<table>
<thead>
<tr>
<th>Title</th>
<th>Empirical comparison of bagging-based ensemble classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Ye, Ren; Suganthan, P. N.</td>
</tr>
<tr>
<td>Citation</td>
<td>Ye, R., &amp; Suganthan, P.N. (2012). Empirical comparison of bagging-based ensemble classifiers. 2012 15th International Conference on Information Fusion (FUSION), 917-924.</td>
</tr>
<tr>
<td>Date</td>
<td>2012</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/19784">http://hdl.handle.net/10220/19784</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2012 International Society of Information Fusion. This paper was published in 2012 15th International Conference on Information Fusion (FUSION) and is made available as an electronic reprint (preprint) with permission of International Society of Information Fusion. The paper can be found at the following official URL: <a href="http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6289900">http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6289900</a>. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.</td>
</tr>
</tbody>
</table>
Empirical Comparison of Bagging-based Ensemble Classifiers

Ren Ye, P.N. Suganthan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
{re0003ye, epnsugan}@ntu.edu.sg

Abstract—This paper compares empirically four bagging-based ensemble classifiers, namely the ensemble adaptive neuro-fuzzy inference system (ANFIS), the ensemble support vector machine (SVM), the ensemble extreme learning machine (ELM) and the random forest. The comparison of these four ensemble classifiers is novel because it has not been reported in the existing literature. The classifiers are evaluated with thirteen binary class datasets and the empirical results show that the ensemble methods employed in the four ensemble classifiers boost the testing accuracy by 1-5% on average from their base classifiers. In addition, the testing accuracy can be improved by increasing the number of base classifiers. The empirical results also show that the bagging SVM is the most favorable ensemble classifier among them.

I. INTRODUCTION

Classification is a common problem in machine learning, which is to group the new observations/samples based on a training set whose identifications (labels) are provided (supervised learning) [1] or unprovided (unsupervised learning, not discussed in the paper) [2]. Classification can be applied in various areas such as medicine, manufacturing, bioinformatics, finance, etc. There are several well-developed tools for classification such as Bayesian, K-nearest neighbors, decision trees, fuzzy neural networks (FNN), support vector machines (SVM) and so on.

There are two phases to design a classifier: (i) supervised learning (unsupervised learning is not discussed in the paper) and (ii) evaluation of generalization performance. However, due to the variance in the training data, the bias of the attributes, etc., different accuracies may be obtained with different classifiers, or same classifiers with different parameters. In other words, there is no universal classifier for all types of data distributions [1].

In order to improve the performance of classifiers, ensemble learning can be used. Ensemble learning is to use multiple models to achieve a better predictive performance than that could be obtained from any of the constituent models [3]. There are two categories of ensemble learning, one is competitive ensemble learning and the other is cooperative ensemble learning [4]. The base classifiers work individually on one set of a problem and vote for the final decision is competitive ensemble learning whereas the base classifiers work associatively with each other and aggregate the decisions to the final decision is cooperative ensemble learning.

The advantage of ensemble learning is the improvement of the accuracy [5]. The key determinant to better accuracy and generalization is ensemble diversity. There are mainly three strategies for ensemble diversity: (i) kernel diversity, (ii) parameter diversity and (iii) data diversity [4].

- Kernel diversity: Kernel function converts the data space to a non-linear space so that a linear decision boundary can be used in the non-linear space to classify the data. Kernel functions play an important role in SVM and ELM. Varying the kernel functions for different ensembles will diversify the base classifiers.
- Parameter diversity: Different parameters will vary the performance of classifiers. For example, the parameters that affect the performance of fuzzy neural network (FNN) are the membership function, number of nodes, rules, aggregation methods, etc. and for SVM, the parameters affects the performance are the margin parameter and the parameters associated with the kernel functions.
- Data diversity: For a finite dataset, different data perturbations may lead to significantly different classification configurations. Data diversity minimizes the bias and variance and increases the generalization capability of the classifiers. Cross validation (CV) is a commonly used method in machine learning to avoid overfitting [6]. Other advanced data diversity algorithms such as bootstrap aggregation (bagging) [7] and boosting [8] are widely used in ensemble methods.

