Optimising Weights for Heterogeneous Ensemble of Classifiers with Differential Evolution

Mohammad Nazmul Haque*†,‡ Nasimul Noman*†, Regina Berretta*† and Pablo Moscato*†§
*Priority Research Centre for Bioinformatics, Biomarker Discovery and Information-Based Medicine,
Hunter Medical Research Institute, New Lambton Heights, NSW, Australia
†School of Electrical Eng and Computer Science, The University of Newcastle, Callaghan, NSW, Australia
§Information-based Medicine Program, Hunter Medical Research Institute, New Lambton Heights, NSW, Australia
Email: MohammadNazmul.Haque@uon.edu.au, {Nasimul.Noman, Regina.Berretta, Pablo.Moscato}@newcastle.edu.au

Abstract—The classification performance of a weighted voting ensemble of classifiers largely depends on the proper weight chosen for each base classifier's vote. In this paper, we propose the use of Differential Evolution algorithm for adjustment of voting-weights of base classifiers used in a heterogeneous ensemble of classifiers (HEoC). We used the average Matthews Correlation Coefficient (MCC), calculated over 10-fold cross-validation, as the quality measure of an ensemble. We applied the vanilla DE algorithm to maximise the average MCC score over the training dataset. The algorithm optimises the base classifiers' voting weights in order to attain better generalisation performance of the ensemble on testing datasets. Experiments were performed using 10 binary-class datasets taken from UCI-Machine Learning Repository. The results show consistent and superior generalisation performance of the constructed ensembles when compared with the base classifiers and other well-known ensemble of classifiers.

I. Introduction

An ensemble of classifiers (EoC) utilises decisions from multiple base classifiers to reduce classification error. It is not usually guaranteed that an ensemble will improve over the single best classifier [1]. However, empirical studies published till date have demonstrated that an EoC is typically more accurate than a single classifier [2]. Therefore, EoCs are receiving increased attention and gaining popularity day by day [3]–[6].

There are two major types of EoCs: one is homogeneous, created with different instances of the same base classifier [7], [8] and the other is heterogeneous, created with diverse types of base classifiers. Each type of base classifier has some advantages over others in learning different aspects of the datasets; therefore, learning from diverse base classifiers could enhance the overall classification performance. Outputs from multiple classifiers need to be merged into a single decision. Accumulation of the voting by base classifiers is one of the most widely used and simplest approaches to decision fusion. In majority voting approach, votes from different base classifiers are treated equally, despite the non-identical classification performances [9]. On the other hand, in weighted voting approach, votes from different base classifiers are weighted differently in the decision fusion process. Moreover, empirical study on decision fusion approach has revealed that weighted voting could significantly improve the classification performances [7], [10], [11].

Different approaches have been utilised to adjust the weights of base classifiers in EoCs so far. Usage of different types of discrimination measures [5], [12], [13], dynamic adjustment [14], linear programming [15] and game theory [16] are some examples of weight adjustment approaches found in the literature. Several Evolutionary Algorithms (EA) have been proposed as well. Genetic Algorithm (GA) [3], [17], [18] and Differential Evolution (DE) algorithm [19], [20] were successfully employed in weight optimisation. Maghsoudia et al. [18] proposed a GA to adjust the weights of base classifiers in homogeneous ensembles. The weights were adjusted for the overall accuracy of each class in random subspaces of the dataset. Bhadra et al. [19] proposed a DE to optimise weights of a homogeneous ensemble of classifiers. They used a combined fitness function with different classifier performance measures and tested it on three datasets. EAs have also been employed for voting-weight optimisation in the small-scale heterogeneous EoC (HEoC). For instance, Ekbal and Saha [3] proposed a GA for optimising the weights of HEoC for named-entity recognition problem. Each base classifier received a separate weight per class-labels based on the f-measure score. They created several instances of base classifiers varying the training features. Liu et al. [17] used a GA for weight optimisation of vote-based extreme learning machine. Optimised weights were used to form the ensemble of neural-network classifiers. Zhang et al. [20] proposed a DE algorithm for optimising weights of five base classifiers for voting in a HEoC. They used accuracy as the fitness score to optimise the weight of base classifiers. These applications of EAs demonstrate the potential in optimising base classifiers' weights in the vote based ensemble of classifiers. Most of the experiments performed for weighted voting in the literature were for the homogeneous ensemble of classifiers. Very few heterogeneous ensemble of classifiers were proposed, and those were formed with a small number of base classifiers. Hence, in this work, we investigate the construction of weighted vote based EoC from a large collection of heterogeneous base classifiers and explored its suitability and robustness.

