I.J. Image, Graphics and Signal Processing, **2019, 4, 55-62**
Published Online April 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2019.04.05

Edibility Detection of Mushroom Using Ensemble Methods

Nusrat Jahan Pinky, S.M. Mohidul Islam, Rafia Sharmin Alice
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
Email: nusratjahan1535@gmail.com, mohid@cse.ku.ac.bd, awesomealice91@gmail.com

Received: 10 March 2019; Accepted: 28 March 2019; Published: 08 April 2019

Abstract—Mushrooms are the most familiar delicious food which is cholesterol free as well as rich in vitamins and minerals. Though nearly 45,000 species of mushrooms have been known throughout the world, most of them are poisonous and few are lethally poisonous. Identifying edible or poisonous mushroom through the naked eye is quite difficult. Even there is no easy rule for edibility identification using machine learning methods that work for all types of data. Our aim is to find a robust method for identifying mushrooms edibility with better performance than existing works. In this paper, three ensemble methods are used to detect the edibility of mushrooms: Bagging, Boosting, and random forest. By using the most significant features, five feature sets are made for making five base models of each ensemble method. The accuracy is measured for ensemble methods using five both fixed feature set-based models and randomly selected feature set based models, for two types of test sets. The result shows that better performance is obtained for methods made of fixed feature sets-based models than randomly selected feature set-based models. The highest accuracy is obtained for the proposed model-based random forest for both test sets.

Index Terms—Fixed Feature Set, Randomly Selected Feature Set, Base Classifier, Bagging, Boosting, Random Forest

I. Introduction

Mushroom is the striking umbrella-shaped fruiting body of a particular fungi that has become so popular presently because of having numerous significant nutrition like niacin, riboflavin, selenium, potassium, and vitamin D which are precluding of hypertension, Alzheimer, Parkinson, and high risk of stroke[1]. Mushroom is a natural agent that helps to promote the environment of the world. It also helps in the recovery of contaminated damaged habitats, acts as a natural pesticide and also supplies sustainable fuel Econol[2]. Furthermore, Mushroom production is a lucrative and profitable cottage industry and this industry is providing mass employment in many developing countries. Like other countries across the world, mushroom consumption in Bangladesh is increasing day-by-day. Bangladesh is one of the most apposite countries for mushroom farming for its high market price comparing with other agro-economic crops and auspicious climate with low production cost[3]. The types of mushroom are two: edible and poisonous. In Bangladesh, about 20 species of mushroom grow wild, 5-6 are poisonous among them[4].

According to [5], 45000 species of mushrooms are identified worldwide; but the number of species of edible mushrooms is only 2000. Unexpectedly, identifying the edibility of mushroom manually is a too difficult task. Because maximum poisonous mushrooms look like edible mushroom owing to color and shape[6]. So, automation is very important in this field to reduce time and labor. There are many classification approaches exist in machine learning. For classifying mushroom, a narrow range of studies has been done using classification. [5,7-11] are some of them where Decision Tree (C4.5), SVM (Support Vector Machine), ANN, ANFIS Naïve Bayes, Bayes Net, ZeroR, and RIDOR single classifier algorithms have been used to classify mushroom. But the performance of single classifier algorithms is poorer than ensemble methods. Ensemble approaches provide a better result than single classifiers on account of merging the predictions of base models.

In this paper we have attempted to classify the mushrooms using ensemble methods: bagging, boosting and random forest. Naïve Bayes and dissimilarity measure are used for bagging, AdaBoost is for boosting and decision tree for random forests. We have used randomly selected feature sets as well as fixed feature sets in each base classifier of ensemble methods to classify mushroom using two test sets. The aim of this research is finding out the best ensemble method for identifying edibility of mushroom with the highest accuracy and lowest error rate.

II. Related Works

Recently many studies have been introduced for mushroom classification. Lavanya et al. [9] used a different kind of classification algorithms to identify whether the mushroom is edible or not. Those algorithms are evaluated using accuracy, mean absolute error and kappa statistic. This technique is called the WEKA (Waikato Environment for Knowledge Analysis). Bayes Net, Naïve Bayes, and ZeroR are used for classification. But the classifier's accuracy rate is low when the dataset is small and their performance increase with the increasing

data set. When 70% of the data have used, the accuracy for Bayes Net, Naïve Bayes, and ZeroR are 97.22%, 96.81%, and 64.25% respectively. Bayes Net has the lowest Mean Absolute error, 0.0289%, and ZeroR has the highest Mean absolute error, 0.4594%. Kappa Statistic rate of Bayes Net is also the best of these three classifiers. The conclusion is Bayes Net has the best result in this scenario and ZeroR has the worst performance.

