1 \geq 0 \text{ where, } i = 1,..,1 \quad (7)$$

D. Instance based technique

Instance based methods usually compares new problem instances with instances which is already seen in training phase, which have been stored in memory.

K- Nearest neighbour

K-NN is a traditional well-known classification algorithm. Unlike other methods, it is not trained to produce a model. Instead, each unknown sample is directly compared with the original data. The unknown sample is assigned to the most common class of the k training samples that are nearest in the feature space to the unknown sample. A low k will therefore produce a very complex decision boundary; a higher k will result in greater generalization. Because a trained model is not produced, K-NN classification would be expected to require greater resources as the number of training sample increases [9]. The distance between the training and testing values are measured by Euclidean distance given in (8). Fig. 6 shows the steps to classify using K-NN.

Euclidean distance: $D(X,Y) = \sqrt{\sum_{i-1}^{N}(x_i - y_i)^2}$ (8)

Where, x and y are the distance points in the region.

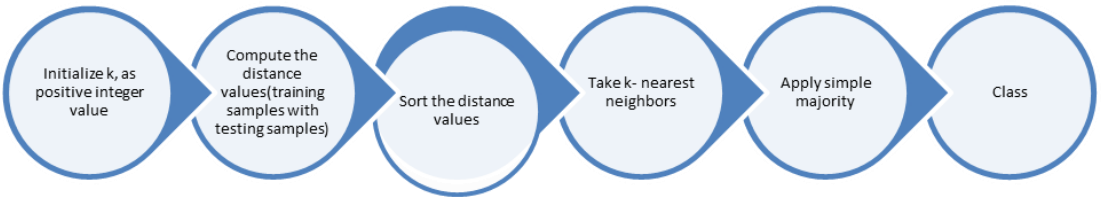

Fig. 6. K-NN algorithm

E. Statistical based technique

Statistical learning theory deals with the problem of finding a predictive function based on data.

Naive Bayes

Naive Bayes is a simple Bayesian networks. It is made up of directed acyclic graph which contains only one parent node and several children nodes with a strong belief that, independence between child nodes in context of the parent. Thus, the Naive Bayes (independence model) is established by valuing and comparing these two probabilities, the higher probability value shows that the class label value that is similar to be the actual [13]. If the maximum likelihood of features is assumed to be Gaussian,

the probability is given in (9). Fig. 7 shows the distribution of Naïve Bayes.

$$p(x_i|y_i) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (9)$$

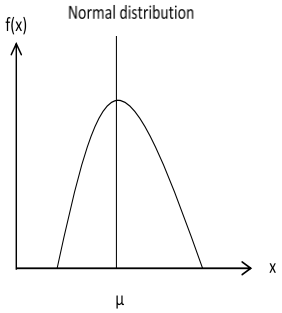

Fig. 7. Naive Bayes

LDA

The linear discriminant analysis (LDA) is developed to change the features from high dimensional space into lower, which increases the ratio of the between-class variance to the within-class variance, so it maximizes the class separability. Class-dependent and class-independent are the two different types of LDA techniques [7]. Fig. 8 shows an example for LDA.

$$\hat{\delta}_k(x) = x.\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}^2 k}{2\hat{\sigma}^2} + log(\hat{\pi}_k) \quad (10)$$

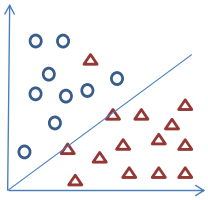

Fig. 8. Linear discriminant analysis

QDA

Quadratic discriminant analysis (QDA) is closely connected with LDA, where the values from the individual classes are distributed normally. But contrary to that, in QDA there is no assumption that covariance of each classes is similar [8]. Fig. 9 shows a simple example for QDA.

$$\delta_k(x) = -\frac{1}{2} log\ |\sum_k - \frac{1}{2}\mu_k\ (x-\mu_k)^T \sum_k^{-1}(x-\mu_k) + log\pi k \quad (11)$$

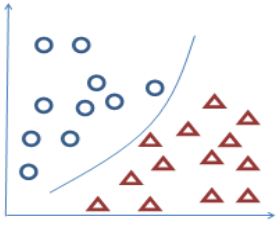

Fig. 9. Quadratic discriminant analysis

III. EXPERIMENTAL DESIGN

In this section, we propose a pure spectral feature based Hyperspectral classification since spectral features usually contain important information for discriminating different kinds of materials presented on the ground. Also, we use all the raw data of spectral bands and its corresponding ground truth label values as an input. This supervised classification contains two phases: training and testing. A general phenomenon behind the selection of samples in training phase is between 70% and 80 % and remaining samples for testing. Here, we have chosen 70% of samples for training and 30% samples for testing.

