Parametric Feature Selection for an Enhanced Random Linear Oracle Ensemble Method

B. P. Ooi[1], N. Abdul Rahim[1], A. Zakaria[1], M. J. Masnan[2], S. A. Abdul Shukor[1] and Paulraj M. P.[3]
[1]*School of Mechatronic Engineering, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia.*
[2]*Institute of Engineering Mathematics, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia.*
[3]*Sri Ramakrishna Institute of Technology, Perur Chettipalayam, Pachapalayam, Coimbatore – 641010, Tamilnadu, India.*
bpooi0702@gmail.com

***Abstract*—Random Linear Oracle (RLO) utilized classifier fusion-selection approach by replacing each classifier with two mini-ensembles separated by an oracle. This research investigates the effect of t-test feature selection toward classification performance of RLO ensemble method. Naïve Bayes (NB) classifier has been chosen as the base classifier due to its elegant simplicity and computationally inexpensive. Experiments were carried out using 30 data sets from UCI Machine Learning Repository. The results showed that RLO ensemble could greatly improve the ability of NB classifier in dealing with more data with different properties. Moreover, RLO ensemble receives benefits from feature selection algorithm, with a properly selected number of features from t-test, the performance of ensemble can be improved.**

***Index Terms*—Ensemble; Feature Selection; Naïve Bayes; Pattern Recognition; Random Linear Oracle.**

I. INTRODUCTION

Pattern recognition is a branch of machine learning where upon receives an input data, it will associate the data to a predefined target class, in short, assign a label to a data [1].

The objective of pattern recognition can be done by means of a classifier, which is any mathematical function that can assign a label to the object [2]. However, the performance of a single classifier is very limited and does not meet public expectation, so classifier ensemble method has been introduced to compromise the weakness [3].

An ensemble method is a combination of two or more classifiers in one classification process. Different classifiers will be used to train on same feature data, or similar classifiers will be trained on different feature subsets to allow more diversity in the ensemble. So, when given an input, each classifier in the ensemble will provide their respective output, and a combiner will be used to combine all the output into a single label. This approach can help in increasing diversity and often lead to a better classification performance [4].

This research studies the performance of RLO ensemble using NB as its base classifier. As well as how RLO ensemble reacts to differently sized feature from t-test feature selection algorithm.

II. BACKGROUND REVIEW

A. Introduction

An ensemble model can be explained using the four layers presented in Figure 1. The very first layer describes the data level. This layer explains how data being divided into training set and testing set. One rule in testing an algorithm's accuracy is that the testing set must not previously "seen" by the learning algorithm to avoid peeking. Some commonly used data manipulation methods are divide-and-conquer, cross-validation, and bootstrap method [2].

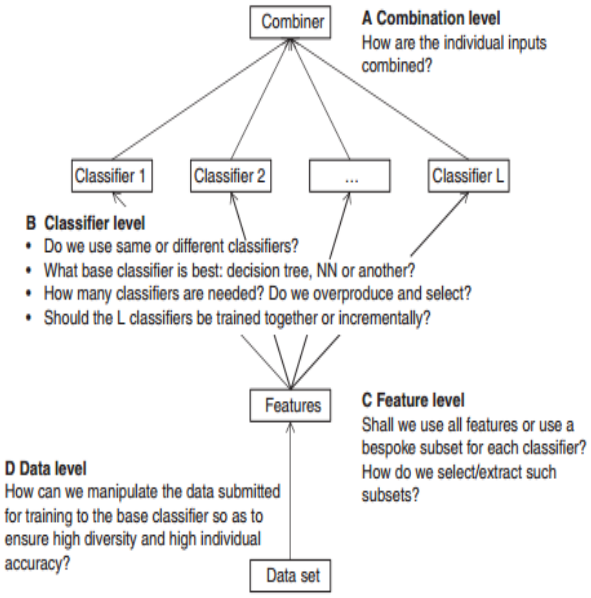

Figure 1: Four levels of ensemble model [2]

Feature level comes after the data level, this is where features are evaluated and selected through feature selection algorithm. By selecting the key features and eliminating less important one, it can greatly speed up the training process, and possibly improve the classification performance.

Next is on the classifier layer to determine the type of base classifier used, the number of the base classifier, and the classifier training procedure. There are two types of classifier combination, homogenous where all base classifiers are the same but trained on different data subsets to allow more variety, or heterogeneous where different base classifiers will be used to undergo the classification process.

