An Enhanced Random Linear Oracle Ensemble Method using Feature Selection Approach based on Naïve Bayes Classifier

Boon Pin Ooi[1], Norasmadi Abdul Rahim[1], Ammar Zakaria[1], Maz Jamilah Masnan[2], Shazmin Aniza Abdul Shukor[1]

[1]*School of Mechatronic Engineering, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia*

[2]*Institute of Engineering Mathematics, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia*

bpooi0702@gmail.com

***Abstract*— Random Linear Oracle (RLO) ensemble replaced each classifier with two mini-ensembles, allowing base classifiers to be trained using different data set, improving the variety of trained classifiers. Naïve Bayes (NB) classifier was chosen as the base classifier for this research due to its simplicity and computational inexpensive. Different feature selection algorithms are applied to RLO ensemble to investigate the effect of different sized data towards its performance. Experiments were carried out using 30 data sets from UCI repository, as well as 6 learning algorithms, namely NB classifier, RLO ensemble, RLO ensemble trained with Genetic Algorithm (GA) feature selection using accuracy of NB classifier as fitness function, RLO ensemble trained with GA feature selection using accuracy of RLO ensemble as fitness function, RLO ensemble trained with t-test feature selection, and RLO ensemble trained with Kruskal-Wallis test feature selection. The results showed that RLO ensemble could significantly improve the diversity of NB classifier in dealing with distinctively selected feature sets through its fusion-selection paradigm. Consequently, feature selection algorithms could greatly benefit RLO ensemble, with properly selected number of features from filter approach, or GA natural selection from wrapper approach, it received great classification accuracy improvement, as well as growth in diversity.**

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

I. INTRODUCTION

Pattern recognition is a branch of machine learning, which involves in receiving a number of data as input features, and associate the data to one of the predefined class, in short, assigning a class label to the data set [1].

The objective of pattern recognition can be achieved by means of a classifier, which in simple, could be explained as any mathematical functions that are able to assign a class label to an object [2]. However, the accuracy of a single base classifier does not meet the public expectation, and that is why classifier ensemble methods are introduced [3].

Ensemble method suggested to combine more than one classifiers that trained under the same or different sets of training subject. Given a set of input features, each classifier will produce their respective output, and some combination approaches are applied to determine the label of the object. This approach encourages extra diversity in the ensemble while often results in better classification performance [4].

Accuracy and diversity are the two terms that arose when talking about pattern recognition [5]–[7]. Accuracy is the ability of a classifier to perform its task as close to the desired target output. While diversity is the ability of classifier to perform classification task on different data sets without compromising the accuracy, and it can be obtained by having several classifiers and choose the best result.

This research studies the effectiveness of a combined fusion-selection approach in ensemble method called Random Linear Oracle (RLO) towards a base classifier, i.e. Naïve Bayes (NB) classifier. As well as how feature selection algorithm could be used to alter the classification performance.

II. BACKGROUND REVIEW

A. Introduction

An ensemble model can be explained by four levels as shown in Figure 1. The first entry level is the data level which involves in manipulating the training data to achieve higher diversity and accuracy. Some popular data manipulation methods are divide-and-conquer, cross-validation, and bootstrap method [2]. The second entry describes the feature level. It involves either to submit all features for training, or choose a bespoke subset by feature selection algorithm.

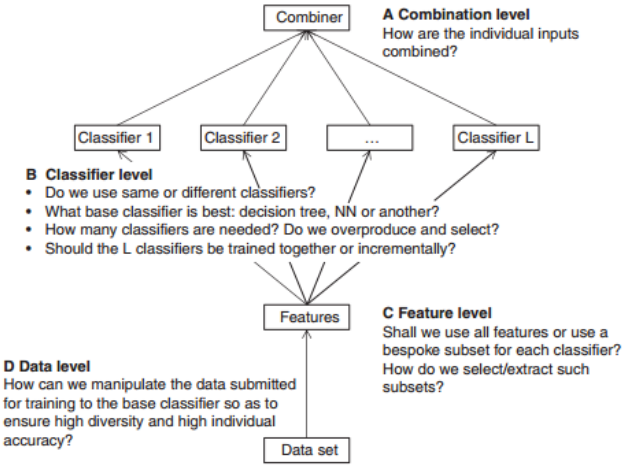

Figure 1: Four levels of ensemble model [2].