The ensemble classifiers can make use one or more of the strategies mentioned above in the actual implementation. This paper will focus on data diversity, especially on bagging-based ensemble classifiers.

Bagging is a method for generating multiple datasets for a classifier and use the classification results of each datasets to obtain an aggregated classifier. Bagging-based ensemble learning can be categorized as competitive learning and data diversity strategy.

In the literature, there are a few bagging classifiers reported. The original bagging algorithm was applied to classification trees [7]. The bagging classification tree improved the performance of the classification tree significantly in terms of error reduction on eight datasets. In [9], the bagging algorithm was employed to a Bayesian predictor. In [10], the bagging algorithm was employed to a tree-based ranking predictor. The
empirical results showed that with bagging, both predictors outperformed the original predictors on most of the datasets. An improved bagging-based classification tree is reported in [2]. The algorithm is called Random Forest. Random forest introduces an additional mode of randomness to bagging, thereby increasing the data diversity. In addition to bootstrapping on training data, which is a standard procedure in bagging, random forest improves the construction of the classification trees. The normal way of constructing a classification tree is to split the node at the best split among all attributes whereas for random forest, the node is split using the best split among a randomly chosen subset of attributes [11]. The authors claimed that random forest yields good accuracy and robustness.

Bagging can also be applied to neural networks and FNNs [12], [13]. In [13], an ensemble fuzzy classifier was created with bagging and adaptive neuro-fuzzy inference systems (ANFIS). Twenty datasets were evaluated and the bagging ANFIS outperformed the original ANFIS on most of the datasets in terms of accuracy.

Support Vector Machine (SVM) is a machine learning tool for classification as well as regression [14]. The advantage of SVM is that it can be implemented in a high-dimensional linear attribute space but with a non-linear classifier by appropriately choosing a kernel function. The kernel function enables SVM to perform well for high-dimensional datasets. However, choosing a kernel function and the parameters of the kernel function usually undergoes an exhaustive search process [5]. Improperly chosen kernel functions and parameters will reduce the accuracy of the SVM.

Two variations of bagging-based ensemble SVMs were reported in [15], [16]. A low bias bagged SVM was reported in [16]. The authors claimed that the bagging algorithm in the learning phase should be subjected to minimizing bias even with a trade-off on increasing variance. Therefore, the bagging algorithm was modified to a lobag algorithm. The lobag algorithm first estimated the parameters of an SVM that minimizes the bias and then based on the chosen parameters creating bags for the SVM classifiers. In [15], a double SVM bagging was presented. The authors modified the conventional bagging SVM with two phase classification: first phase was an SVM classification and second phase was a decision tree classification. Unlike the conventional bagging SVM classifiers, the reported double SVM bagging algorithm used the out-of-bag samples for training and combined the bagging sample to a probability matrix. In the testing phase, decision tree was employed and the final decision was voted by the confidence levels. Both ensemble SVMs returned a conclusion of improved performance.

An ensemble SVM based on cooperative ensemble learning was presented in [1]. The data diversity was realized by iteratively learning from the training data and adding the false predicted data in the previous iteration as support vectors in the next iteration. The accuracy was determined by aggregating the decisions of all iterations. However, the authors noticed that for certain datasets, there were no convergence but oscillations and the oscillation problem is still un-solved.

Extreme learning machine (ELM) is an effective machine learning algorithm based on single-hidden layer feedforward neural network (SLFN) architecture [17]. It is a fast algorithm and can provide good generalization. There are several ensemble ELM developed in the literature.

A parameter diversity based ensemble ELM was reported in [18]. The ELM classifiers were created by k-fold CV. For each fold, there were a few training iterations and the parameters $w$ and $b$ of the output neuron feature space $\mathbf{H}$ were tuned for better performance in each training iteration. The best performed parameters were saved for each ELM classifiers. The testing results were decided by a majority vote.