In paper [12], Mushroom Classification is done using a different kind of features of mushroom such as *gill's type* or *color*, *shape* or *size*, *color of the cap*, *population*, *odor*. Here Principal Component Analysis (PCA) is used to identify the mushroom type and gives the highest accuracy to differentiate between poisonous and edible mushrooms by applying Decision Tree (DT) algorithm. J48 is used to produce a decision tree. PCA is applied to the decision tree and for ranking the features. The dataset which is used here has 22 attributes, 3916 poisonous mushrooms, and 4208 edible mushrooms. After applying PCA the highest-ranking attribute is an *odor*, that means among those 22 attributes the contribution of *odor* is highest to classify the mushroom.

Agung Wibowo et al. [8] compared the performance among three data mining algorithms: C4.5 based decision Tree, Naïve Bayes, and SVM (Support Vector Machines). For performing the experiment, data set is taken from *Audubon Society Field Guide to North American Mushrooms*, available in the UCI machine learning repository[13] which includes *Agaricus* and *Lepiota* families of mushroom. Both C4.5 and SVM have better accuracy than Naïve Bayes. Between C4.5 and SVM, C4.5 is faster than SVM by 0.02 seconds. Therefore, C4.5 is considered as the best among these three algorithms. In addition, C4.5 discard 5 from 22 attributes and classify based on these five attributes which are the *odor*, *spore-print-color*, *gill-size*, *gill-spacing*, and *population*[8].

In paper [14], different classifiers such as decision tree, a voted perceptron algorithm, a covering algorithm, the nearest neighbor algorithm are used for developing interactive applications of human machines. Database took from Schlimmer's compilation of the *Audubon Society's Mushroom* data. From 8124 instances, a subset with 3000 instances is used. Train dataset contains 1000 instances whereas test 1 and test 2 each contains 1000 instances, for a total of 3000 examples. Schlimmer achieved 95% accuracy on the 8124 records, whereas this method achieved 99.6% accuracy at the lowest in the J48 unpruned tree analysis on the subset of 3000 records.

III. Proposed Method

The proposed method classifies mushroom using ensemble methods and compares the result to determine which algorithm have the best accuracy. The ensemble is a made-up model for classification, formed of a combination of classifiers, where a set of weaker learners are integrated to achieve better performance than a single one. Due to noise, bias, and variance, errors in learning are created which factors are minimized by ensemble methods that are designed to improve the stability and the accuracy of machine learning methods[15]. The most familiar ensemble methods such as bagging, boosting and random forests are used for Mushroom classification. Selecting a base learner algorithm is obligatory to use these ensemble methods; Naïve Bayes and dissimilarity measure are chosen for bagging, AdaBoost is for boosting and decision tree for random forests.

The mushroom dataset has taken from *The Audubon Society Field Guide to North American Mushrooms*, contributed by Jeff Schlimmer available at UCI machine learning repository. The dataset has 8124 instances and 22 attributes with 2 possible class levels (edible and poisonous). The attributes are cap-shape, cap-surface, cap-color, bruises, odor, gill-spacing, gill-attachment, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil- type, veil-color, ring-number, ring-type, spore-print-color, population, and habitat. From UCI Machine Learning Repository, we know that six features (odor, spore-print-color, stalk-surface-below-ring, stalk-color-above-ring, habitat, cap-color) out of 22 features in five rules give a more accurate result to classifying mushroom. That's why we have made five feature sets where each feature set contains the $\log_2 N+1$ number of attributes some of which are selected from the six features and rest attributes are selected randomly. The five features sets are as follows:

- Feature set-1: Odor, Cap-surface, bruises, cap-color, gill-attachment, and cap-shape.
- Feature set-2: odor, gill-spacing, gill-size, gill-color, spore-print-color, and population.
- Feature set-3: odor, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, and stalk-color-above-ring.
- Feature set-4: cap-color, stalk-color-below-ring, ring-number, ring-type, population, and habitat.
- Feature set-5: cap-color, veil- type, veil-color, ring-type, population, and habitat.

The Features sets and randomly selected 2/3rd of total instances is used to make five models, for training in all classifications. After training, testing with rest 1/3rd instances as well as with all instances is performed.

A. Bagging

In Bagging, the different randomly selected subset of the main dataset is used for each model[16]. Each subset predicts a class, final class is based on majority voting. Naïve Bayes, as well as Dissimilarity measure, is used as a base learner algorithm for the proposed bagging method.