Combination layer is the last process in ensemble learning. It describes how all output of classifiers being combined to form a single label. Some widely-used combination approaches are majority voting, Naïve Bayes combiner, and multinomial method.

B. Naïve Bayes Classifier

Naïve Bayes (NB) classifier is a simple probabilistic

classifier where it assumes that every feature in the data is conditional independent from each other. It is chosen as the base classifier of this research because of its inexpensive computational property can help to reduce the processing time [5].

Assuming N data samples and C number of classes from one experiment where $\tilde{x} = \{x_1, \dots, x_n\}$ is the feature vector for one sample and $\tilde{\omega} = \{\omega_1, \dots, \omega_c\}$ is the class vector that are available for label.

NB classifier can be formulated by the equation:

$$P(\omega_i|\tilde{x}) = P(\omega_i) \prod_{i=1}^{n} p(\tilde{x}|\omega_i) \quad (1)$$

where: $P(\omega_i|\tilde{x})$ = Posterior probability for class $i = \{1, \dots, c\}$.
$P(\omega_i)$ = Prior probability for class $i = \{1, \dots, c\}$.
$p(\tilde{x}|\omega_i)$ = Class-conditional probability for class $i = \{1, \dots, c\}$.

From Equation (1), posterior probability refers to the probability of an object $\bar{x}$ belongs to class ω_i, thus higher posterior probability indicates the likelihood of the object to be from class ω_i.

C. Random Linear Oracle

Random Linear Oracle (RLO) introduced by Kuncheva and Rodríguez (2007) is a unique ensemble method that combines both classifier fusion and selection approaches, by replacing each classifier with two mini-ensembles along with a random oracle chosen between them [6-10].

Figure 2 illustrates the application of RLO ensemble on the two-class problem. Through separating the entire feature space using an oracle, each feature subset will be used to train a classifier, allows the classifier to an expert in a specific subset instead of the whole feature space. Furthermore, the same feature space will be used to train several RLO ensemble, each with a different feature split.

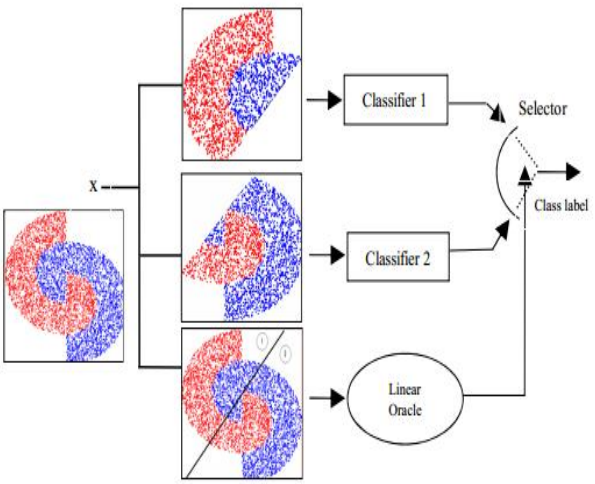

Figure 2: RLO method applied to the two-class problem [11]

During classification process, the location of incoming data will first be determined by the oracle, and correspond classifier with the subset expertise will be called to carry out the classification. Results from all classifiers will be combined with classifier fusion approach, i.e. simple majority voting.

The pseudo code for RLO training and classification algorithms is shown in Figure 3.

Random Linear Oracle Ensemble (RLO)

Training

```
L      = number of ensemble size
for each l ϵ L
        Splits feature space into D_l+ and D_l- with random hyperplane (oracle).
        Train NB classifiers with each subset, NB_l+ and NB_l-.
end
return oracle, NB_l+, NB_l-
```

Operation

```
x      = new input object
for each l ϵ L
        Apply lth oracle to determine region of x.
        Use classifier correspond to region to classify x.
end
Count number of votes for each class.
return class label with max(votes)
```

Figure 3: Pseudo code for RLO ensemble training and operation phases

D. Feature Selection

Feature selection is a process that chooses a feature subset that well represents the whole feature space [12, 13]. The reason for feature selection is to reduce the dimensions of data so that to ease the classification process. Also, it is believed that with a properly chosen feature subset, this process can improve the classification performance [14]. There are two main approaches in feature selection, namely filter approach and wrapper approach.