Another parameter diversity based ensemble ELM was presented in [19]. The authors created the ensembles by assigning different initial weight values to different ELM classifiers and followed by training with ELM. Therefore, a lot of ELM classifiers were created. Finally the testing result was aggregated by a majority voting. Both of the ensemble ELMs [18], [19] outperformed the original ELM in classification.

A bagging-based ensemble ELM regressor was developed in [20]. The bagging algorithm was employed to bootstrap the training dataset to several sub-training datasets. ELM regressor was applied on each sub-training datasets and the error rates were calculated and stored as weights of vote. The testing results were decided by a weigh vote combining method. This paper will use the similar algorithm but with ELM classifier for empirical study on binary classification problem.

The above-mentioned classifiers belong to FNN, kernel machine and decision trees. This paper will study four typical classifiers from these four categories, they are ANFIS of FNN, SVM of kernel machine, ELM of FNN and kernel machine and random forest of decision trees.

The objective of this paper is to evaluate four bagging-based ensemble classifiers: the bagging ANFIS, the bagging SVM, the bagging ELM and the random forest. The performance of these four bagging ensemble methods are compared. The evaluation is based on thirteen UCI binary class datasets. To the authors’ knowledge, the comparison of the four ensemble classifiers has not been studied in the existing literature.

The remaining of the paper is organized as follows: Section II elaborates the details of the bagging algorithm and four classifiers; Section III details the ensemble SVM, the ensemble ANFIS, the ensemble ELM and the random forest; Section IV discusses and compares the empirical results and finally Section V concludes the paper.

II. CLASSIFIERS

A. Preliminary Definition

Let $\mathbf{D} = \{\mathbf{x}\}$ be a dataset with $n$ samples and each sample has $d$ attributes. Let $\mathbf{T}$ be a training set and $\mathbf{S}$ be a testing set, where $\mathbf{D} = \mathbf{T} + \mathbf{S}$. There is an array of identification labels $y \in \{-1, 1\}$ mapping to $\mathbf{D}$ and therefore $y = y_T + y_S$, where $y_T$ and $y_S$ correspond to the labels of $\mathbf{T}$ and $\mathbf{S}$ respectively. The general idea of a classifier is to find $\hat{y}_S$ based on a
function $\hat{y}_S = f(S, y_S)$, where the function $f(\cdot)$ is determined by a learning function $l(T, y_T)$.

### B. ANFIS

Adaptive Neural Fuzzy Inference System (ANFIS) [21] is an FNN structure based on TSK fuzzy inference system model [22]. It processes the data in a fuzzy space. The adaptive and learning abilities are realized by the ANN structure. The architecture of a typical ANFIS is shown in Fig. 1.

As shown in Fig. 1, there are six layers in the ANFIS architecture (six-layer feed-forward neural network). The example ANFIS architecture has two inputs $x_1, x_2$, and one output $y$. Each input is fuzzified by two fuzzy sets $A_1, A_2, B_1, B_2$ corresponding to four membership functions.

- **Layer 1** is the input layer. No specific function is applied.
  \[ O_1^i = x_i \]  

- **Layer 2** is the fuzzification layer. Inputs from Layer 1 are converted into fuzzy numbers by the membership functions $\mu(\cdot)$. The membership functions are usually chosen to be non-linear bell shaped functions.
  \[ O_2^i = \mu_{A_i}(x_i) \times \mu_{B_i}(x_2) \]  

- **Layer 3** is the rule layer. Each fuzzified input undergoes an IF-ELSE rule and combined by multiplication (T-NORM).
  \[ O_3^i = O_1^i \times O_2^i \]  

- **Layer 4** is the normalization layer. The outputs of the rule layers are normalized by the firing strength (the weights) and then are combined by summation (S-NORM).
  \[ O_4^i = O_3^i \sum_j O_3^j \]  

- **Layer 5** is the defuzzification layer. The defuzzification layer formed a series of linear equations based on the inputs and the outputs from the normalization layer.
  \[ O_5^i = O_4^i f_i = O_4^i (p_i x_1 + q_i x_2 + r_i) \]  

- **Layer 6** is the output layer. The linear equations are summed to form a single linear equation. The output of the ANFIS is the output of the linear equation.
  \[ O_6 = \sum_i O_5^i \]  

The learning algorithm of ANFIS comprises two parts, one is the back-propagation gradient descent method which is usually used in ANN and the other is the least-square estimator which is usually used in linear statistics. The gradient descent method is mainly used for updating the antecedent parameters and the least-square estimator is mainly used for updating the consequent parameters.