In this paper, we have adopted an approach to select the best combination of base classifiers for creating the heterogeneous EoC and to optimise their voting weights using differential evolution (DE) algorithm [21]. DE is a simple but efficient and

978-1-5090-0623-6/16/$31.00 ©

robust evolutionary algorithm for optimisation of real-valued parameters. This powerful optimisation algorithm is used with the Matthews correlation coefficient (MCC) score [22] as the fitness value to determine the optimal weights for base classifiers used in a HEoC. The algorithm finds the best combination of base classifiers alongside optimising their weights. We use the MCC as the objective function of DE, because it gives more balanced measure about the generalisation performances of a classifier than other popular measures (such as accuracy, precision and recall) [23]. The proposed algorithm that optimises the voting-weights of base classifiers in a HEoC using DE is called `DE-HEoC`.

II. The Differential Evolution

Storn and Price [21], [24] introduced differential evolution (DE) in 1995 for numerical optimisation. DE is much simpler and easier to implement compared to many other EAs. Despite its simplicity, DE exhibited remarkable performance in solving a wide variety of real-world problems in reasonable amount of computation time [25].

DE is a parallel direct search method for optimising a D-dimensional real-valued parameters. It starts with a randomly generated initial population of NP individuals and evaluates their fitness. The i-th individual (parameter vector) in the population for the current generation G is given as:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, \cdots, x_{D,i,G}]$$

where $i = 1, 2, \cdots, NP$ and $G = 0, 1, 2, \cdots, G_{max}$ is the generation number. Each of the parameter values is usually bounded by lower and upper limits. DE utilises the same computational framework that is used by other standard EAs, therefore, three operators, namely mutation, crossover and selection, are used for creating a new generation of individuals. This process is repeated until the stopping criterion is satisfied. Brief descriptions of the DE operators are presented below.

1) Mutation: After initialisation of the population, DE creates a donor vector $\vec{V}_{i,G}$ for each population member or target vector $\vec{X}_{i,G}$ in the current population using the mutation operation. Storn and Price have proposed a couple of alternative mutation and crossover strategies for DE [21], [24]. In this work, we used the $DE/rand/1$ mutation strategy which is given by

$$DE/rand/1 : \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F.(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$$

where, r_1^i, r_2^i and r_3^i are mutually exclusive integer indices randomly chosen from the population for individual $\vec{X}_{i,G}$. The amplification factor, F, controls the scaling of the difference vectors.

2) Crossover: DE uses crossover operation to generate the trial vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, \cdots, u_{D,i,G}]$ from the donor vector and the target vector. Generally, two types of crossover operators are used in classic DE, namely exponential and binomial crossover. The DE variant used in this work utilises the binomial crossover (bin). For each j-th variable form D dimensions, the binomial crossover operates as follows:

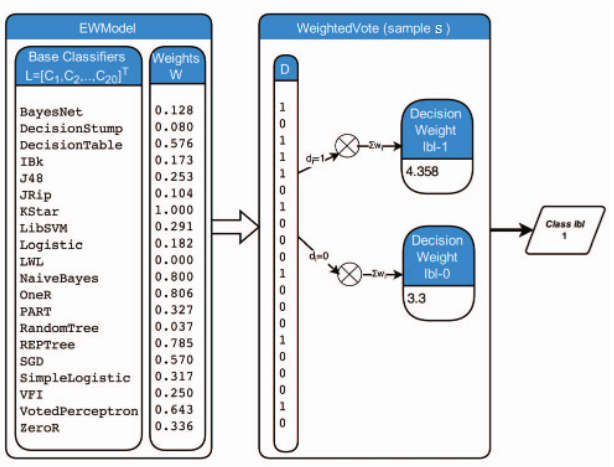


Fig. 1. An example showing the process of deciding the class label of a sample (individual evaluation process) from the weighted voting of an ensemble of heterogeneous classifiers.

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } (rand_{j,i}(0,1) \leq C_r) \text{ or } j = I_{rand} \\ x_{j,i,G}, & \text{otherwise} \end{cases}$$

The crossover probability $0 \leq C_r \leq 1$ is a user-defined parameter to control the fraction of parameter values that are copied from the donor vector. I_{rand} is a random integer from $\{1, 2, \cdots, D\}$ to ensure that at least one variable is copied from the donor vector $\vec{V}_{i,G}$. This is called binomial crossover, because the number of inherited parameters from the donor has almost the binomial distribution.

3) Selection: The selection operator determines whether a target or the corresponding trial vector survives to the next generation. The selection is made as:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G}, & \text{if } Obj(\vec{U}_{i,G}) \geq Obj(\vec{X}_{i,G}) \\ \vec{X}_{i,G}, & \text{otherwise} \end{cases}$$

where, $Obj(\vec{X}_{i,G})$ is the objective function to be maximised. Therefore, the new trial vector $\vec{U}_{i,G}$ promoted to the next generation only if it produces better or equal objective score compared to the objective score of the target vector.