Feature selection by filter approach selects the feature subset by analyzing data properties without the needs of training and testing phase being conducted. Student's t-test from hypothesis testing will be used in this research to examine the properties of feature data, and rank it from the most interesting feature until the least significant feature.

However, t-test is only used for ranking of features, the selection will be made by choosing the percentage offset from the overall feature. This research allocated 10%, 20% to 90% of the overall features size for each training and testing, respectively. The selection starts with the most interesting features down until the least significant ones.

a. T-test

The T-test is a parametric hypothesis testing method proposed by Gosset (1908) under the pseudonym "Student" [15-17]. It is used to compare two population means to determine whether both populations are significantly different from each other assuming populations are normally distributed. The main purpose is to look for features that are unequal as distinctive features.

The test statistic in t-test is denoted by the variable *t*, and is calculated with the formula as below:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{{s_x}^2}{n} + \frac{{s_y}^2}{m}}} \quad (2)$$

where: t = Test statistic
$\bar{x}$ = Sample mean of first class data
$\bar{y}$ = Sample mean of second class data
n = Number of samples in first class data
m = Number of samples in second class data
${s_x}^2$ = Sample variance of first class data
${s_y}^2$ = Sample variance of second class data

Once the test statistic is obtained, the *p*-value can be determined using t-distribution table [18]. Since Equation (2) is available for two-population test only, multiple pair-wise comparisons of every class data in each feature need to be calculated. Therefore, to compare each and every class data from many different classes in a feature, multiple comparisons of t-test is carried out to determine the *p*-value [17, 19]. Thus, the ranking of features could be done by sorting all the mean *p*-values of every feature.

E. Test of Hypothesis

To test the algorithms' performance, Mann-Whitney U-test has been introduced for this purpose. The objective is to check the performance of a given algorithm has a significant difference from a fixed control class.

a. Mann-Whitney U-test

Mann-Whitney test is also known as Wilcoxon rank-sum test. It is a non-parametric test that is for distribution free data, and assuming samples from both groups are independent of each other [20]. U-test will be used to analyze the significant difference in the median between two algorithms on the same data set. In this research, the classification results of NB classifier will be used as the control class.

The procedures of U-test involved assigning a rank to each data in the testing set regardless of their group in ascending order, beginning with rank 1 for smallest value, and the test statistic *U* is calculated using Equation (3)

$$U = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - T \quad (3)$$

where: n_1 = Number of observations in algorithm 1
n_2 = Number of observations in algorithm 2
T = Sum of ranks assigned to observations

The test statistic is then used to obtain the *p*-value from U-distribution table [18]. Since the classification results of NB classifier will be used as the control class, so if the *p*-value obtained from an algorithm tested against NB classifier is less than the α value 0.05, the algorithm is significantly different from NB classifier. Hence, if the median of the algorithm's accuracy is greater than of NB classifier, it can be concluded that the algorithm performs significantly better than NB classifier for that data set.

III. EXPERIMENT

This research begins with reading a data set and randomly split into a 6:4 ratio, whereby 60% of the data will be used for training, and remaining 40% will be used for testing. The training data will directly present to NB classifier and RLO ensemble for training purpose, while to t-test for properties evaluation. Rank-sorted features will be selected based on the desired offset percentage and passed to another RLO ensemble for training.

The testing process will begin after training phase, the results from classification will be compared to the desired target class, and accuracy will be calculated. A total of five iterations will be conducted, each with a different train-test split. Five accuracy values from one algorithm will be used for U-test comparison, whereby setting the results of NB classifier as the control class, to determine the performance of an algorithm.

Table 1 shows the properties of each dataset obtained from UCI repository [21], as well as the components of each data set such as the number of classes, number of objects in data set, number of features for one object. The fifth column in the table indicates the balance of data in each class. 'yes' means that each class in the data set has the same number of objects, '~yes' means it is almost balance, and 'no' means each class in the data set has an unequal number of objects. Finally, D/C column states the property of values in the data set whether it is discrete or continuous, where 'D' stands for discrete and 'C' stands for continuous.