In the training phase, all the bands in hyperspectral imagery with its corresponding ground truth image are provided to the system as input. After feature extraction, all the spectral features with their labels are fed into supervised classifier as discussed in Section II. Further, classifier learns with the help of ground truth. Finally, the trained model is formed. In the testing phase, feature vector is formed from the remaining 30% of samples. Then, the feature vector extracted from dataset is fed into the learned supervised classifier known as predictive model. This predictive model produces the correct class label for each pixel. These training and testing processes are performed as depicted in Fig. 10. The performance of the classifiers are evaluated using the parameters such as Overall accuracy, Precision, Recall, and F1-score based on the confusion matrix formed with the help of ground truth and outcome of the predictive model and execution time of each classifiers as discussed in Section IV.

IV EXPERIMENTAL RESULTS AND DISCUSSION

A. Datasets

Experiments are demonstrated on commonly used hyperspectral dataset collected using AVIRIS sensor which is available publically on the website [17].

Indian pines: The Indian pines dataset was captured over the area of Indian pines set in the North western Indiana. The AVIRIS sensor collects images in 220 spectral bands in the spectral range 0.43 to 0.86m at 20m spatial resolution. This original image contains two-third of agricultural land and one-third of forest.145X145 pixels with 16 class of interest. Here, water absorbed regions are considered as class 0.

Salinas: The second dataset named Salinas was collected over Salinas Valley, California, and is characterized by high spatial resolution. The area covered image comprises of 512X217 pixels with 224 bands. This scene ground truth contains 16 class of interest. In this paper, water absorbed area in Salinas considered as class of 0.

Salinas-A: Third dataset is a small sub scene of Salinas image, denoted as Salinas-A. It comprises of 86X83 pixels with 224 bands located within the same scene and includes six classes.

Class name with their samples of Indian pines, Salinas-A and Salinas datasets [17] are tabulated in I, II and III respectively.

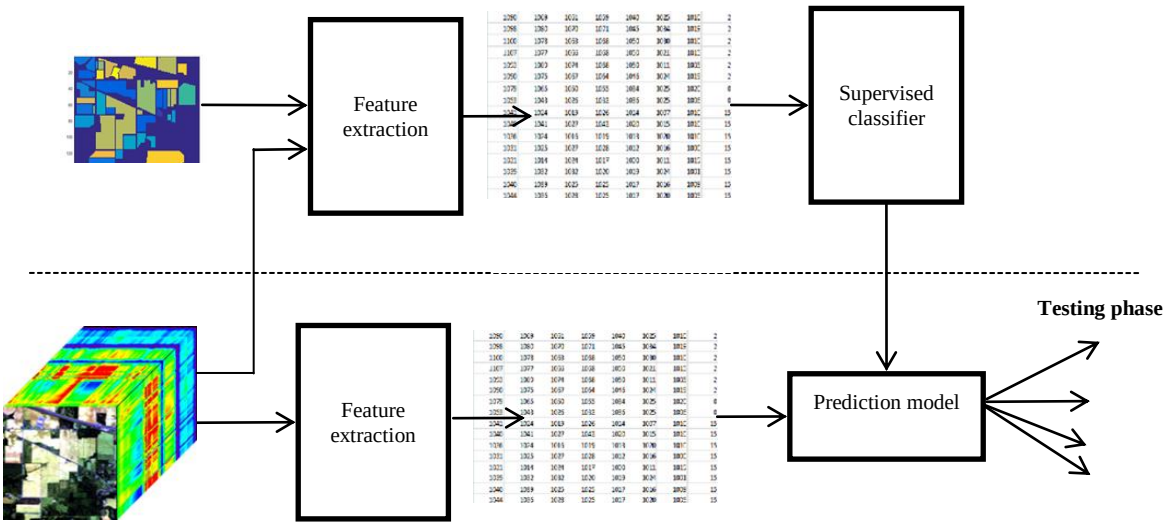

Fig. 10. General block diagram of Hyperspectral image with supervised classification