A. The Proposed DE-HEoC

We employ DE to optimise the voting-weights of base classifiers in a heterogeneous ensemble of classifiers (HEoC). First, we describe how the ensemble of classifiers is constructed, and then we explain the weighted voting approach. Finally, we present how the differential evolution algorithm is used for weight optimisation.

1) Construction of Ensemble of Classifiers: We build an ensemble of classifiers using 20 heterogeneous classifiers listed in Fig. 1. We have taken diverse types of commonly used base classifiers from WEKA data mining software suite [26]. Each base classifier is associated with a weight. Each classifier

in the ensemble casts a weighted vote for deciding the class label (cl) of a data sample S. The class label with higher total weighted score becomes the class label for the sample.

2) Decision with Weighted Majority Vote: Here, we define the decision process in the weighted voting ensemble (EW), constructed from D base classifiers, for a binary classification problem. Assume that class labels $\Omega = [cl_1, cl_2, \cdots, cl_D]^T$ for an unknown sample S are given by D base classifiers $L = [C_1, C_2, \cdots, C_D]^T$, where $cl_i \in \{0, 1\}$ and $i = 1, 2, \cdots, D$. Each base classifier C_i is associated with one weight $0 \leq x_i \leq 1$ and encoded as a parameter of DE individuals (shown in Fig. 1). From the weighted voting ensemble, total weights for class label 0 (ω^0) and class label 1 (ω^1) is calculated as $\omega^0 \leftarrow \sum_{i=1}^{D} x_i, i/cl_i = 0$ and $\omega^1 \leftarrow \sum_{i=1}^{D} x_i, i/cl_i = 1$, respectively. The weighted voting ensemble decides the class label of S using ω^0 and ω^1 as follows:

$$EW(S) = \begin{cases} 0, & \text{if } \omega^0 > \omega^1 \\ 1, & \text{if } \omega^0 < \omega^1 \\ Rand(0,1), & otherwise \end{cases}$$

Here, the class decision goes for the maximum weight gaining class label. The class label is randomly selected in case of a tie. The decision making process for a sample, S, is further explained in Fig. 1.

3) Differential Evolution for Weight Optimisation: We used $DE/rand/1/bin$ for optimising the value of 20 parameters. Each parameter in the individuals of DE corresponds to the voting-weight of one heterogeneous base classifier. The objective in DE-HEoC is to maximise the MCC score of heterogeneous ensemble of classifiers. The MCC score quantifies the strength of classification considering both the true positive rate and the true negative rate. It is calculate from the confusion matrix as:

$$MCC = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}}$$

where tp, tn, fp and fn denotes the true positive, true negative, false positive and false negative values, respectively. A higher MCC score indicates better prediction. The experimental results in [27] indicate it as an ideal measure for the analysis of confusion matrix. Therefore, we consider MCC as our measure of classification performance.

The working process of DE-HEoC is shown in Algorithm 1 which returns the best weighted voting ensemble (EW_{best}) for the training data. For internal validation, we created 10 train-fold data (TD) and 10 validation-fold data (VD) from the training data. Each base classifier in the pool L is trained on each fold of TD and saved as trained model (TM) for future usage. The fitness of an individual is determined (line 3-5) by calculating the average MCC score on validation-fold data VD. DE iterates by creating a new generation from the current generation (line 6-22) until the stopping condition is satisfied. The DE-HEoC returns the optimised weights of the base classifiers in the ensemble which produced the maximum average MCC score validation-fold data VD.

Algorithm 1: The DE-HEoC Algorithm

```
   Input: NP, D, F, Cr, TM, VD
   Output: EW_best
1  G ← 0
2  Pop ← InitialisePopulation(NP, D)
   /* Calculate Fitness Value of each
      Individuals in Initial Population  */
3  for X⃗_{i,G} ∈ Pop do
4  |  Pop.fit[i] ← FitnessEvaluation(X⃗_{i,G}, TM, VD, D)
5  end
6  while StopCondition! = Satisfied do
7  |  NewPop ← ϕ
8  |  for X⃗_{i,G} ∈ Pop do
   |  |  /* Randomly chose 3 mutually
   |  |     exclusive parents              */
9  |  |  X⃗_{r1^i,G} ← RandomMember(Pop)
10 |  |  X⃗_{r2^i,G} ← RandomMember(Pop)
11 |  |  X⃗_{r3^i,G} ← RandomMember(Pop)
   |  |  /* Mutation Operation             */
12 |  |  V⃗_{i,G} ← DE/rand/1(X⃗_{r1^i,G}, X⃗_{r2^i,G}, X⃗_{r3^i,G}, F)
   |  |  /* Crossover Operation            */
13 |  |  U⃗_{i,G} ← binomCrossover(X⃗_{i,G}, V⃗_{i,G}, Cr)
   |  |  FitnessEvaluation(U⃗_{i,G}, TM, VD, D)
   |  |  /* Selection Operation            */
14 |  |  if Obj(U⃗_{i,G}) ≥ Obj(X⃗_{i,G}) then
15 |  |  |  NewPop.add(U⃗_{i,G})
16 |  |  else
17 |  |  |  NewPop.add(X⃗_{i,G})
18 |  |  end
19 |  end
20 |  Pop ← NewPop
21 |  G ← G + 1
22 end
23 EW_best ← GetBestSolution(Pop, NP)
24 return EW_best
```