Table 1
Properties of Data Sets

No.	Data Set	Classes	Objects	Features	Balance	D/C
1	Abalone	29	4177	8	no	C
2	Balance	3	625	4	~yes	D
3	Blood	2	748	4	no	D
4	Car	4	1728	6	no	D
5	Ecoli	8	336	7	no	C
6	Glass	7	214	9	no	C
7	Ionosphere	2	351	34	no	C
8	Iris	3	150	4	yes	C
9	Leaf	36	340	14	no	C
10	Lenses	3	24	4	no	D
11	Magic	2	19020	10	no	C
12	mfeat-fac	10	2000	216	yes	D
13	mfeat-fou	10	2000	76	yes	C
14	mfeat-kar	10	2000	64	yes	C
15	mfeat-mor	10	2000	6	yes	C
16	mfeat-pix	10	2000	240	yes	D
17	mfeat-zer	10	2000	47	yes	C
18	page	5	5473	10	no	C
19	Pima	2	768	8	no	C
20	PokerTrain	10	25010	10	no	D
21	Segmentation	7	2310	19	yes	C
22	Spect	2	267	22	~yes	D
23	vehicle	3	846	18	no	D
24	vowel	11	528	10	yes	C
25	wfsonar-2	4	5456	2	no	C
26	wfsonar-24	4	5456	24	no	C
27	wfsonar-4	4	5456	4	no	C
28	Wine	3	178	13	no	C
29	yeast	10	1484	8	no	C
30	Zoo	7	101	6	no	D

IV. RESULTS

The acronyms of the algorithm used in this section are explained in Table 2.

Table 2
Definition of Acronyms

Acronym	Definition
NB	Naïve Bayes classifier
RLO	Random Linear Oracle ensemble
RLO-TT-*x*	RLO ensemble using t-test feature selection with *x*% selected features

A. Mean Accuracy

Table 5 describes the results for each algorithm of each data set. From the table, NB classifier recorded 64.41% accuracy, while RLO ensemble recorded 64.51% of accuracy, showing that RLO ensemble is not able to improve the classification performance of NB classifier.

The relatively weaker algorithms from the mean accuracy aspect are RLO-TT-10, RLO-TT-20, and RLO-TT-30 with only 52.27%, 58.30%, and 60.36%, respectively. Judging from this trend, an assumption can be made stating that with more number of features, RLO ensemble can have better performance.

However, the highest accuracy is achieved by RLO-TT-70 with a value of 68.06%, instead of RLO-TT-90 (66.58%). So, the previous assumption is rejected with a new assumption stating that with a properly chosen number of features, RLO ensemble can perform better than the one without feature selection.

Mean accuracy alone does not provide much information in deciding a better algorithm. Thus the results from Mann-Whitney U-test are required.

B. Win, Lose, and Tie

Table 3 summarizes the performance of all algorithms when compared to the results of NB classifier. RLO ensemble scored 10 wins out of a total of 30 data sets, means that RLO ensemble can perform significantly better than NB classifier in one third out of all data sets.

Table 3
Total Number of Win, Lose, Tie for Mann-Whitney U-Test

	Win	Tie	Lose
RLO	10	17	3
RLO-TT-10	3	7	20
RLO-TT-20	7	10	13
RLO-TT-30	9	12	9
RLO-TT-40	8	14	8
RLO-TT-50	8	15	7
RLO-TT-60	8	16	6
RLO-TT-70	11	15	4
RLO-TT-80	10	16	4
RLO-TT-90	11	14	5

To ease the comparison process, the win-lose ratio will be calculated by adding the number of ties into winning count and losing count, respectively. Then divide the winning count by losing count to obtain the win-lose ratio. Table 4 shows the calculated win-lose ratio for all algorithms. The worst performance algorithm is RLO-TT-10, this may be due to the insufficient of feature number for RLO ensemble to carry out a proper classification process.

On the other hand, the best performance algorithm is RLO-TT-70 with 1.37:1 ratio, proving that with a properly chosen number of features, RLO ensemble can perform better than the one without feature selection.