Next is the classifier level, where types of classifiers are determined, number of classifier is decided, and how the classifiers are being trained. All results from individual classifier will be combined at the combination level, and a label will be assigned to the object based on the combined result. Some common combining algorithms are majority voting, Naïve Bayes combiner, and multinomial methods.

B. *Naïve Bayes classifier*

Naïve Bayes (NB) classifier is chosen as the base classifier because it involves a simple mathematical model which effectively shortens the training time. If the conditional independence assumption holds true, this classifier can actually outperform many other classification models [8].

NB classifier mainly revolves around three aspects: the prior probability, the posterior probability, and the class-conditional probability [9], [10]. The main purpose is to calculate the posterior probability in term of prior probability and class-conditional probability. 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:

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

Where $P(w_i|\tilde{x})$, $P(w_i)$, and $p(\tilde{x}|w_i)$ are the posterior probability, prior probability, and class-conditional probability for class $i = \{1, \dots, c\}$, respectively. From Equation (1), posterior probability refers to the probability of object $\bar{x}$ belongs to class ω_i, thus higher posterior probability indicates the likelihood of the object to be of class ω_i.

C. *Random linear oracle*

Random Linear Oracle (RLO), introduced by Kuncheva and Rodríguez, is a unique ensemble method that combines both classifier fusion and selection approaches [11]–[15]. In RLO ensemble method, each classifier is replaced with two mini-ensembles plus a random oracle chosen between them.

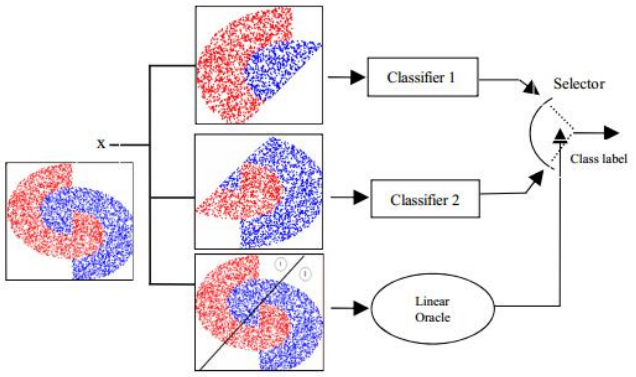

Figure 2: RLO method applied to two-class problem [16].

Training of RLO ensemble involves in a randomly generated oracle of hyperplane over the whole feature space, separating it into two different subspaces. Two classifiers will be trained on each subspace respectively, yielding two different classifiers. Figure 2 illustrates how the RLO method is applied to a two-class problem. Moreover, a number of RLO ensembles will be trained on the same feature space, each with a randomly generated oracle, and two classifiers that are trained on the separated feature subspaces.

When a new sample data comes to test, the location of that data will first be determined by oracle in each ensemble, and a corresponding classifier will be selected to classify the object. All results from all ensembles will be combined by means of 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 of choosing feature subsets that well present the whole data space [17], [18]. The main objective is to choose the bespoke subset while eliminate meaningless features. It is believed that through feature selection process, the classification accuracy of learning algorithm can be improved [19]. This process required more time to be performed, however, when the selection is done, less data will be submitted for training and testing, which greatly reduces the operation time. Three feature selection approaches are available, i.e. filter, wrapper, and embedded. Only filter and wrapper approaches will be discussed as embedded approach is just a combination of both.

1) *Filter approach*

Feature selection by filter approach generates the feature subset by analyzing the properties of data, without the needs of training or testing phase being conducted. Most filter methods are done by ranking and subset selection: by examines the distinctive nature of each feature, the most interesting feature will be rated first, and so on until the least interesting feature. Selection will be done based on the user desired offset percentage out of the overall features number. In this research, the well-known Student's t-test and Kruskal-Wallis test from hypothesis testing will be used to rank each feature based on their test statistic.

However, these tests do not select the important features subset, it is solely used to arrange feature data from the most significant until the least significant. For selection to hold true, only part of the features subset will be used for training and testing, and this can be accomplished by taking a percentage out of the number of features available. This research allocated 75% and 25% of the overall features size for training and testing, respectively.

a) T-test

T-test is a parametric hypothesis testing method proposed by Gosset under the pseudonym "Student" [20]–[22]. 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. This can be done by comparing all features in each data set through testing the null hypothesis. Rejecting the null hypothesis which implies unequal means requires small *p*-value obtained using the test statistic value. In other word, the smaller the *p*-value (i.e. less than α) the higher the confidence level that features are significantly different. Thus, by arranging the *p*-values of each feature, taking the lower *p*-valued features while ignoring the higher *p*-valued features, allows the algorithm to examine most distinctive feature.