The function FitnessEvaluation() for an individual is shown in Algorithm 2. Here, we created one weighted vote ensemble (EWModel) for each fold (f). We used base classifiers weights (W) and pre-trained base classifiers (TM) on respective train-fold data to build the ensemble. This weighted voting ensemble is evaluated on respective validation-fold data (f-th fold of VD) and MCC score is calculated. The average MCC on 10 validation-folds is used as the fitness score (fit) of an individual and returned by FitnessEvaluation() process.

The function GetBestSolution() is shown in Algorithm 3. It compares the fitness score of each individual in population (Pop) and saves the best individual ($\vec{Best}_{i,G}$). Finally it returns the best individual in the population having the maximum

Algorithm 2: Pseudo-code of FitnessEvaluation

```
Input: X⃗_{i,G}, TM, VD, D
Output: fit
/* Get each base classifiers weights */
for c ← 1 to D do
    W[c] ← x_{c,i,G}
end
/* Create and evaluate weighted vote
   ensemble for each folds */
MCC ← 0.0
for f ← 1 to 10 do
    EWModel ← ϕ
    /* Build weighted vote EoC for f */
    for c ← 1 to D do
        Cls ← TM[c][f]
        EWModel.add(Cls)
    end
    /* Evaluate and get MCC score on
       validation data VD for fold f */
    fMcc ← EWModel.evaluate(W, VD[f])
    Mcc ← Mcc + fMcc
end
fit ← Mcc/10
X⃗_{i,G}.fitness ← fit
return fit
```

Algorithm 3: Pseudo-code of GetBestSolution

```
Input: Pop, NP
Output: Best⃗_{i,G}
Best⃗_{i,G} ← Pop[1]
/* Compare fitness value for each
   Individual in the Population */
for i ← 2 to NP do
    if Obj(Pop[i]) ≥ Obj(Best⃗_{i,G}) then
        Best⃗_{i,G} ← Pop[i]
    end
end
return Best⃗_{i,G}
```

fitness score.

The success of DE is highly dependent on the parameter selection [28]. However, choosing the parameters value is itself a combinatorial optimisation problem. To select the value for parameters, we assigned a combination of amplification factor (F) and crossover rate (Cr), both within the range of 0.5 to 1.0. The changes of objective value for each combination is recorded for 50 generations on three datasets and most suitable parameter values were selected for the DE-HEoC. They are shown in Table I. We use the maximum evaluation threshold as the stopping criterion in DE-HEoC.

The algorithm returns the optimised weights of base classifiers in DE-HEoC. Next, we will use the constructed HEoC to validate its generalisation performances on unknown testing data.

TABLE I
PARAMETER SETTINGS OF THE PROPOSED DE-HEOC

Parameter	Value
Individual length (D)	20
Population Size (NP)	100
Mutation Strategy	DE/rand/1/bin
Scaling Factor (F)	0.9
Crossover Rate (C_r)	0.6
Objective Function	max(MCC)
Maximum Evaluation	1000

TABLE II
CHARACTERISTICS OF 10 BINARY CLASS DATASETS TAKEN FROM THE UCI-ML REPOSITORY.

Dataset	#Feat (R/I/N)[a]	#Samp	Class Distribution
appendicitis	7 (7/0/0)	106	85, 21
australian	14 (3/5/6)	690	383, 307
bupa	6 (1/5/0)	345	145, 200
haberman	3 (0/3/0)	306	81, 225
monk-2	6 (0/6/0)	432	204, 228
pima	8 (8/0/0)	768	500, 268
saheart	9 (5/3/1)	462	302, 160
sonar	60 (60/0/0)	208	97, 111
titanic	3 (3/0/0)	2201	1490, 711
wdbc	30 (30/0/0)	569	212, 357

[a]The count of Real (R), Integer (I) and Nominal (N) types of features.