1. Bagging Based on Naïve Bayes Classifier

Naïve Bayes is a statistical classification model which predict the membership prospects in a class. The base theorem of Naïve Bayes is Bayes theorem[15]. The formula of Naïve Bayes is:

$$P(X|C) = P(H|X)P(X) \quad (1)$$

where,

X: data with unknown class
H: the hypothesis of *X* data, is the specific class
P(H|X): Probability of *H* hypothesis based on *X* conditions *X* (posterior probability)
P(H): the probability of *H* hypothesis (prior probability)
P(X|H): *X* probability based on the conditions of *H* hypothesis
P(X): the probability of *X*

The five models and randomly selected 2/3rd of total instances is used for training in Naïve Bayes classifier. The rest 1/3rd instances, as well as all instances, are used for two different testing. Five Naïve Bayes classifier has made five different results. After that, Bagging is applied to the results predicted by each separate model, that means the final class is predicted based on the majority. The system architecture of bagging approach using a Naïve Bayes model is shown in Fig.1.

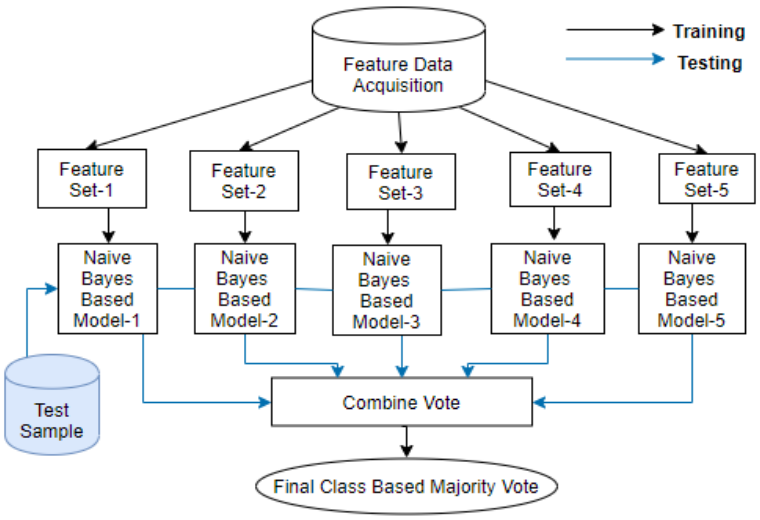


Fig.1. The system architecture of Naïve Bayes classifierbased Bagging

2. *Bagging Based on Dissimilarity Measure*

Here we have also used five models and randomly selected 2/3rd of total instances for training. Similar to above, after completing training, the rest 1/3rd instances, as well as all instances, are used for two different testing. The dissimilarity between training data and test data is calculated using the equation

$$d(i,j) = \frac{P-M}{P} \quad (2)$$

where,

P: the total number of attributes describing the objects
M: number of matches

From the dissimilarity matrix, the class with the lowest dissimilar value is predicted by each separate model. Five models predict five classes. Then, the final class is selected based on the majority. The system architecture of bagging approach using dissimilarity measure is shown in Fig.2.

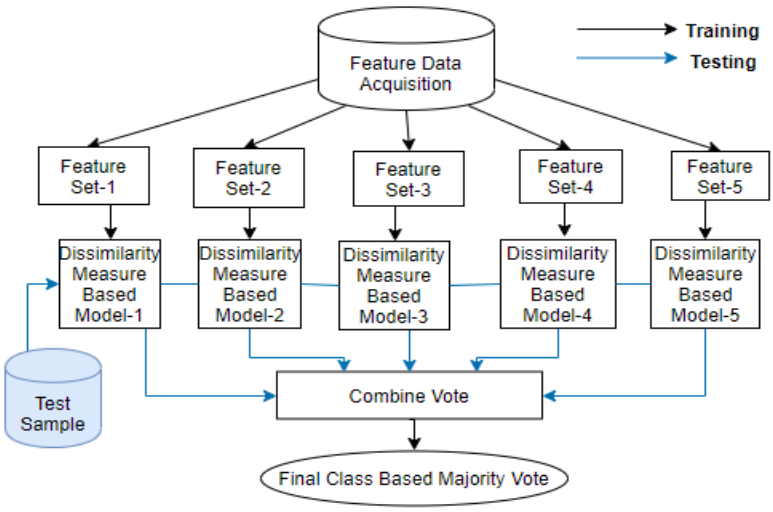


Fig.2. The system architecture of dissimilarity measure-based Bagging

B. Boosting

Unlike bagging, where each classifier is assigned an equal vote, boosting assigns a weight to each classifier's vote, based on how well the classifier performed[17]. In the candidate method, AdaBoost, a popular boosting algorithm is used for boosting.