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 [23]. 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 comparison of t-test is carried out to determine the *p*-value [22], [24]. A feature contains data from many different classes, so data from two different classes will be compared during each successive t-test, yielding gC_2 combination of test statistics for one feature, where g is the number of classes. Thus, the ranking (ascending) of features could be done by sorting all the mean *p*-values of every feature.

b) Kruskal-Wallis test

Kruskal–Wallis (KW) test developed by Kruskal and Wallis in 1952 [25], is a non-parametric statistical test used when the normality assumption of data is not met. However, like any non-parametric tests, it has less statistical power compared to t-test which is parametric [26].

KW test deals with ranks instead of means and standard deviations. It assigns rank to each data in the testing set regardless of their group in an ascending order, beginning with rank 1 for smallest value, and calculates the test statistic, *H* using Equation (3).

$$H = \frac{12}{N(N+1)}\left(\sum_{i=1}^{k}\frac{{T_i}^2}{n_i}\right) - 3(N+1) \quad (3)$$

Where:

N = total number of data in all features.
k = total number of features.
T_i = sum of ranks assigned to data in ith feature.
n_i = number of data in ith feature.

The *p*-value of test statistic is then obtained from Chi-Square (χ^2) distribution table [23]. Similar steps from previous criteria are repeated for KW except this criterion is even simpler.

2) Wrapper Approach

Feature selection based on wrapper approach generates the feature subset by undergoing training and testing phase, through trial and error method searching for the optimum feature subset that produce the best results. Since this approach requires carrying out training and testing phase, so it will be more time consuming than filter approach, however it often leads to a better performance. Some examples of wrapper approach are genetic algorithm, recursive feature elimination algorithm, and flower pollination algorithm. This research utilizes the popular genetic algorithm for wrapper approach.

a) Genetic algorithm

Genetic algorithm (GA) is based on natural evolution and natural selection approaches, conceived by Holland (1965) [27]. It is a randomized search and optimization technique inspired from natural genetic system. Figure 4 illustrates the working principle of GA.

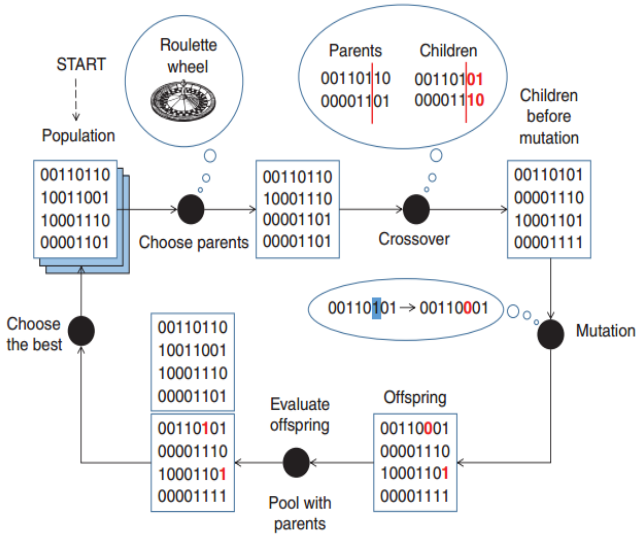

Figure 4: Overall GA flow chart [2].

To use GA in feature selection, it is required to represent each feature in a data set as an individual binary gene, where 1 means the feature is selected and 0 means otherwise. The

string of genes is combined to form a chromosome, and each chromosome represents the possible features combination. The process starts with 20 randomly generated chromosomes, each with number of genes same as the number of features in the data set. Then, each chromosome will be presented to a fitness function to determine its fitness value. In this case, each possible combination of features will be applied to the data set, and the fitness function will be the accuracy of the algorithm with the selected features.