III. EXPERIMENTS

In this section, we describe the datasets, details of the experimental setup and results.

A. Datasets

We considered 10 binary-class benchmark datasets from the University of California Machine Learning (UCI-ML) repository [29] to evaluate the proposed method. Key characteristics of those datasets are shown in Table II. It shows the number of features, sample counts and the class distribution of samples for each dataset. Features in these datasets have already been selected at the source. Therefore, they contain a small number of features (the maximum feature count is 60) for the datasets. The table also shows the count for each type of features: Real number (R), Integer (I) and Nominal (N) within parenthesis. The appendicitis, sonar, titanic and wdbc datasets contain only real numbers as feature values. On the other hand, the haberman and monk-2 datasets have only integer feature values. The rest of the datasets contain a mixture of real, integer and nominal feature values. Thus, we have selected diverse types of datasets for the experiment.

TABLE III
SUMMARY OF CLASSIFICATION PERFORMANCES (IN MCC AND ACCURACY SCORES) ACHIEVED BY PROPOSED DE-HEOC FOR 30 RUNS ON 60-40 SPLIT OF PARTICIPATING DATASETS.

Datasets	MCC			Accuracy (%)		
	Best	Avg.	Std.	Best	Avg.	Std.
appendicitis	0.62	0.55	0.04	88.68	86.16	1.41
australian	0.81	0.76	0.02	90.72	88.19	0.84
bupa	0.51	0.34	0.07	75.00	66.90	3.11
haberman	0.50	0.44	0.03	81.70	78.82	1.08
monk-2	1.00	1.00	0.00	100.00	100.00	0.00
pima	0.53	0.49	0.02	79.17	77.48	0.82
saheart	0.40	0.33	0.04	73.59	71.20	1.37
sonar	0.78	0.71	0.04	88.46	84.26	2.05
titanic	0.54	0.53	0.01	80.00	79.56	0.36
wdbc	0.96	0.94	0.02	98.24	97.14	0.74

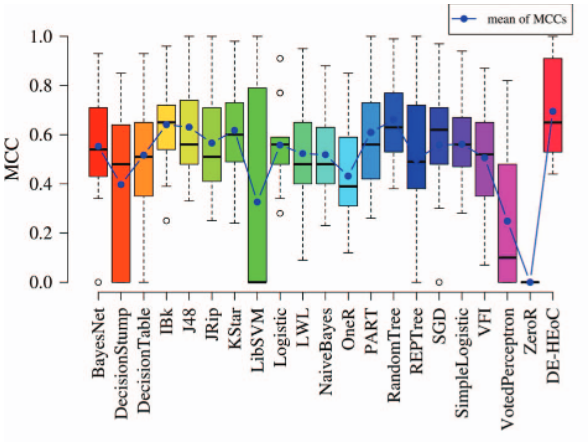


Fig. 2. Comparisons of MCC scores achieved by base classifiers and DE-HEoC (average of 30 runs) on 10 datasets from UCI-ML repository.

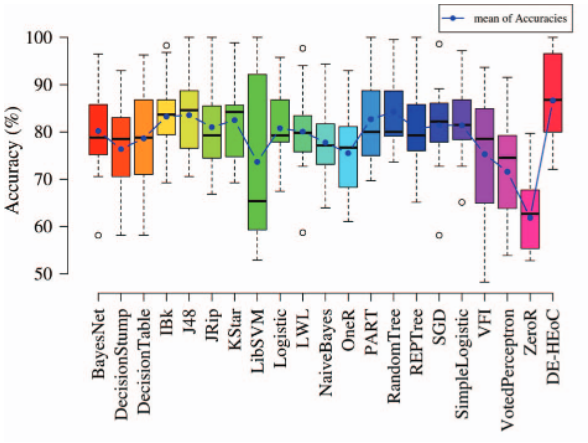


Fig. 3. Comparisons of accuracies achieved by base classifiers and DE-HEoC (average of 30 runs) on 10 datasets from UCI-ML repository.

B. Experimental Setup

The DE-HEoC algorithm was implemented in Java language and was compiled with JDK version 7. We used Weka 3.7 data mining framework [26] and jMetal framework 4.3 [30] for implementing the DE-HEoC. All the experiments were executed in Dell PowerEdge III equipped with Dual Intel Xeon 5405 CPU of 2.00 GHz (8 Cores) and 32 GB RAM. The operating system of the machine was Red Hat Enterprise Linux Server 6.6. The experiments were repeated 30 times on each of the datasets with different random seeds, and the average score is reported. The DE-HEoC program and source code are available for non-commercial usage at the website: http://sourceforge.net/projects/de-heoc/.