1. AdaBoost Classifier

AdaBoost is a short form of "Adaptive Boosting", a machine learning meta-algorithm. The classifier is adapted to the sense that the next weak learners are altered by the previous classifier's misclassification. AdaBoost is sensorial to noisy data, an outlier, and also good to overfitting. We have developed the AdaBoost method with the help of the following algorithm. This algorithm creates the five weak models by itself, which finally combines to create a strong classifier. The algorithm[18] is:

Input: training set T
Output: the final classifier G(x).

a. Initialize weights of training examples:

$$D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1n}),\ w_{1i} = \frac{1}{n},\ i = 1,2, \dots, n$$

b. For m= 1, 2, …, M (where M is the number of weak classifiers)

- Fit a classifier $G_m(x)G_m(x)$ to the training data using weights w_i
- Compute misclassification error of $G_m(x)$:

$$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^{n} w_{mi} I(G_m(x_i) \neq y_i) \quad (3)$$

- Compute the weight α_m for this classifier $G_m(x)$

$$\alpha_m = \frac{1}{2}\ln 1 - \frac{1-e_m}{e_m} \quad (4)$$

- Update weights of training examples:

$$D_{m+1} = \left(w_{m+1,1,\dots\dots}, w_{m+1,i,\dots\dots}, w_{m+1,n}\right) \quad (5)$$

where,

$$w_{m+1,i} = \frac{w_{m,i}}{Z_m} exp\left(\alpha_m y_i G_m(X_i)\right) \quad (6)$$

is a regularization term and renormalize to w_i to sum to 1.

c. The final classifier $G(x)$ is a weighted sum of on each iterations M' α value and classifier output.

$$G(x) = sign\left(f(x)\right) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right) \quad (7)$$

α_m stands for the weight of the m-th classifier according to Equation (2), $\alpha_m \geq 0$ when $e_m \leq 1/2$. In addition, $\alpha_m\alpha_m$ increase with the decrease of e_m. Therefore, the classifiers with lower classification error have higher weights in the final classifier. The system architecture of the boosting method using AdaBoost is shown in Fig.3.

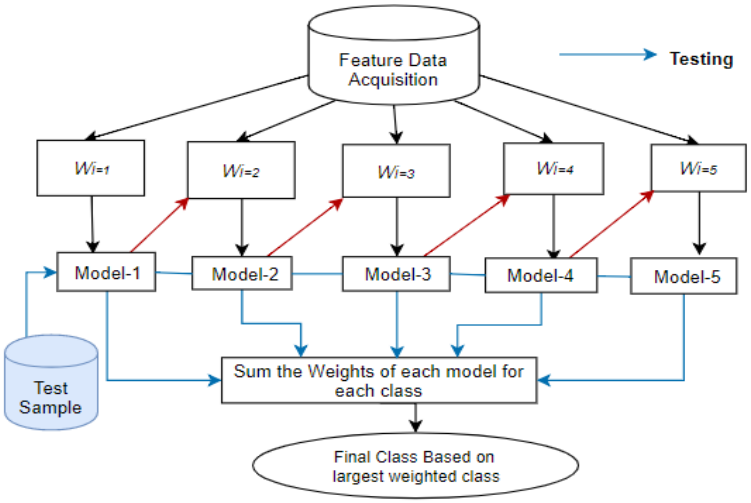


Fig.3. The system architecture of AdaBoost

C. Random Forest

Like bagging, A various random subset of the main dataset is used for each model. The decision tree is used as a base learner algorithm in this ensemble method. Among various algorithms such as Iterative Dichotomiser (ID3), Classification And Regression Trees (CART), C4.5, Chi-squared Automatic Interaction Detector (CHAID), MARS, Conditional Inference Trees algorithms, CART is used as decision tree algorithm. Due to the selection, interpretation, imagination, and simplicity of CART, it requires relatively little effort in preparation of user data for the decision trees. On the contrary, these trees can handle numerical and distinct information. It can handle multiple output problem. This algorithm does not affect the performance of trees in the nonlinear relationship between parameters. Additionally, Decision trees implicitly perform variable screening or feature selection[19]. The system architecture of Random Forest using CART based decision tree is shown in Fig.4.