Table 4
Win-Lose Ratio of Overall Results

	Ratio
RLO	1.35
RLO-TT-10	0.37
RLO-TT-20	0.74
RLO-TT-30	1.00
RLO-TT-40	1.00
RLO-TT-50	1.05
RLO-TT-60	1.09
RLO-TT-70	1.37
RLO-TT-80	1.30
RLO-TT-90	1.32

Table 5
Accuracy for Each Algorithm in Percentage (%)

Data Set	NB	RLO	RLO-TT-10	RLO-TT-20	RLO-TT-30	RLO-TT-40	RLO-TT-50	RLO-TT-60	RLO-TT-70	RLO-TT-80	RLO-TT-90
Abalone	23.19	23.53	8.77	25.38	25.58	25.22	23.22	22.66	22.31	22.17	21.36
Balance	89.33	89.17	50.39	44.07	45.43	66.93	67.66	67.18	76.72	76.00	76.40
Blood	74.72	77.19	75.92	75.92	75.92	76.05	76.39	76.52	76.86	76.72	76.59
Car	80.82	82.96	64.93	70.93	70.78	70.78	72.12	72.78	72.72	79.03	79.03
Ecoli	41.51	41.51	33.63	44.50	66.24	69.96	69.38	69.67	74.14	65.50	67.73
Glass	48.72	48.95	31.48	49.41	52.73	51.61	55.57	55.68	57.57	55.92	54.31
Ionosphere	65.01	65.01	54.06	34.99	34.99	34.99	34.99	34.99	34.99	34.99	34.99
Iris	93.67	93.67	74.00	73.00	68.67	94.67	95.00	88.00	94.33	94.00	94.33
Leaf	66.32	57.79	38.24	44.26	52.06	57.35	63.09	63.97	65.59	67.35	64.26
Lenses	61.23	34.45	20.64	20.64	20.64	30.77	36.95	36.77	52.23	54.05	40.59
Magic	73.00	75.85	65.11	79.00	78.66	76.79	74.75	75.49	76.39	76.00	75.88
mfeat-fac	80.20	82.98	83.15	87.78	88.98	90.60	91.68	91.93	92.05	92.15	90.63
mfeat-fou	76.03	77.70	75.95	79.03	79.23	78.60	78.45	78.05	78.40	78.30	77.78
mfeat-kar	93.73	95.10	77.93	90.98	94.10	94.25	94.50	94.45	94.53	95.15	95.20
mfeat-mor	36.53	33.40	10.05	10.05	49.98	49.55	55.03	58.30	58.33	59.55	59.25
mfeat-pix	34.95	20.70	72.55	63.20	49.90	46.85	36.55	34.08	30.80	27.13	29.98
mfeat-zer	72.90	74.58	54.48	59.53	65.93	68.88	69.58	70.98	72.93	73.23	74.45
page	89.67	89.45	89.28	90.73	91.24	90.55	92.36	91.01	91.69	91.34	89.82
Pima	72.95	73.02	53.90	74.06	73.86	74.71	73.80	73.73	73.08	73.47	72.82
PokerTrain	49.75	52.23	49.75	49.88	50.24	51.01	51.43	51.68	52.19	52.10	52.58
Segmentation	14.20	14.20	76.49	78.12	77.14	81.82	86.04	86.47	83.96	80.06	75.15
Spect	68.98	71.05	61.47	70.68	66.72	66.56	66.18	64.66	70.48	64.66	70.11
vehicle	60.06	71.72	55.21	62.07	61.66	66.51	67.99	67.28	68.93	68.46	70.18

Data Set	NB	RLO	RLO-TT-10	RLO-TT-20	RLO-TT-30	RLO-TT-40	RLO-TT-50	RLO-TT-60	RLO-TT-70	RLO-TT-80	RLO-TT-90
vowel	64.51	74.01	14.51	56.05	58.15	63.93	66.78	67.36	71.05	73.71	71.83
wfsonar-2	90.81	94.28	70.84	58.87	57.73	61.94	64.81	45.99	61.33	62.25	60.67
wfsonar-24	52.27	60.92	46.88	49.06	50.59	53.48	54.65	55.09	60.43	58.41	59.67
wfsonar-4	88.94	91.03	38.41	37.95	32.92	79.62	80.42	79.14	84.12	83.82	83.82
Wine	96.33	96.90	53.58	90.11	92.10	95.21	93.78	94.63	95.19	95.19	96.04
yeast	31.10	31.10	25.67	38.11	37.80	49.09	53.57	55.52	57.58	57.71	41.10
Zoo	40.76	40.76	40.76	40.76	40.76	40.76	40.76	40.76	40.76	40.76	40.76
Average	64.41	64.51	52.27	58.30	60.36	65.30	66.25	65.49	68.06	67.64	66.58

V. CONCLUSION

This research studies the effectiveness of Random Linear Oracle (RLO) ensemble method in solving different real-life classification problems, as well as the effect of applying different sized data through feature selection to this method.