C. Results

The performance of DE-HEoC on the 60-40 split of the chosen benchmark is presented in Table III. The best, average (avg.) and standard deviation (std.) of the classification performance measured both in terms of MCC and accuracy are reported in the table. The standard deviations of MCCs achieved by the DE-HEoC are within 0.07 for all 10 datasets. It is less than 4% in terms of accuracy scores. These low deviant measures (both in terms of MCC and accuracy) in all experiments highlight the consistency in DE-HEoC's performance.

D. Discussion

We have conducted further analysis on the experimental results to have a deeper insight in DE-HEoC's performance. We compared the performances of the base classifiers, other state-of-the-art ensemble of classifiers and DE-HEoC on the same train-test data splits. We have used AdaBoostM1 (AB) [31], Bagging (BG) [8], Random Forest (RF) [32] and Random Committee (RC) [26] as other state-of-the-art ensemble of classifiers.

1) Comparison with Base Classifiers: We have compared the classification performances obtained by the base classifiers with the DE-HEoC. The classification performances on the scale of MCC and accuracy achieved by each base classifier and DE-HEoC for the chosen 10 datasets are summarised in Fig. 2 and Fig. 3, respectively. In these box-and-whisker plots, the upper and lower hinges represent the first (Q_1) and third (Q_3) quartiles (the 25^{th} and 75^{th} percentiles) of data. The whiskers are expanded as far as the highest and lowest values that are not outliers. Data points outside the 1.5 inter quartile range ($IQR = Q_3 - Q_1$) from hinges are marked as outliers. The second quartile (Q_2) or median is shown by the horizontal line on the box. The trends of average score is shown by dot-line.

We have created box-and-whisker plots of MCCs achieved for 10 datasets by each base classifiers and DE-HEoC in Fig. 2. The dot-line expresses the trends of mean MCCs of these classifiers. Between 20 base classifiers, the IBk classifier performs better than others with higher median and the mean MCC scores. The ZeroR classifier is the worst performing base classifier considering the MCC score. It is clear from the figure that the mean and median of MCC scores achieved by

TABLE IV
COMPARISON OF MCC SCORES ACHIEVED BY DE-HEOC (AVERAGE OF 30 RUNS) AND FOUR STATE-OF-THE-ART ENSEMBLE CLASSIFIERS FOR 10 BENCHMARKING DATASETS. RESULTS ARE SUMMARISED AS THE NUMBER OF BEST AND WORST PERFORMANCES.

Dataset	AB	BG	RF	RC	DE-HEoC
appendicitis	0.51	0.51	0.56	0.69	0.62
australian	0.64	0.77	0.77	0.80	0.81
bupa	0.22	0.43	0.59	0.50	0.51
haberman	0.40	0.42	0.67	0.63	0.50
monk-2	0.88	1.00	1.00	1.00	1.00
pima	0.45	0.54	0.51	0.56	0.53
saheart	0.26	0.23	0.26	0.40	0.40
sonar	0.68	0.62	0.58	0.71	0.78
titanic	0.48	0.54	0.54	0.54	0.54
wdbc	0.91	0.96	0.92	0.94	0.96
#Best	0	3	4	5	6
#Worst	8	3	1	0	0

TABLE V
COMPARISON OF ACCURACY (%) ACHIEVED BY DE-HEOC (AVERAGE OF 30 RUNS) AND FOUR STATE-OF-THE-ART ENSEMBLE CLASSIFIERS FOR 10 BENCHMARKING DATASETS. RESULTS ARE SUMMARISED AS THE NUMBER OF BEST AND WORST PERFORMANCES.

Dataset	AB	BG	RF	RC	DE-HEoC
appendicitis	84.91	84.91	86.79	90.57	88.68
australian	81.16	88.41	88.12	89.57	90.72
bupa	63.37	71.51	79.65	75.58	75.00
haberman	75.16	76.47	83.66	79.08	81.70
monk-2	93.52	100.00	100.00	100.00	100.00
pima	76.04	79.69	78.13	80.47	79.17
saheart	69.70	68.40	69.70	74.46	73.59
sonar	83.65	79.81	77.88	84.62	88.46
titanic	78.55	80.00	80.00	80.00	80.00
wdbc	95.77	98.24	96.48	97.18	98.24
#Best	0	3	4	5	5
#Worst	8	2	1	0	0

DE-HEoC for 10 datasets are higher than those of all base classifiers.

In the box-and-whisker plots of Fig. 3 we summarise the accuracies achieved by DE-HEoC and 20 base classifiers for 10 datasets. Base classifiers named IBk, SGD and VFI achieved similar mean and median accuracy in all experiments. Their performances are also better than other base classifiers' achievements. The median and the mean accuracy achieved by DE-HEoC for 10 datasets have a higher value than all base classifiers.