The users can choose different combinations of learning algorithms, such as gradient descent only, gradient descent plus one pass of least square estimator, etc. based on the design objectives (speed v.s. accuracy).

However, ANFIS has a fixed architecture for a particular dataset. The learning algorithm can only change the parameters of the membership functions and the coefficients of the output linear equations but not the fuzzy rules. This limitation reduces the adaptability of ANFIS for complex datasets with large number of attributes and samples. In addition, the output of ANFIS is a floating number and therefore, the output should be round off to the nearest values of $y$ for final decision.

### C. SVM

$d$-dimensional samples can be mapped on a $d$-dimensional hyperspace. The basic idea of SVM is to draw the best hyperplane in the hyperspace to separate the samples according to their corresponding labels $y$. The hyperplane is optimum when the margin of separation between samples with different labels is maximized. The samples sitting on the margins are called support vectors [14].

Suppose the samples are linearly separable, then a hyperplane in the hyperspace can be defined as:

\[ wx_i + b \]  

where $w$ is a vector perpendicular to the hyperplane and $b$ is a bias.

For a linearly-separable case, two classes are separated by $wx + b = 0$ (a solid line in Fig. 2) and there is also a region bounded by $wx + b = 1$ and $wx + b = -1$ (two dashed lines in Fig. 2) in between the two classes so that a soft margin is created.
The distance between two margins is \( \frac{2}{|w|} \). So, the best hyperplane is obtained by solving:

\[
\min \frac{1}{2} w^T w
\]

subject to

\[ y_i (w x_i + b) \geq 1 \]

For linearly un-separable samples, certain samples will be misclassified. They are represented by non-negative variables \( \xi_i \) such that:

\[ y_i (w x_i + b) \geq 1 - \xi_i. \]

Therefore, the optimization problem becomes:

\[
\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i
\]

subject to

\[ y_i (w \phi (x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \]

where \( C \) is the penalty factor to weight the misclassification and \( \phi (\cdot) \) is a nonlinear function to map the input to higher dimensional space.

If \( C \) is large, the solution converges towards the solution obtained by the optimal separating hyperplane with more tolerance of misclassification. If \( C \) is small, the solution converges to one where the margin maximization dominates [23]. The optimization equation (10) can be represented more formally by Lagrange method as shown:

\[
\max L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi (x_i)^T \phi (x_j)
\]

subject to

\[
\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C
\]

where the term \( \phi (x_i)^T \phi (x_j) \) can be represented by a kernel function \( K(x_i, x_j) \).

The advantage of the kernel function is to save the computational effort of computing the higher-dimensional mapping of \( x_i \) and \( x_j \) separately by computing the internal product of \( x_i \) and \( x_j \) first and then map to the specific higher-dimensional space.

There are several popular kernel functions such as polynomial, radial basis function (RBF) and sigmoid, etc. The users can choose a proper kernel with proper kernel parameters for a specific dataset.

**D. ELM**

ELM has a neural network like structure, but it is advantageous over neural network because the hidden layer needs not be neuron alike and needs not be tuned [17]. The output function of an ELM binary classifier is a function of the hidden layer feature space \( h(\cdot) \) and the output weights \( \beta \) as shown:

\[
f_L(x) = sign(h(x)\beta)
\]

This function is equivalent to mapping a \( d \)-dimensional input data to an \( L \)-dimensional hidden layer feature space \( h \) and then scale by the output weights of \( L \) hidden neurons.