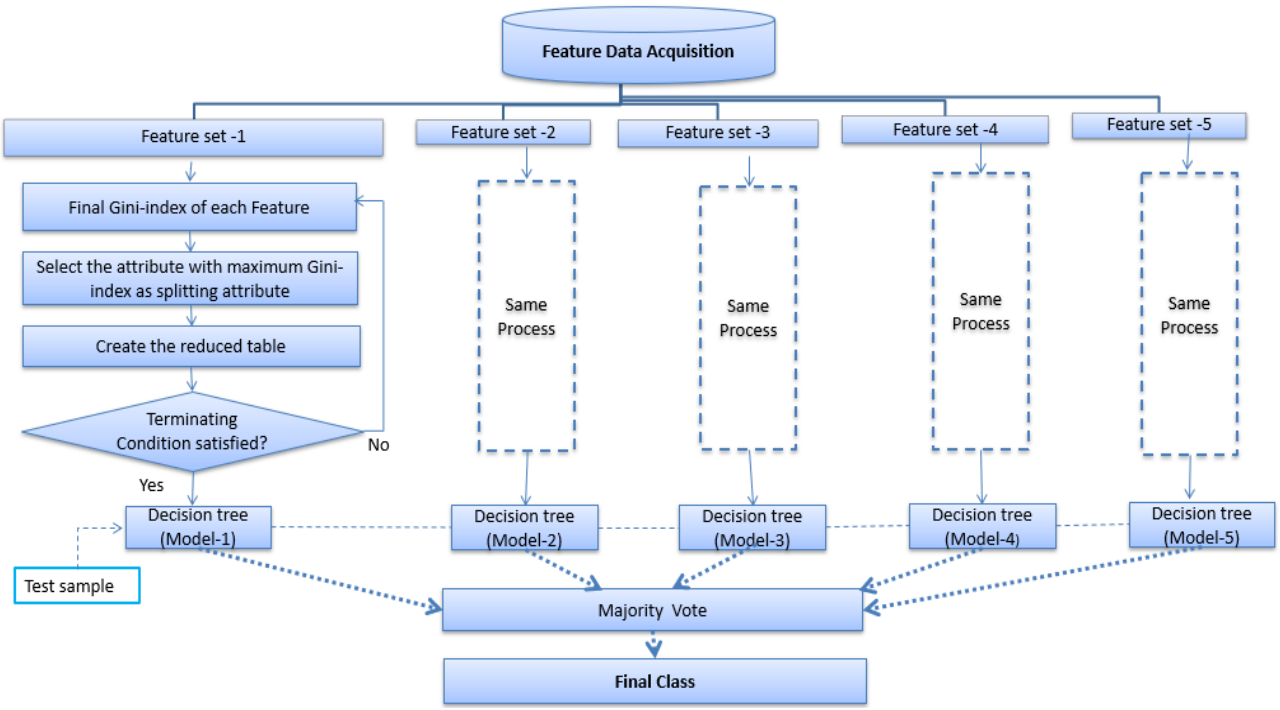


Fig.4. The system architecture of Random Forest

1. *CART Algorithm*

The CART algorithm creates a binary decision tree. Gini index is used in the CART algorithm for measuring the training tuples set, the impurity of data or data partition[17]. The algorithm of CART used in the proposed random forest is as follows[17]:

I. Establish Classification Attribute in the dataset
II. Computing classification Gini Index.

$$Gini_{Index(D)} = \sum_{i=1}^{m} p_i^2 \quad (8)$$

III. For each attribute in the dataset with the subset, calculate the Gini Index using the classification attribute.
IV. Select Attribute with subset with the highest reduction in impurity (or, minimum Gini index) to be the next Node with the subset in the tree (starting from the Root node).
V. Remove Node Attribute,
VI. Create reduced dataset R_s.
VII. Repeat steps III to VI till there are no samples left, or the same classification value remains for all rows in the reduced dataset (a branch with the entropy of 0), or all attributes have been used.

IV. Experimental Results

For the experiment, we have used the *Audubon Society Field Guide to North American Mushrooms dataset*[13]. Data are split into two sets: training and test sets. Two-third data of the dataset i.e. total 5416 instances are used for training. Testing is performed in two ways: using rest one-third of the dataset (which is not used for training), and using all the data of the dataset. The attributes are fixed for each model. Table.1 shows the accuracy of five different models as well as the result of their bagging for two different base learner algorithms. We have used two base learner algorithms for bagging: Naïve Bayes and Dissimilarity measure. Table 1 shows, as expected, bagging gives comparatively better accuracy than individual models. We also see bagging accuracy of dissimilarity measure is better than Naïve Bayes. But dissimilarity measure-based bagging takes a longer time to show the result. It has taken almost 7 hours for testing using all dataset.

Table 1. Accuracy (in %) for Bagging method

	Dissimilarity Measure		**Naïve Bayes**	
	1/3 Test	**All Test**	**1/3 Test**	**All Test**
Model-1	99.41	99.49	81.39	80.03
Model-2	100	100	83.31	82.62
Model-3	99.52	99.46	87.26	86.61
Model-4	98.01	99.08	74.63	74.74
Model-5	96.79	97.05	61.34	61.94
Bagging result	99.93	99.93	88.18	87.35

In Table 2, the accuracy of the individual weak model and their boosting method is shown. We have used AdaBoost as a boosting method. Boosting method integrates the predictions from the weak learner to produce a strong learner which increases the prophecy power of the method[20]. We have not used a fixed feature set in AdaBoost because in this method weak models are

always created by itself. That means, instead of using our predefined models, it used its own developed 5 models.