Table- I: Information of each class samples of Indian pines

Class No.	Class name	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

Table- II: Information of each class samples of Salinas-A

Class No.	Class	Samples
1	Brocoli_green_weeds_1	391
2	Corn_senesced_green_weeds	1343
3	Lettuce_romaine_4wk	616
4	Lettuce_romaine_5wk	1525
5	Lettuce_romaine_6wk	674
6	Lettuce_romaine_7wk	799

From the available ground truth samples, we randomly selected 80 per cent samples for training and 20 per cent for testing. In both the datasets, all the pixels are considered as a feature and pixels used in the training set are excluded from the testing dataset. The experimental results were assessed by overall accuracy (OA), Precision, Recall, F1-score and execution time. The results of each classifier are compared with each other.

Table- III: Information of each class samples of Salinas

Class No.	Class	Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

Overall Accuracy: It is the ratio between number of correctly predicted data and total number

of input values.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (12)$$

Precision: It is the ratio between number of true positive values and number of positive results predicted by the classifier.

$$Precision = \frac{True\ positives}{True\ positives+False\ positives} \quad (13)$$

Recall: It is the ratio between number of true positive results and the number of all relevant samples.

$$Recall = \frac{True\ positives}{True\ positives+False\ neagtives} \quad (14)$$

F1-score: F1 Score is the mean value between precision and recall. F1 Score ranges from [0, 1]. From that we can infer, how many instances it classifies correctly, and also how robust the classifier is.

$$F1-score = 2 * \frac{(Recall\ X\ Precision)}{(Recall+precision)} \quad (15)$$

B. Discussion

These supervised classifiers are implemented in Python version 3.2 under the System configuration of Intel Core i7-7500U CPU @ 2.70GHz 2.90 GHz with 16.0 GB memory. In this section, the performance of these classifiers are evaluated using metrics like Precision, Recall and F1-score [18], and also time taken to process the Indian pines, Salinas and Salinas-A datasets are tabulated in IV,V and VI. Fig. 11 shows the performance comparison of all the classifiers for three publically available dataset in terms of overall accuracy.

Table- IV: Experimental results of Indian pines

METRICS	K-NN	Decision tree	Random forest	Ada boost	Gradient boost	Gaussian NB	LDA	QDA	SVM
Accuracy (%)	71.22	68.31	77.55	48.24	78.17	30.83	68.29	60.92	50.74
Precision (%)	73.00	69.00	77.00	48.00	78.00	31.00	68.00	61.00	51.00
Recall (%)	72.00	69.00	76.00	38.00	78.00	58.00	72.00	66.00	26.00
F1-score	0.71	0.69	0.76	0.42	0.77	0.29	0.69	0.59	0.34
Time	5.88(s)	3.9(s)	9.37(s)	10.31(s)	26(m)	10.31(s)	5.00(s)	26.25(s)	43(m)

Table- V: Experimental results of Salinas

Metrics	K-NN	Decision tree	Random forest	Ada boost	Gradient boost	Gaussian NB	LDA	QDA	SVM
Accuracy (%)	89.82	89.06	92.18	50.72	90.76	47.41	80.73	83.86	-
Precision (%)	90.00	89.00	92.00	51.00	91.00	47.00	81.00	84.00	-
Recall (%)	90.00	89.00	92.00	37.00	91.00	76.00	84.00	87.00	-
F1-score	0.90	0.89	0.92	0.40	0.91	0.45	0.81	0.84	-
Time	1.19(m)	1.87(s)	1.9(s)	1.21(m)	7.0(m)	1.54(s)	4.3(s)	2.11(s)	-

Table- VI: Experimental results of Salinas-A

METRICS	K-NN	Decision tree	Random forest	Ada boost	Gradient boost	Gaussian NB	LDA	QDA	SVM
Accuracy (%)	85.80	85.80	87.58	61.90	89.02	65.87	81.41	70.72	54.34
Precision (%)	86.00	86.00	87.00	62.00	89.00	66.00	81.00	71.00	54.00
Recall (%)	85.00	86.00	88.00	57.00	89.00	69.00	84.00	78.00	77.00
F1-score	0.85	0.86	0.87	0.56	0.89	0.61	0.81	0.70	0.56
Time	39.56(s)	0.11(s)	0.59(s)	5.08(s)	3.11(s)	3.19(s)	0.27(s)	6.68(s)	3.45(m)