The output weights \( \beta \) is determined by the hidden layer output matrix \( H = [h(x_1) \ldots h(x_{|T|})] \) as well as the pre-defined labels \( y_T \) shown:

\[
\beta = \begin{cases} 
(H^T H)^{-1}H^T y_T & H^T H \text{ is nonsingular}, \\
(H(HH^T)^{-1}H^T y_T & HH^T \text{ is nonsingular}.
\end{cases}
\]

To minimize the error \( |\hat{y}_S - y_S| = |H\beta - y_S| \) and \( |\beta| \) Lagrange multiplier method is used and finally the equation of binary ELM classifier is shown:

\[
f(x) = sign(h(x)H^T (\frac{1}{C} + HH^T)^{-1}y_T)
\]

ELM can be easily adapted to multi-class classification and regression. The advantage of ELM is the fast training speed due to its neural network alike structure.

**E. Bagging**

Usually \( T \) is small and with a large variance. The bootstrap phase in bagging under-samples data with replacement from \( T \) according to the uniform probability distribution and creates \( B \) independent datasets (bags) \( T^{(1)} \ldots T^{(B)} \). The purpose of making \( B \) bags is to reduce the variance of the original dataset \( T \). The aggregation phase in bagging aggregates the decisions drawn from each bag by voting or other equivalent methods to reduce bias [7], [16]. Therefore, bagging can reduce variance and in the mean time reduce bias. The schematic diagram of bagging is shown in Fig. 3.

The reasons why bagging over performs a single classifier are listed in [24]. There are three reasons: statistical, computational and expressiveness. The statistical reason is that the learning in bagging aggregates different possible hypotheses to obtain the best hypotheses, which is advantageous over a single hypothesis. The computational reason is that the learning algorithm in bagging has less chance to be trapped in a local minimum than the single classifier. The expressive reason is that the bagging classifier can expand the search space to higher dimensional space which improves the expressiveness from the single classifier.

**F. Random Forest**

Random forest is developed from decision trees. The basic idea of decision trees is to classify a sample by binary partition. At each node of the tree, a test is applied to the input sample and a decision (the direction of the preceding branches) is made by evaluating the outcome of the test. The test terminates at a leaf node (no more preceding branches) and the final classification is made. Decision tree is fast, easy to understand and insensitive to a missing data [25].

Random forest improves the decision trees by bagging and random subspace method. A random forest consists of a set of decision tree classifiers where each classifier is trained by independent identically distributed random vectors and the final decision is aggregated by a majority vote [2].
The advantage of using bagging for creating the decision trees is to increase the diversity of the base classifiers. In each node, random forest employs random subspace method for node splitting, which adds one more degree of randomness to the base classifier. Therefore, the author claimed that the random forest is a promising ensemble method [2].

### III. Ensemble Classifiers

This section shows the detailed algorithms of the bagging ANFIS, the bagging SVM, the bagging ELM and the random forest. The generalized algorithm of the bagging-based ensemble classifier is shown in Fig. 4. There are four nested loops in the algorithm: the outer loop is to repeat the classification process to get an average testing result; the first inner loop is to train a classification model for each bag; and the second and third inner loop is to find the best parameters for the single classifier for that particular bag based on a k-fold CV. The testing phase includes a majority vote to obtain the final decision.

For the bagging ANFIS, the best performing ANFIS model is obtained by k-fold CV only because ANFIS will learn and update the parameters by its back-propagation and least-square estimator. For the bagging SVM, the parameter C and kernel parameter γ (for RBF kernel) will be tuned by a grid search. For the bagging ELM (sigmoid kernel), the parameter C and the number of hidden layer nodes L will be tuned by a grid search as well.

In order not to overfit each base classifier, the parameter is searched once before bootstrapping. If the grid search is implemented for each base classifier after bootstrapping, the base classifiers will have high chance to be overfitted to the training data.

The random forest is pre-compiled from a R package [11]. There are several parameters can be tuned in the program but the only parameter to be tuned in the experiment is the number of bags so that there will be a fair comparison against the other three ensemble classifiers.