Table 2. Accuracy (in %) for Boosting method

	1/3 Test	All Test
Weak Learner, Model-1	98.2	98
Weak Learner, Model-2	98.2	98
Weak Learner, Model-3	98.8	98.7
Weak Learner, Model-4	99.0	98.8
Weak Learner, Model-5	99.9	99.7
Boosting Result	99.9	99.7

Table 3 shows the accuracy of five different decision tree models as well as the result of Random forest. In the random forest method, we used the same fixed feature set as bagging.

Table 3. Accuracy (in %) for Random Forest

	1/3 Test	All Test
Decision Tree Model-1	99.41	99.53
Decision Tree Model-2	100	100
Decision Tree Model-3	99.52	99.46
Decision Tree Model-4	98.15	98.08
Decision Tree Model-5	97.98	97.05
Random Forest Result	99.93	99.93

Table 4 shows the performance comparison for various ensemble methods (Naïve Bayes based bagging, Dissimilarity based bagging, AdaBoost, and Random Forest). The result shows that the highest accuracy is 99.93%, which is found for both Random Forest and Dissimilarity-measure based bagging, for testing using both 1/3rd data and all data of the dataset.

Table 4. Comparison of the Ensemble methods for Fixed feature set

		1/3 Test	All Test
Bagging	**Naïve Bayes Based**	88.18%	87.35%
	Dissimilarity measure Based	99.93%	99.93%
AdaBoost		99.90%	99.70%
Random Forest		99.93%	99.93%

We have also measured accuracy for all ensemble methods (except Boosting) using five models where features for each model is selected randomly instead of fixed feature set, discussed above. Here we have also used the $\log_2 N+1$ feature that means 6 features are selected among 22 features. The only difference is that the features are selected randomly. The results of ensemble methods for randomly selected features are shown in Table 5.

Table 5. Comparison of the Ensemble methods for the Randomly selected feature set

	Dissimilarity Measure based Bagging		Naïve Bayes based Bagging		Random Forest	
	1/3 Test	All Test	1/3 Test	All Test	1/3 Test	All Test
Model-1	97.97	95.18	71.09	83.09	98.97	96.31
Model-2	97.45	96.5	80.76	64.5	96.42	83.31
Model-3	93.39	95.57	78.07	50.25	95.24	99.51
Model-4	98.23	89.66	86.63	86.02	93.87	93.75
Model-5	99.41	98.72	66.14	82.23	95.86	97.54
Ensemble Result	**99.52**	**98.52**	**86.08**	**83.51**	**99.15**	**99.26**

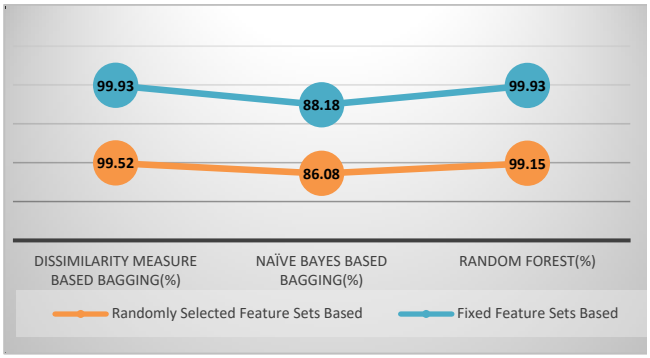
Table 6 shows the comparison between the randomly selected feature set and fixed feature set based ensemble methods for two different types of the test set. The table shows that the accuracy of the ensemble methods is higher for using fixed feature sets than using randomly selected feature sets. The highest accuracy for randomly selected feature sets-based method is 99.90% for AdaBoost (testing using 1/3 data) where the highest accuracy for fixed feature sets-based ensemble method is 99.93% for dissimilarity measure-based bagging as well as for random forest (using both test sets). But dissimilarity measure-based bagging takes more time than random forest. That's why we conclude that the best method among all ensemble methods for our fixed feature-based models is the random forest.

Table 6. Comparison of the Ensemble methods for Fixed feature set and Randomly selected feature set

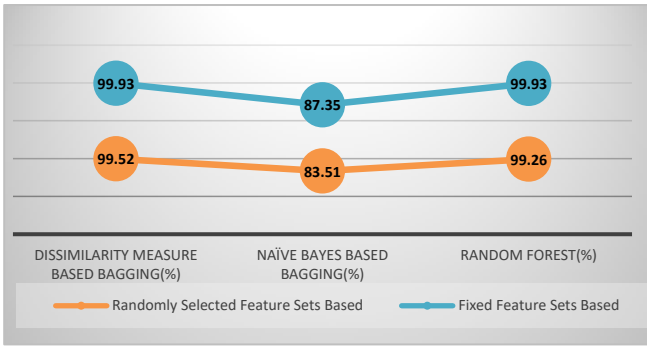
		For Randomly made of models		For Fixed models	
		1/3 test	all test	1/3 test	All test
Bagging	**Naïve Bayes Based**	86.08	83.51	88.18	87.35
	Dissimilarity Measure Based	99.52	98.52	99.93	99.93
AdaBoost		99.9	99.7	-	-
Random Forest		99.15	99.26	99.93	99.93