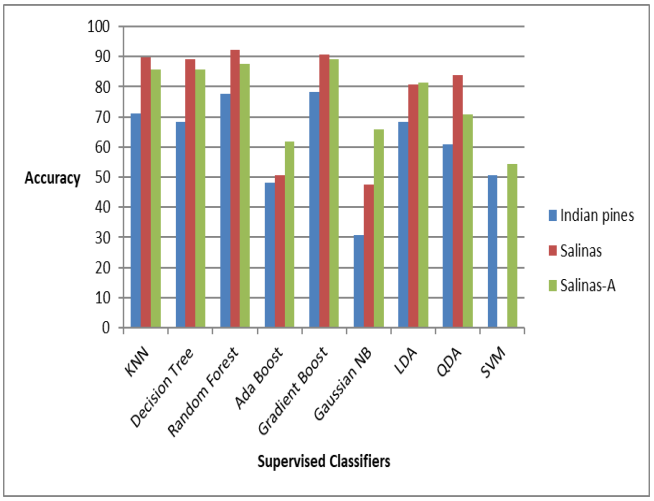

Fig. 11. Performance evaluations of datasets using Accuracy value

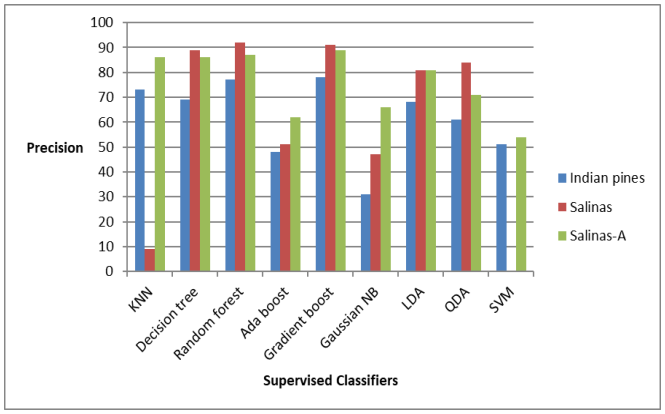

Fig. 12. Performance evaluations of datasets using Precision value

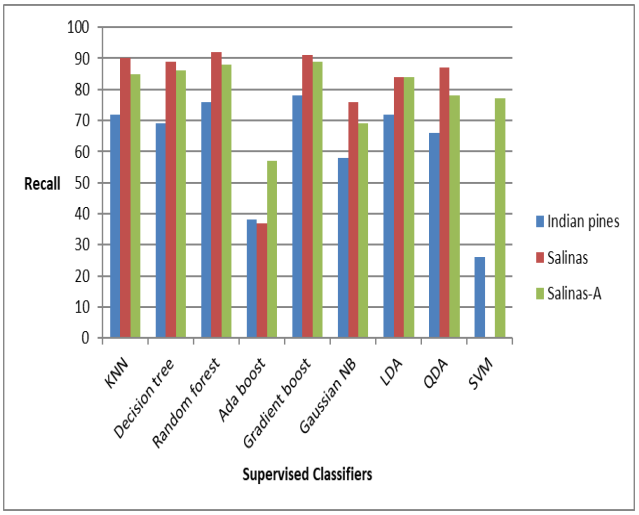

Fig. 13. Performance evaluations of datasets using Recall value

From Fig. 11, we infer that, Gradient boosting and Random forest achieves more or less similar results for all three datasets, which is higher than all other classifiers. Gradient boost out performs Random forest in Indian pines and Salinas-A datasets which is smaller dataset than Salinas, So,