From these experimental results, we conclude that DE-HEoC clearly outperformed all base classifiers in experiments on 10 datasets, in terms of both MCC and accuracy measures.

2) Comparison with Ensemble of Classifiers: We compared the proposed DE-HEoC with four state-of-the-art ensemble of classifiers, namely AdaBoostM1 (AB), Bagging (BG), RandomForest (RF) and RandomCommittee (RC) available in the WEKA data mining suite [26]. We used the default parameter values for those algorithms in our experiments. The classification performances obtained by these EoCs are shown in Table IV and Table V in terms of the MCC and accuracy scores, respectively. The last two rows of the table show the number of times that an ensemble has appeared as the top (#Best) and bottom (#Worst) performer in all experiments.

The MCCs achieved by each classifier ensemble are shown in Table IV for experiments on 10 datasets. Here, the RandomCommittee exhibited top performance in five datasets and never became the worst performer. The AdaBoostM1 could never become the best performer among selected EoCs on these dataset classifications, considering the MCC score. Moreover, it has been reported eight times as the worst performing ensemble in the experiments. DE-HEoC has been highlighted six times as the top performing EoC considering the MCC scores and has never appeared as the worst performing EoC. Therefore, DE-HEoC scored better MCCs than other classifier ensembles used in the experiments.

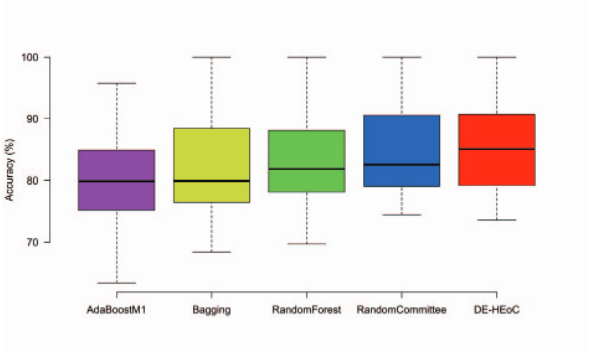


Fig. 4. Comparison of accuracy scores achieved by state-of-the-art ensemble of classifiers and DE-HEoC (average of 30 runs) for 10 benchmarking datasets.

In terms of accuracy, the classification performances achieved by the ensemble of classifiers are shown in Table V. We can see that, AdaBoostM1 exhibited the worst accuracy eight times in experiments among all ensembles of classifiers. Moreover, it has never scored the best accuracy in any of the datasets classification. On the other hand, both of the RandomCommittee and the DE-HEoC have been highlighted five times as better performing classifier ensemble for experiments in 10 datasets. Both EOC were never appeared worse as in any of the test cases. Fig. 4 shows the box and whisker plots for the accuracies achieved by the state-of-the-art ensemble of classifiers and the DE-HEoC for 10 benchmarking dataset. From the figure it is clear that the median accuracies of DE-HEoC is higher than that of any other ensemble methods. Thus, DE-HEoC produces better generalisation than other ensemble of classifiers in the selected 10 benchmark datasets in terms of accuracy.

Comparing both the MCC and accuracy measures on the benchmark of 10 datasets, it can be concluded that DE-HEoC exhibited the best performance compared to other state-of-the-

TABLE VI
CHARACTERISTICS OF HEART DISEASE PREDICTION DATASETS.

Dataset	#Feat (R/I/N)	#Samp	#(Train, Test) Samp
ERIC	7(0/3/4)	209	10-fold CV
HeartDisease	13(1/12/0)	303	10-fold CV
SPECT	22(0/44/0)	267	(80, 187)
SPECTF	44(0/44/0)	267	(80, 187)
Statlog	13(7/3/3)	270	10-fold CV

art EoCs.

IV. CASE STUDY: APPLICATION IN HEART DISEASE PREDICTION

In 2015, Bashir et al. [33] proposed a weighted voting ensemble of classifier system to predict heart disease. They used 5 binary class benchmark datasets to verify the effectiveness of their weighted voting EoC. These benchmark datasets are taken from UCI-ML repository [29], except the ERIC [34] dataset. Descriptions of those datasets including the test and train distribution are tabulated in Table VI. We use the same set of data to compare the performance of our proposed weighted voting-based ensemble of classifier in terms of MCC score. We repeated the experiments on each dataset 30 times.