After that, the fitness values will be used to form a roulette wheel, where higher fitness value contributes to higher proportion in the wheel. Two parents will be chosen by spinning the roulette wheel, meaning that chromosome with higher fitness value will has a higher chance to be chosen as the parent. Two chosen parents will undergo crossover, randomly exchanging part of their genes with each other, to form two new children. Crossover procedure will be repeated until the number of new children is equal to the population number. Next, the newly formed children will undergo mutation process, some or none of the genes will be flipped, 1 to 0 and 0 to 1, to encourage extra diversity in the searching. Mutation does not occur frequently as it will directly be causing the result unable to converge, so it is controlled by a value called mutation chance, in this research the mutation chance of chromosomes is set to 0.1.

After mutation, the children are now called offspring, which ends the first iteration, or generation in GA term. Beginning the next generation, the offspring are now treated as the parent, and the overall process restart, until the fitness value converge to a minima, or the maximum number of generations is reached. Finally, the bespoke subset of features will be returned by GA.

In order to observe more varieties and effects in feature selection, GA will be used twice with different approaches, one with accuracy of NB classifier as the fitness function, and another with accuracy of RLO ensemble as the fitness function.

The pseudo code for GA feature selection algorithm is shown in Figure 5.

```
Genetic Algorithm (GA)

G      = number of generations
P      = number of populations
C      = number of chromosomes
Initialize population to P × C matrix.
for each g ϵ G
    for each p ϵ P
        Evaluate fitness value.
    end
    if g == G
        break the loop.
    end
    Generate roulette wheel.
    for i = 0 until P/2
        Select two parents from roulette wheel.
        Crossover at random point.
    end
    Mutation based on chances.
    Set new child as next generation's parents.
end
return individual with max(fitness)
```

Figure 5: Pseudo code for GA feature selection.

E. Test of hypothesis

In order to test the algorithms' performance, two tests were introduced in this research: Mann-Whitney U-test and Friedman test. Both proposed methods are non-parametric, although with less statistical power than the parametric tests, they do not require the conditions of safe usage such as independence, normality, and heteroscedasticity, which were hardly attained in machine learning cases [28], [29]. The purpose of hypothesis test is to check whether the algorithm had a significant different from a control class through comparing the test statistic with a set value of confident level.

1) 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 [30]. Different from the previous tests, U-test will be used to analyze the significant different in median between two algorithms on a same data set. In this research, the classification results of NB classifier will be used as the control class, so the null and alternative hypotheses of this test are written as:

$$H_0 : median_{NB} = median_{algorithm}$$
$$H_A : median_{NB} \neq median_{algorithm}$$

The procedures of U-test are similar to KW test, where rank is assigned to each data in the testing set regardless of their group in an ascending order, beginning with rank 1 for smallest value, but the test statistic U is calculated using Equation (4).

$$U = n_1 n_2 + \frac{n_2(n_2+1)}{2} - T \quad (4)$$

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 [23]. Since the classification results of NB classifier will be used as the control class, so if the *p*-value obtained from a particular 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 particular data set. Else, if the *p*-value is greater than 0.05, there is no significant difference between those two classifiers.

2) Friedman test

Friedman test developed by Friedman (1937) is a statistical test known as the non-parametric counterpart of repeated measures ANOVA [24]. This research uses Friedman test to analyze the overall results of all algorithms. Friedman test uses a ranking approach that is different from KW and U tests.

Assuming the results are $k \times N$ matrix, with k number of rows (blocks), N number of column (treatments), where the blocks show different number of data set, and the treatments are the different number of algorithms. The null and alternative hypotheses of this test are hereby:

$$H_0 : treatment_{NB} = treatment_{algorithm}$$
$$H_A : treatment_{NB} \neq treatment_{algorithm}$$

Friedman test assigns ranks within each block, in a reversing order, beginning with rank 1 for largest value. If there is a tie, average rank will be assigned to all tied members. Friedman test follows the Chi-Square (χ^2) distribution and the test statistic of this test is calculated using the equation:

$$\chi_F{}^2 = \frac{12N}{k(k+1)}\left[\sum_{i=1}^{N} R_i{}^2 - \frac{k(k+1)^2}{4}\right] \quad (5)$$

Where:

$\chi_F{}^2$ = test statistic.
N = number of treatments.
k = number of blocks.
R_i = sum of ranks assigned to *i*th treatment.