The number of stratified testing S is set to be 50 in the empirical analysis; the number of bags B is set to 20, 50 and 80 for three different experiments; and the ratio of the training data and the testing data is \( n_T : n_S = 70\% : 30\% \). The training epochs of ANFIS is set to 10. The grid search of the bagging SVM has two dimensions: \( C \in \{2^i, i = -5, -4, \ldots, 5\} \) and \( \gamma \in \{2^i, i = -4, -3, \ldots, 4\} \). The grid search of the bagging ELM has two dimensions as well: \( C \in \{2^i, i = -5, -4, \ldots, 5\} \) and \( L \in \{10, 20, \ldots, 1000\} \).

**Fig. 3: A Schematic Diagram of Bagging**

**Fig. 4: The Bagging-based Ensemble Classification Algorithm**

### IV. Empirical Analysis

The bagging ANFIS, the bagging SVM, the bagging ELM and the random forest were evaluated using thirteen UCI
binary classification datasets [26] and shown in Table I. The ANFIS algorithm is the standard release in the Matlab Fuzzy Logic Toolbox based on [21]; The SVM algorithm is built from a package called libsvm [27]; The ELM algorithm is obtained from [17] and the random forest algorithm is obtained from [28] that is converted from the R package reported in [11]. In order to speed up the computation of ANFIS, three attributes were chosen for each datasets under ANFIS and the bagging ANFIS evaluation. The attributes were chosen based on one-pass single attribute ANFIS classification with ascending classification error.

The tuned parameters for SVM and ELM averaged over 5-fold CV are shown in Table II. The tuned parameters of SVM did not vary much among each fold of CV whereas the tuned parameters of ELM, especially L, had a large standard deviation among each fold of CV.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of Samples</th>
<th>No. of Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian (credit)</td>
<td>690</td>
<td>14</td>
</tr>
<tr>
<td>Breast (wisconsin)</td>
<td>699</td>
<td>9</td>
</tr>
<tr>
<td>Bright</td>
<td>2462</td>
<td>14</td>
</tr>
<tr>
<td>Credit</td>
<td>690</td>
<td>15</td>
</tr>
<tr>
<td>Diabetes</td>
<td>768</td>
<td>8</td>
</tr>
<tr>
<td>Dimdata</td>
<td>4192</td>
<td>14</td>
</tr>
<tr>
<td>Heart</td>
<td>294</td>
<td>12</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>155</td>
<td>19</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>33</td>
</tr>
<tr>
<td>Liver</td>
<td>345</td>
<td>6</td>
</tr>
<tr>
<td>Mammo (graphic)</td>
<td>961</td>
<td>5</td>
</tr>
<tr>
<td>Mushroom</td>
<td>8124</td>
<td>22</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>8</td>
</tr>
</tbody>
</table>