The results from Section IV.A and IV.B claimed that RLO ensemble could not significantly improve the overall classification accuracy of NB classifier, but it does have an advantage of providing a better data set coverage than NB classifier. Also, the number of features submitted for RLO ensemble will have a significant impact on the performance. Without enough number of features, RLO ensemble will perform even worse than a single NB classifier. However, with a properly selected number of features, RLO ensemble can produce a better result than the one without feature selection, and in this research, 70% selected features will be the best option.

In conclusion, there is no best classifier or ensemble method per se. For further improvement, it is suggested to apply and test diverse types of random oracle in the same ensemble method. Also, the potential of RLO ensemble can be greatly improved through feature selection. Thus, another research can be undergone to extensively investigate the effect of different feature selection algorithms toward this method.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my profound gratitude and deep regards to my project supervisor, Dr. Norasmadi Bin Abdul Rahim who has guided and coordinated me at every aspect of this project especially for the enlightenment of interest in the field of pattern recognition.

Also, to Professor Dr. Paulraj Murugesa Pandiyan, thanks to his conscientious teaching in C programming and Artificial Intelligence courses, whom encouraged me in taking on this challenge which requires in-depth programming knowledge and machine learning theory.

At last, I would like to convey my warmest regards to all who supported and guide me through this project. Wholehearted thanks.

REFERENCES

[1] F. Y. Shih, *Image Processing and Pattern Recognition: Fundamentals and Techniques*. 2010.

[2] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms: Second Edition*. 2014.

[3] F. Roli, G. Giacinto, and G. Vernazza, "Methods for Designing Multiple Classifier Systems," *Mult. Classif. Syst.*, vol. 1857, pp. 78–87, 2000.

[4] T. G. Dietterich, "Ensemble Methods in Machine Learning," *Mult. Classif. Syst.*, vol. 1857, pp. 1–15, 2000.

[5] I. Rish, "An Empirical Study of the Naive Bayes Classifier," *IJCAI 2001 Work. Empir. methods Artif. Intell.*, pp. 41–46, 2001.

[6] L. Kuncheva and J. Rodriguez, "Classifier Ensembles with a Random Linear Oracle," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 500–508, Apr. 2007.

[7] J. Rodrïguez and L. Kuncheva, "Naive Bayes Ensembles with a Random Oracle," *Mult. Classif. Syst.*, pp. 450–458, 2007.

[8] C. Pardo and J. Rodríguez, "Random Oracles for Regression Ensembles," *Ensembles Mach.*, pp. 181–199, 2011.

[9] K. Li and L. Hao, "Naive Bayes Ensemble Learning based on Oracle Selection," *Control and Decision Conference, 2009. CCDC '09. Chinese*. pp. 665–670, 2009.

[10] A. Ahmad and G. Brown, "A Study of Random Linear Oracle Ensembles," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5519 LNCS, pp. 488–497, 2009.

[11] G. Armano and N. Hatami, "Random Prototype-based Oracle for Selection-fusion Ensembles," *2010 20th Int. Conf. Pattern Recognit.*, pp. 77–80, 2010.

[12] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[13] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[14] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature Selection using Forest Optimization Algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, 2016.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. 2006.

[16] D. Caprette, "'Student's' t Test (For Independent Samples)," *Rice University*, 1999. [Online]. Available: http://www.ruf.rice.edu/~bioslabs/tools/stats/ttest.html. [Accessed: 30-Mar-2016].

[17] N. Zhou and L. Wang, "A Modified T-test Feature Selection Method and Its Application on the HapMap Genotype Data," *Genomics, Proteomics Bioinforma.*, vol. 5, no. 3–4, pp. 242–249, 2007.

[18] M. N. Aishah, M. Maz Jamilah, M. A. Nor Azrita, M. N. Nor Fashihah, and S. Syafawati, *Engineering Statistics*. Kedah: Institut Matematik Kejuruteraan, Fotocopy Ent., 2016.

[19] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[20] N. Nachar, "The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution," *Tutor. Quant. Methods Psychol.*, vol. 4, no. 1, pp. 13–20, 2008.

[21] K. Bache and M. Lichman, "UCI Machine Learning Repository," *University of California Irvine School of Information*, 2013. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html