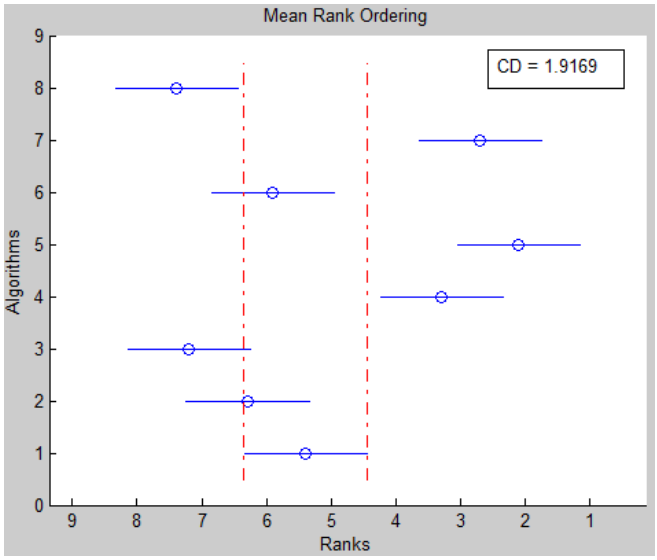

Figure 6: Critical difference graph.

The main purpose of using Friedman test is the ability to construct the Critical Difference Graph for convenient interpretation. Calculating the mean rank of each treatment, and the Critical Difference value (CD), it allows construction of critical difference graph as shown in Figure 6. Since Friedman test assigns rank 1 to the largest value within each block, so a smaller mean rank indicating the algorithm performs better [24].

The performance of the two algorithms is significantly different when their mean ranks differ by at least a critical difference, calculated using the equation:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (6)$$

where:

q_α = critical value based on Studentized range statistic divided by $\sqrt{2}$.

Figure 6 shows the mean ranks of eight algorithms, with extension of half CD to its left and right, so if the whiskers of two algorithms are not overlapping, that two algorithms are significantly different from each other. Dotted lines refer to the lower bound and upper bound of the control class.

III. EXPERIMENT

This research begins with reading a data set, randomly split the data into 6:4 ratio, where 60% is for training, and 40% for testing. The training phase is separated into two parts, one with feature selection, and one without feature selection. Considering the part without feature selection, training data will directly presented to NB classifier and RLO ensemble for training, and using the trained classifiers to operate on testing data. Consider the part with feature selection, training data will first present to selection algorithms namely genetic algorithm, t-test, and Kruskal-Wallis test to choose the bespoke subset of the features. Thus, only important features will be submitted for training and testing a RLO ensemble to investigate the effect of feature selection towards the suggested method.

Results from the above classifications will be recorded, and the procedures were repeated five times using different split of train-test pairs. With a total number of five results from one algorithm, it is possible to investigate the significant different in accuracy between algorithms. For this purpose, Mann-Whitney test will be used, by letting the results from NB classifier as the control class. All the results from each algorithm will be compared with the results from the control class, so that to determine the performance of that algorithm in a particular data set.

Average from all five results will be recorded as the overall result for that algorithm on that data set. The process is repeated for 30 data sets from UCI repository [31], where all data sets have been proven classification feasible. Finally, Friedman test will be used to observe the overall performance across all data sets, using mean rank ordering technique to develop Critical Difference Graph for easy observation. All testing will assume a significant level at $\alpha = 0.05$.

Table 1 shows the properties of each data set obtained from UCI repository, as well as the components of each data set such as 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 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 algorithm used in this section are explained in Table 2.

A. Mean accuracy

Table 3 describes the results for each algorithm of each data set. From the table, mean accuracy of NB classifier is 64.62%, which indicates availability for improvement, but based on the accuracy of RLO ensemble, improvement is only 0.31% using NB classifier. The highest mean accuracy is achieved by RLO-GA-NB algorithm, a 4.75% improvement from NB classifier.

Among the two algorithms that perform weaker than NB classifier are RLO-TT-25 and RLO-KW-25 have almost similar performances, with mean accuracy of 60.98% and 60.05%, respectively. On the other hand, the 75% counterparts of these algorithms yield a great improvement from NB classifier, with 69.26% and 67.08% respectively. This situation leads to an assumption that the output sequences of both feature selection methods are almost the same, but due to the different number of features being selected, the performance of RLO ensemble was greatly affected.

However, reaching into conclusion from average accuracies is somewhat biased because some algorithms prefer data with more objects, but others may favor data with more features. Therefore, it would be better to compare the performance of algorithms with respect to each data set using Mann-Whitney test.