Fig.5 shows the Comparison of ensemble methods for randomly made of models and fixed models for testing using 1/3 data of the total dataset and testing using all data of the total dataset. This figure depicts that accuracy of fixed feature set based methods is higher than randomly created model-based methods for both test sets.

The performance of proposed methods and other existing works is compared based on the performance measures, Accuracy and Error Rate on the mushroom dataset. We have compared the best result of [9-11] with our best result. The best accuracy and lowest error rate



(a)



(b)

Fig.5. Comparison of ensemble methods for randomly made of models and fixed models using (a) 1/3 test data of dataset (b) all data as the test set

for all methods used in these papers and our proposed method are shown in Table 7. In [9], they have used three data mining algorithms Bayes Net, Naïve Bayes, ZeroR and measured performance of the algorithms with the respective training dataset size and their highest accuracy is 97.22% for Bayes Net when 70% tuples are used for the training set and 30% are for testing. Similarly, the highest accuracy for Naïve Bayes, ZeroR is 96.81%, 64.25% for same data size. In [10], they have analyzed the performance of ID3, CART, Hoeffding tree on the mushroom dataset and the accuracy of ID3 is 69%, CART is 90% and Hoeffding tree is 100%. In paper [11], they have used three classifiers: ANN, ANFIS, Naïve Bayes. It shows that the highest accuracy of [9] is 97.22% for Bayes Net to test using 30% data of mushroom dataset among their three algorithms Bayes Net, Naïve Bayes, and ZeroR. The highest accuracy of [10] is 100% among the three algorithms ID3, CART, HOFFDING tree for testing all data. The highest accuracy of [11] is 99.88% for ANFIS.

Table 7 indicates that the accuracy of the candidate system is higher than all methods except HOFFDING tree of [10]. The performance comparison of highest accuracy, as well as the lowest error rate of the proposed method with the previously described existing methods, is shown in Fig.6.

Table 7. Comparison with some Existing works

	Accuracy	Error Rate	Methods	Testing Size
Proposed	99.93%	0.07%	Random forest	1/3 testing, all testing
S. Beniwal[9]	97.22%	2.78%	Bayes Net	30% testing
	96.81%	3.18%	Naïve Bayes	30% testing
	64.25%	35.75 %	ZeroR	30% testing
B. Lavanya[10]	69%	31%	ID3	all testing
	90%	10%	CART	all testing
	100%	0%	HOEFFDING TREE	all testing
S.K Verma[11]	96.82%	3.19%	ANN	30% testing
	99.88%	0.12%	ANFIS	20% testing
	96.82%	3.19%	Naïve Bayes	30% testing

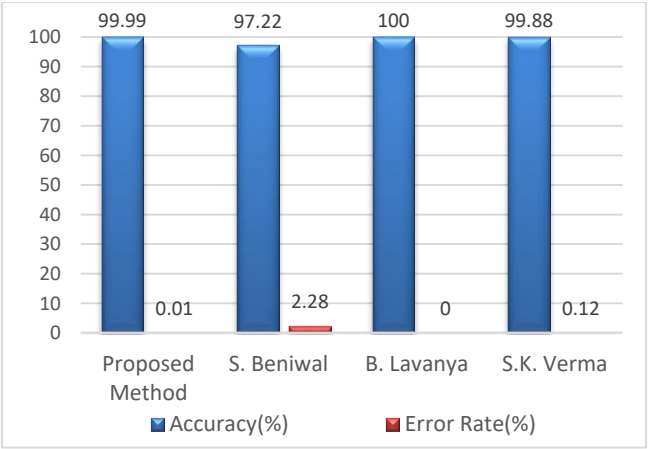


Fig.6. System Architecture of Random Forest

Fig.6 depicts that the proposed method provides better results with the highest Accuracy and least Error rate except [10] on mushrooms dataset.

V. Conclusion

The aim of the proposed method is to assess the risk of human life for the edibility of mushroom. We have used three ensemble methods: bagging (Naïve Bayes based Bagging, Dissimilarity based Bagging), Boosting (AdaBoost), and Random Forest. The proposed fixed feature sets-based methods show higher accuracy than randomly selected feature set based methods. The highest accuracy is found for Random forest and dissimilarity measure-based bagging but dissimilarity measure-based bagging takes more time than the Random Forest. The experimental result shows that the proposed methods are robust than many existing methods.