The base classifier accuracies (averaged over 80 base classifiers) before majority vote is shown in Table III. Compared with the aggregated accuracy shown in Table VI, the average of base classifier accuracies is generally 1 – 5% less. For example, the improvements of the average testing accuracies for ionosphere from before to after aggregation are 8.03%, 2.54%, 2.26% and 7.80% for bagging ANFIS, bagging SVM, bagging ELM and random forest, respectively. In terms of diversity, the standard deviations of the base classifiers are generally more than unity. These two tables show that bagging indeed increases the accuracy by aggregating base classifiers’ decisions. For example, the standard deviations of the base classifiers’ testing accuracies of ionosphere are 1.94, 2.27, 3.35 and 3.15 for bagging ANFIS, bagging SVM, bagging ELM and random forest, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Bagging ANFIS</th>
<th>Bagging SVM</th>
<th>Bagging ELM</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>83.29(0.49)</td>
<td>84.36(1.67)</td>
<td>85.20(2.41)</td>
<td>79.96(4.43)</td>
</tr>
<tr>
<td>Breast</td>
<td>92.77(2.00)</td>
<td>95.87(1.45)</td>
<td>95.01(1.26)</td>
<td>93.76(1.23)</td>
</tr>
<tr>
<td>Bright</td>
<td>97.10(0.38)</td>
<td>99.03(0.31)</td>
<td>99.06(0.38)</td>
<td>97.59(0.77)</td>
</tr>
<tr>
<td>Credit</td>
<td>85.12(1.69)</td>
<td>82.96(2.32)</td>
<td>81.55(2.53)</td>
<td>79.44(3.68)</td>
</tr>
<tr>
<td>Diabetes</td>
<td>73.86(1.22)</td>
<td>74.67(2.30)</td>
<td>72.98(3.40)</td>
<td>67.51(3.35)</td>
</tr>
<tr>
<td>Dimdata</td>
<td>68.34(0.48)</td>
<td>95.53(0.51)</td>
<td>95.60(0.54)</td>
<td>92.01(0.88)</td>
</tr>
<tr>
<td>Heart</td>
<td>60.68(2.04)</td>
<td>77.27(5.23)</td>
<td>80.86(5.14)</td>
<td>76.63(3.47)</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>80.20(4.45)</td>
<td>79.21(1.85)</td>
<td>78.09(4.70)</td>
<td>76.36(5.98)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>81.02(1.94)</td>
<td>92.15(2.27)</td>
<td>85.76(3.35)</td>
<td>85.17(3.15)</td>
</tr>
<tr>
<td>Liver</td>
<td>59.92(3.55)</td>
<td>68.17(4.11)</td>
<td>67.21(4.19)</td>
<td>61.72(5.44)</td>
</tr>
<tr>
<td>Mammo</td>
<td>81.40(1.61)</td>
<td>82.20(1.42)</td>
<td>81.03(2.09)</td>
<td>78.30(2.00)</td>
</tr>
<tr>
<td>Mushroom</td>
<td>74.69(0.75)</td>
<td>88.90(0.75)</td>
<td>88.90(0.76)</td>
<td>83.04(3.11)</td>
</tr>
<tr>
<td>Pima</td>
<td>75.14(1.34)</td>
<td>74.79(2.21)</td>
<td>75.92(2.58)</td>
<td>68.36(2.75)</td>
</tr>
</tbody>
</table>

Tables IV, V and VI show the testing accuracies of four ensemble classifiers with 20 bags, 50 bags and 80 bags, respectively. The testing accuracies are averaged over 50 stratified testings. Generally, the testing accuracies of the four bagging-based ensemble classifiers increase when the number of bags increases. This is because that the data diversity is related to the number of base classifiers [24].

For the four bagging-based ensemble classifiers with the same number of bags, the bagging SVM is the most favorable ensemble classifier among them. When there are 20 bags, the bagging SVM has the best predicted accuracies on 7 datasets out of 13; when there are 50 bags, the bagging SVM has the best predicted accuracies on 6 datasets out of 13; and when there are 80 bags, the bagging SVM has the best predicted accuracies on 8 datasets out of 13. The second most favorable ensemble classifier is the random forest. When there are 20 bags, the random forest has the best predicted accuracies on 3 datasets out of 13; when there are 50 bags, the random forest has the best predicted accuracies on 5 datasets out of 13; and when there are 80 bags, the random forest has the best predicted accuracies on 4 datasets out of 13.

The four ensemble classifiers are ranked by their testing accuracies of each dataset for 20 bags, 50 bags and 80 bags, respectively. The overall rank is calculated by the summation of the individual ranks. The overall rank of the four ensemble classifiers for different bags is: bagging SVM < random forest < bagging ELM < bagging ANFIS.

V. CONCLUSION

This paper has discussed four bagging-based ensemble classifiers, namely the ensemble ANFIS, the ensemble SVM, the ensemble ELM and the random forest. The discussion of the four ensemble classifiers is novel because it is not reported in the existing literature. The ensemble classifiers have been evaluated on thirteen UCI binary class datasets with different bagging numbers (20, 50 and 80). The empirical results have
shown that all four ensemble classifiers have returned high testing accuracy. The effect of data diversity has also been reflected from the comparison of base classifier accuracies and the ensemble classifier accuracies. In addition, the effect of data diversity has also been reflected from the number of bags because the results have shown that the testing accuracies increase with increasing of the base classifiers. Among the four ensemble classifiers, the ensemble SVM has been identified to be the most favorable ensemble classifier and the random forest has been identified to be the second most favorable one. The bagging ELM and the bagging ANFIS have been identified to be the third and fourth most favorable classifiers.

### REFERENCES