Table 2
Definition of acronyms

Acronym	Definition
NB	Naïve Bayes classifier
RLO	Random Linear Oracle ensemble
RLO-GA-NB	RLO ensemble using genetic algorithm feature selection with accuracy of NB classifier as fitness function
RLO-GA-RLO	RLO ensemble using genetic algorithm feature selection with accuracy of RLO ensemble as fitness function
RLO-TT-75	RLO ensemble using t-test feature selection with 75% selected features
RLO-KW-75	RLO ensemble using Kruskal-Wallis feature selection with 75% selected features
RLO-TT-25	RLO ensemble using t-test feature selection with 25% selected features
RLO-KW-25	RLO ensemble using Kruskal-Wallis feature selection with 25% selected features

Table 3
Accuracy for each algorithm.

Data Set	NB	RLO	RLO-GA-NB	RLO-GA-RLO	RLO-TT-75	RLO-KW-75	RLO-TT-25	RLO-KW-25
Abalone	23.36	24.13	24.58	25.35	22.75	22.81	25.98	26.28
Balance	88.70	88.78	78.93	78.93	76.44	76.76	38.78	42.95
Blood	76.57	77.31	75.97	77.84	75.57	75.77	70.69	68.81
Car	80.82	82.70	79.84	80.04	82.73	82.99	69.57	73.47
Ecoli	43.81	43.81	63.11	64.90	71.72	66.92	64.69	59.04
Glass	45.14	55.57	57.16	62.28	59.50	58.13	48.85	47.00
Ionosphere	64.19	64.19	73.87	68.73	35.81	35.81	35.81	35.81
Iris	95.67	95.67	95.00	95.67	95.67	95.67	69.33	65.33
Leaf	63.82	58.97	64.71	56.18	64.56	63.82	50.74	50.00

Lenses	60.99	34.46	31.64	46.26	55.13	29.41	16.30	16.30
Magic	72.85	75.57	77.90	77.79	76.13	75.90	78.32	76.38
mfeat-fac	80.55	83.40	93.25	92.98	92.80	92.68	89.23	89.10
mfeat-fou	75.68	76.83	76.15	74.73	77.18	78.55	77.93	79.00
mfeat-kar	93.80	95.00	91.83	92.03	94.48	94.98	92.45	93.95
mfeat-mor	38.75	33.88	58.10	58.60	57.78	35.30	54.53	34.63
mfeat-pix	35.95	25.63	52.98	49.33	27.35	23.55	59.05	54.35
mfeat-zer	73.33	74.78	73.45	72.50	73.15	73.63	65.93	62.78
page	92.00	90.60	93.34	94.48	91.65	93.31	91.80	93.72
Pima	75.05	75.77	76.29	76.61	74.98	76.35	75.44	75.70
PokerTrain	50.19	53.31	51.04	51.94	52.88	52.73	50.79	50.22
Segmentation	14.78	14.78	52.10	49.46	80.32	83.48	80.26	67.25
Spect	69.46	65.72	71.52	68.71	66.10	65.18	67.59	69.83
vehicle	60.06	67.64	67.39	67.63	66.28	67.87	63.31	64.67
vowel	62.25	73.51	70.68	67.75	73.89	73.51	58.56	59.22
wfsonar-2	90.67	94.45	94.35	94.23	93.93	94.02	56.81	57.26
wfsonar-24	52.74	61.86	67.40	61.05	58.01	60.88	54.05	63.96
wfsonar-4	89.04	90.86	90.74	93.36	89.50	92.62	52.57	56.28
Wine	96.90	97.19	94.96	96.34	94.95	98.30	90.45	94.39
yeast	30.07	30.07	41.38	42.59	54.95	30.07	38.18	32.29
Zoo	41.50	41.50	41.50	41.50	41.50	41.50	41.50	41.50
Mean	64.62	64.93	69.37	69.33	69.26	67.08	60.98	60.05

B. Win, lose, and tie

Table 4 summarizes the performance of all algorithms when compared to the results using NB classifier. RLO ensemble with only 0.31% improvement in mean accuracy scored 13 wins in the Mann-Whitney test, meaning that RLO ensemble can works significantly better upon 43.3% data sets compared to NB classifier, justifying the improvement in diversity of this method. However, RLO-GA-RLO algorithm with mean accuracy 4.4% higher than RLO ensemble, scored the same win, lose, tie ratio with the ensemble, this indicates that RLO-GA-RLO algorithm could be used to improve the accuracy of RLO ensemble, but maintaining the same diversity distribution. The same applied to RLO-TT-75 algorithm with similar win, lose, tie scores, however with less mean accuracy improvement but shorter computational time as compared to RLO-GA-RLO algorithm.