References

[1] M. E. Valverde, T. Hernández-pérez, and O. Paredeslópez, "Review Article Edible Mushrooms : Improving Human Health and Promoting Edible Mushrooms : Improving Human Health and Promoting," no. January, 2015.

[2] Mercola.com. (2019). The Mushroom Advantage: Benefits and Uses of Mushroom. [online] Available at: https://www.mercola.com/infographics/mushrooms.htm [Accessed 23 Jan. 2019].
[3] Md Mahabub Alam, Md Waliul Bari, 'Investment in Mushroom Cultivation at Savar Upazila: A Prospective Sector for Bangladesh', ASA University Review, Vol. 4 No. 2, July–December, 2010
[4] 'Mushroom',http://en.banglapedia.org/index.php?title=Mushroom,Accessed Date: 01 August, 2018
[5] D. R. Chowdhury and S. Ojha, "An Empirical Study on Mushroom Disease Diagnosis: A Data Mining Approach," 2017.
[6] C. FM, "Amanita phalloides in Victoria," p. 849–850., 1993.
[7] A. R. Khan, S. S. Nisha, and M. M. Sathik, "Clustering Techniques For Mushroom Dataset," no. June, pp. 1121–1125, 2018.
[8] Agung Wibowo, Yuri Rahayu, Andi Riyanto, and Taufik Hidayatulloh. "Classification algorithm for edible mushroom identification." In Information and Communications Technology (ICOIACT), 2018 International Conference on, pp. 250-253. IEEE, 2018.
[9] Beniwal, Sunita and Bishan Das. "Mushroom Classification Using Data Mining Techniques." International Journal of Pharma and Bio Sciences, Vol 6, issue 1, pp. 1170-1176, 2015.
[10] Lavanya, B. "Performance Analysis of Decision Tree Algorithms on Mushroom Dataset." International Journal for Research in Applied Science and Engineering Technology. Vol. 5, issue XI, pp. 183-191, 2017.
[11] S.K. Verma, M. Dutta, "Mushroom Classification Using ANN & ANFIS Algorithm", IOSR Journal of Engineering (IOSRJEN), Vol. 08, Issue 01, PP 94-100, January. 2018.
[12] Ismail, Shuhaida, Amy Rosshaida Zainal, and Aida Mustapha. "Behavioural features for mushroom classification." In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 412-415. IEEE, 2018.
[13] "UCI Machine Learning Repository: Mushroom Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Mushroom.
[14] Eusebi, Clara, Cosmin Gliga, Deepa John, and Andre Maisonave. "Data Mining on Mushroom Database." Journal of CSIS, Pace University (2008): 1-9.
[15] "What is the difference between Bagging and Boosting?" Quantdare,. [Online]. Available: https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/.
[16] "What is the difference between Bagging, Random Forest and Boosted Tree? Which should I use?" [Online]. Available: https://support.bigml.com/hc/en-us/articles/206739539-What-is-the-difference-between-Bagging-Random-Decision-Forest-and-Boosted-Tree-Which-one-should-I-use-
[17] J. Han, M. Kamber, J. Pei. "Data Mining Concepts & Techniques.", Morgan Kaufmann Publishers, ELSEVIER, 3rd edition.
[18] "AdaBoost,". [Online]. Available: https://infinitescript.com/2016/09/adaboost/ .
[19] Anon, (2019). [online] Available at: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052. [Accessed 10 Jan. 2019].
[20] "Ensemble learning, beginners guide, machine learning, data science, analytics," [Online]. Available: https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/.

Authors' Profiles

Nusrat Jahan Pinky is a student at the Computer Science and Engineering Discipline, Khulna University, Bangladesh. She was born on 13th July 1997. Her research interests include Machine learning, Data Mining, and Digital Image Processing.

S.M. Mohidul Islam is an Associate Professor at the Computer Science and Engineering Discipline, Khulna University, Bangladesh. He received his B.Sc. Engg. and M.Sc. Engg. degree from Khulna University. His research interests include Machine learning, Data Mining, Pattern Recognition, and Digital Image Processing.

Rafia Sharmin Alice is a student at the Computer Science and Engineering Discipline, Khulna University, Bangladesh. She received her B.Sc. Engg. degree from Khulna University. Her research interests include Data Mining, Pattern Recognition, and Computer Vision.

How to cite this paper: Nusrat Jahan Pinky, S.M. Mohidul Islam, Rafia Sharmin Alice, " Edibility Detection of Mushroom Using Ensemble Methods", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.4, pp. 55-62, 2019.DOI: 10.5815/ijigsp.2019.04.05