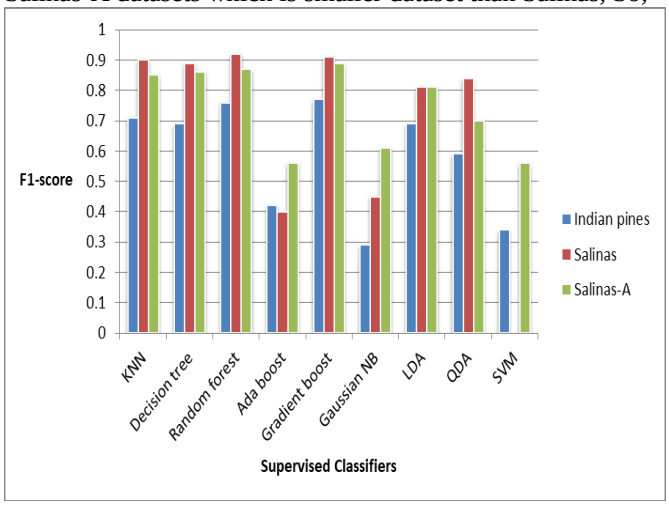

Fig. 14. Performance evaluations of datasets using F1-score Value

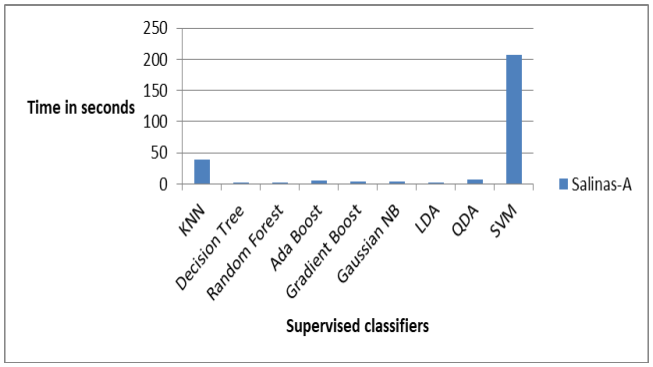

Fig. 15. Time taken to classify of Salinas-A dataset

Random forest gives much accurate result to the dataset which contains higher number of training and testing samples. K-NN outperforms Single decision tree and gives reasonable accuracy for both Indian pines and Salinas but for Salinas-A both K-NN and decision tree produces same result. It shows that ensemble based classifiers produces better accuracy rate on imbalanced training and testing dataset than single classifier like decision tree. Because of limited training samples, SVM and Adaboost have shown sensitive to the dataset. That is why, SVM got lower accuracy score on Salinas-A. Traditional statistical based method QDA and LDA performs better than SVM, where LDA gives better results on Indian pines dataset and Salinas-A, QDA results better than LDA on Salinas. Gaussian NB produces lower accuracy rate on both Indian pines and Salinas dataset but on Salinas-A Gaussian NB gives better performance than SVM. Fig. 12, 13 and 14 illustrate the comparative evaluation of supervised classifiers in terms of precision, recall and F1-score values. It is seen that, like accuracy value Gradient boost shows better precision, recall and F1-score value than other classifiers on all three datasets. Following that, Random forest, DT, and K-NN shows similar value for Salinas and Salinas-A dataset. LDA gives better result than QDA on smaller dataset like Indian pines and Salinas-A, but on Salinas, QDA gives better performance on Salinas. Ada boost and SVM produces second lowest precision value and Gaussian NB produces lowest precision value. But, Gaussian NB performs better than Ada boost classifier on recall and F1-score value. Fig. 15 illustrates time taken by the classifiers to perform classification on Indian pines dataset. From that, we observed that classification time of SVM is longer than all other classifiers. Generally, training time of SVM is much longer as it is much more computationally intensive and classification time which require around 3 minutes for the entire process. Next to that, K-NN needs more time to perform Salinas-A dataset. All other classifiers utilizes only lesser period of time among them Decision tree needs only 1.3 seconds for the process which is lesser than rest of other classifiers.

V CONCLUSION

Generally, machine learning algorithms provide better classification performance on remote sensing images. Hence, this paper analyse different supervised classifiers in terms of performance based on spectral features of hyperspectral images and we found that the sample size and quality of training samples have shown huge impact on accuracy of classification. Further, we found that time taken by SVM is longer than all other classifiers. Therefore, it is better to choose better quality and large samples to get better performance in terms of accuracy. As a conclusion remark, it is better to use ensemble Decision tree method like Random forest and Gradient boosting methods, if size of the training sample is small.

REFERENCES

1. F. G. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," IEEE Trans.inf. Theory, vol. 14, no. I, pp. 55–63, 1968.
2. S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random Forest Classifiers for Hyperspectral Data," IEEE, pp. 160–163, 2005.
3. S. Kawaguchi, R. Nishii, and A. A. Learning, "Hyperspectral Image Classification by AdaBoost with Decision Stumps Based on Composed Feature Variables," IEEE, pp. 928–931, 2006.
4. T. A. Moughal, "Hyperspectral image classification using Support Vector Machine," J. Phys. Conf. Ser., 2013.
5. R. Sharma, A. Ghosh, and P. K. Joshi, "Decision tree approach for classification of remotely sensed satellite data using open source support," J. Earth Syst. Sci., vol. 122, no. 5, pp. 1237–1247, 2013.