On the other hand, RLO-GA-NB algorithm which showed the highest improvement in accuracy, recorded only 9 wins in this test, indicating that this algorithm can improves the classification accuracy by a significant amount. But it works well only for limited number of data sets. Also, RLO-KW-75 algorithm scored 14 wins, 12 ties, and 4 loses, which means this algorithm can works well across 46.67% more data sets compared to NB classifier. However, the accuracy improvement is no better than other feature selection algorithms except filter approached with 25% selected features.

Table 4
Total number of win, lose, tie for Mann-Whitney U-test.

Algorithm	Win	Lose	Tie
RLO	13	3	14
RLO-GA-NB	9	3	18
RLO-GA-RLO	13	3	14
RLO-TT-75	13	3	14
RLO-KW-75	14	4	12
RLO-TT-25	8	12	10
RLO-KW-25	8	11	11

An interesting circumstance arouse when combining both results from

Table 3 and Table 4. All filter approach feature selection methods with 25% selected features performed relatively weaker than their counterpart with 75% selected features in term of accuracy and diversity. Such scenario implies that RLO ensemble is highly sensitive to the number of features processed. A properly selected features number can improve the performance of algorithm, whereas a poorly selected features number will worsen the result.

C. Critical difference

Figure 7 combines all algorithms performance in one graph for overall performance analysis. Each algorithm performed just as discussed earlier, with RLO-GA-RLO algorithm as the best performance method at the rank of 3.48, the control class NB classifier ranked at 5.42. RLO-GA-NB placed at second with rank 3.62, followed by RLO-KW-75 algorithm ranker 3.88. RLO ensemble basically ranked at 4.27.

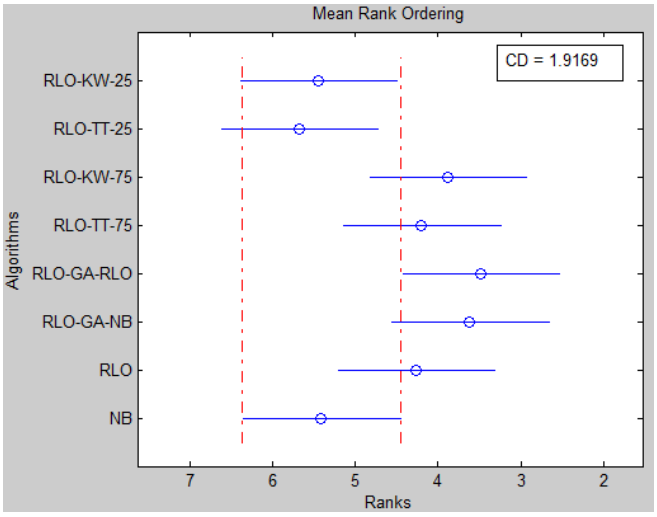

Figure 7: Overall critical difference graph.

NB classifier and RLO ensemble perform better than RLO-TT-25 and RLO-KW-25, but weaker than the other 75% counterparts of these algorithms. This further proves that RLO ensemble will greatly affected by the number of

features used for operation. It is suggested that another extensive research should be carried out to investigate how exactly the number of features selected through feature ranking will affect the performance of ensemble method.

On the other hand, GA uses the concept of randomized search and natural selection technique allowed in building a more robust system. By performing classification by randomly generated feature subsets, and observes the performance from each subset, the algorithm undergoes an evolution process to obtain the best combination of features. Thus, even with a smaller number in features from GA feature selection, the quality of classification is higher than that of filter approach.

Also, RLO ensemble works better than NB classifier, RLO-TT-25, and RLO-KW-25, but not RLO-TT-75, RLO-KW-75, RLO-GA-NB, and RLO-GA-RLO, shows that RLO ensemble is capable of adapting itself to both feature selection filter and wrapper approaches to further enhance its ability, making it a versatile method to be used for improvement.

Eventually, with the CD value of 1.92, no algorithm is considered as significant differences from NB classifier except RLO-GA-RLO algorithm.

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 feature selection algorithms to this method. This research shows that RLO ensemble allows more diversity in data set selection compared to the base classifier alone as discussed in Section IV.A and IV.B. Even though the mean accuracy across 30 data sets only provides 0.31% of improvement, this statistic is not agreed by a critical difference method in Section IV.C which states that the ranking of RLO ensemble is better than NB classifier.

RLO ensemble greatly benefited from GA. In the critical difference method, RLO-GA-RLO ranked at 3.48, whereas NB classifier ranked at 5.42, with a rank difference of 1.94, this algorithm performs significantly better than NB classifier. On the other hand, RLO-GA-NB algorithm is capable of increasing the classification accuracy by a significant amount, but suffer from lost in diversity. Whereas RLO-GA-RLO algorithm with 0.04% less in mean accuracy as compared to RLO-GA-NB, but receives an advantage of having greater diversity. So, it can be concluded that the overall performance RLO-GA-RLO algorithm is better than RLO-GA-NB algorithm.

In conclusion, there is no best classifier or ensemble method per se, RLO ensemble could significantly improve the diversity of NB classifier in dealing with distinctively selected feature sets through its fusion-selection paradigm, but failed in the improvement of classification accuracy. However, when combined with GA feature selection method, RLO ensemble can receive an additional boost in accuracy from the overall mean accuracy perspective. For further improvement, it is suggested to apply and test different 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.

Acknowledgment

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 along 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*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 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] G. Brown and L. I. Kuncheva, "'Good' and 'Bad' diversity in majority vote ensembles," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 5997 LNCS, pp. 124–133.

[6] L. Didaci, G. Fumera, and F. Roli, "Diversity in classifier ensembles: Fertile concept or dead end?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7872 LNCS, pp. 37–48.

[7] Y. Bi, "The impact of diversity on the accuracy of evidential classifier ensembles," *Int. J. Approx. Reason.*, vol. 53, no. 4, pp. 584–607, 2012.

[8] I. Rish, "An empirical study of the Naive Bayes classifier," *IJCAI 2001 Work. Empir. methods Artif. Intell.*, pp. 41–46, 2001.

[9] A. R. Webb, *Statistical pattern recognition*, vol. 71, no. 8. 2011.

[10] A. Jamain and D. J. Hand, "The Naive Bayes mystery: A classification detective story," *Pattern Recognit. Lett.*, vol. 26, no. 11, pp. 1752–1760, 2005.

[11] L. I. Kuncheva and J. J. Rodríguez, "Classifier ensembles with a random linear oracle," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 500–508, Apr. 2007.

[12] J. J. Rodríguez and L. I. Kuncheva, "Naive Bayes ensembles with a random oracle," *Mult. Classif. Syst.*, pp. 450–458, 2007.

[13] C. Pardo, J. J. Rodríguez, J. F. Díez-Pastor, and C. García-Osorio, "Random oracles for regression ensembles," *Ensembles Mach. Learn. Appl.*, pp. 181–199, 2011.

[14] 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.

[15] 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.

[16] G. Armano and N. Hatami, "Random prototype-based oracle for selection-fusion ensembles," *2010 20th Int. Conf. Pattern Recognit.*, pp. 77–80, 2010.

[17] 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.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[19] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature selection using forest optimization algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, 2016.

[20] C. M. Bishop, *Pattern recognition and machine learning*, vol. 4, no. 4. 2006.

[21] 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].

[22] 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.

[23] 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.

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[25] J. McDonald, *Handbook of biological statistics: Introduction*, 3rd ed. Baltimore, Maryland: Sparky House Publishing, 2012.

[26] J. Frost, "Choosing between a nonparametric test and a parametric test," 2015. [Online]. Available: http://blog.minitab.com/blog/adventures-in-statistics/choosing-between-a-nonparametric-test-and-a-parametric-test. [Accessed: 01-Apr-2016].

[27] N. P. Padhy, *Artificial intelligence and intelligent systems*, 14th ed. NewDelhi, India: Oxford University Press, 2015.

[28] M. Graczyk, T. Lasota, Z. Telec, and B. Trawiński, "Nonparametric statistical analysis of machine learning algorithms for regression problems," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2010, pp. 111–120.

[29] B. Trawiński, M. Smętek, Z. Telec, and T. Lasota, "Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms," *Int. J. Appl. Math. Comput. Sci.*, vol. 22, no. 4, pp. 867–881, 2012.

[30] 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.

[31] 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.