6. K. Zhao, S. Popescu, and X. Zhang, “Bayesian learning with gaussian processes for supervised classification of hyperspectral data,” Photogramm. Eng. Remote Sensing, vol. 74, no. 10, pp. 1223–1234, 2008.
7. H. Yuan, Y. Lu, L. Yang, H. Luo, and Y. Y. Tang, “Spectral-Spatial Linear Discriminant Analysis for Hyperspectral Image Classification,” in IEEE International Conference on Cybernetics, 2013, pp. 144–149.
8. R. Das, R. Dash, and B. Majhi, “Hyperspectral Image Classification Based on Quadratic Fisher ’ s Discriminant Analysis and Multi-class Support Vector Machine,” IETE J. Res., vol. 60, no. December, pp. 37–41, 2014.
9. K. H. Weiwei Song , Shutao Li , Xudong Kang, “Hyperspectral Image Classification Based on KNN Sparse Representation,” in IGARSS, 2016, pp. 2411–2414.
10. K. Loggenberg, A. Strever, B. Greyling, and N. Poona, “Modelling water stress in a Shiraz vineyard using hyperspectral imaging and machine learning,” Remote Sens., vol. 10, no. 2, pp. 1–14, 2018.
11. C. Bo, H. Lu, and D. Wang, “Spectral-spatial K-Nearest Neighbor approach for hyperspectral image classification,” Multimed. Tools Appl., 2017.
12. L. Wei, Z. Yuan, Y. Zhong, L. Yang, X. Hu, and Y. Zhang, “An Improved Gradient Boosting Regression Tree Estimation Model for Soil Heavy Metal (Arsenic) Pollution Monitoring Using Hyperspectral Remote Sensing,” Appl. Sci., vol. 9, 2019.
13. C. Solares and A. M. Sanz, “Different Bayesian Network Models in the Classification of Remote Sensing Images,” in Intelligent Data Engineering and Automated Learning - IDEAL, 2007, pp. 10–16.
14. L. Fiumi, “Evaluation of MIVIS Hyperspectral Data for Mapping Covering Materials,” in IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas Analysis, 2001, pp. 324–327.
15. U. Heiden, S. Roessner, K. Segi, and H. Kaufmann, “Analysis of Spectral Signatures of Urban Surfaces for their Identification Using Hyperspectral HyMap Data,” in IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas Analysis, 2001, pp. 1–5.
16. S. Roessner, K. Segl, U. Heiden, and H. Kaufmann, “Automated Differentiation of Urban Surfaces Based on Airborne Hyperspectral Imagery,” IEEE Trans. Geosci. Remote Sens., vol. 39, no. 7, pp. 1525–1532, 2001.
17. Hyperspectral Remote Sensing Scenes, Computational Intelligence Group, Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.
18. Important Model Evaluation Error Metrics Everyone should know, Available:https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/.

AUTHORS PROFILE

N. Aswini received the M.Sc. degree in Information Technology from Annamalai University in 2014. She received the M.Phil. degree in Computer Science from Thiruvalluvar University in the year 2016. She is currently doing Ph.D. in Computer Science at Annamalai University, from the year 2017. She has published 3 papers in international conferences and journals. She got best presentation award in International Conference on Research and Developments in Engineering & Technology, Management & Sciences conducted by International Association of Research and Developments in Engineering and Technology at Chennai. Her research interest includes image processing, Hyperspectral image classification and big data analytics.

R. Ragupathy received the B.E.degree in Computer Science and Engineering from Annamalai University in 1999. He received the M.E degree in Computer Science and Engineering from Bengal Engineering and Science University under quality improvement programme (QIP), in the year 2004. He has been with Annamalai University, since 1999. He completed his Ph.D. degree in Computer Science and Engineering, Annamalai University, in the year 2014. He published 25 papers in international conferences and journals. He is professionally affiliated with bodies like ISTE, CSI, IAENG, etc., as a member. He is a reviewer in journals like Egyptian Informatics Journal, Elsevier, Journal of Computer Science, Science Publication, IETE Journal of Research, Taylor and Francis Publication etc., His research interest includes image processing, mobile ad-hoc networks, pattern recognition, and big data analytics.