After executing the DE-HEoC for 30 times on each of the heart disease datasets, we plot the performance with the aid of box and whisker plots for accuracy and MCC in Fig. 5. The sub-figure 5a shows the performances in terms of MCC scores for each datasets. Here we can see that the performances of DE-HEoC over 30 repetitions are very consistent because the spread of boxes are very narrow. This consistency is also observed in accuracy scores, shown in sub-figure 5b. From the performance analysis on heart disease prediction, we can claim that the DE-HEoC is a robust method for classification.

In Table VII, we compared the average performances of DE-HEoC with the best performance of the EoC proposed in Bashir et al. [33] in different measures. For the sake of fairness, we compared our result in terms of F-Measure, Accuracy and MCC scores calculated from the confusion matrices of Bashir et al. [33]. In terms of F-Measure, our method achieved a better average score for HeartDisease, SPECT and SPECTF datasets but outperformed for all experiments if considering the standard deviation. In terms of accuracy, our method outperformed Bashir et al.'s method in three datasets. In MCC measure, the DE-HEoC exhibited better scores in all datasets, except the Statlog.

Taking all measures and datasets in consideration, it is very clear that the proposed DE-HEoC algorithm can achieve better generalisation in heart disease prediction than the weighted-voting EoC proposed in Bashir et al. [33].

V. CONCLUSION

We propose the use of differential evolution algorithm to optimise the voting-weights of base classifiers used in a heterogeneous ensemble of classifiers. These weights are optimised to maximise the average MCC scores calculated in 10-fold cross-validation of a training dataset. The performance of the weighted ensemble of classifiers has been evaluated on 10 benchmark datasets and compared with the results achieved by base classifiers and four other state-of-the-art ensembles of classifiers. The overall classification performance achieved by the proposed method is found to be better in experiments on the selected datasets. The experimental performances were compared with a recently proposed weighted voting ensemble method for heart disease prediction datasets. The result comparison revealed that the DE-HEoC as a better choice for predicting heart disease on those datasets. However, the proposed method exhibited an overall superiority in data classification over individual classifiers and other ensembles of classifiers compared in the experiments. It is expected that it would perform consistently on other datasets. Furthermore, the differential evolution algorithm demonstrates its potential in optimising weights of base classifiers for voting in heterogeneous ensemble of classifiers.

ACKNOWLEDGMENT

The wdbc breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

REFERENCES

[1] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms, 2nd Edition*. John Wiley & Sons, Inc., 2014.

[2] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4 – 20, 2008, special Issue on Applications of Ensemble Methods.

[3] A. Ekbal and S. Saha, "Weighted Vote-Based Classifier Ensemble for Named Entity Recognition: A Genetic Algorithm-Based Approach," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 10, no. 2, pp. 9:1–9:37, 2011.

[4] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble Classifiers for Steganalysis of Digital Media," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 2, pp. 432–444, 2012.

[5] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles," *Knowledge and Information Systems*, vol. 38, no. 2, pp. 259–275, 2014.

[6] S. Saha and A. Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data & Knowledge Engineering*, vol. 85, pp. 15–39, 2013, natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.

[7] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

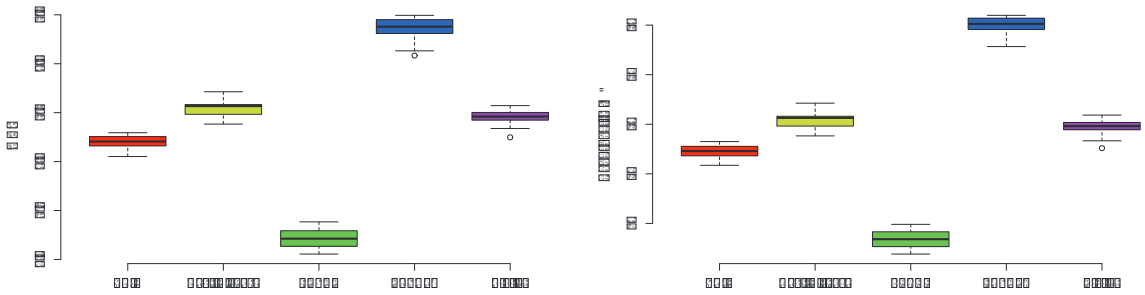
[8] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[9] S. Mao, L. Jiao, L. Xiong, S. Gou, B. Chen, and S.-K. Yeung, "Weighted classifier ensemble based on quadratic form," *Pattern Recognition*, vol. 48, no. 5, pp. 1688–1706, 2015.

[10] Y. Sun, M. S. Kamel, and A. K. Wong, "Empirical Study on Weighted Voting Multiple Classifiers," in *Pattern Recognition and Data Mining*, ser. Lecture Notes in Computer Science, S. Singh, M. Singh, C. Apte, and P. Perner, Eds. Springer Berlin Heidelberg, 2005, vol. 3686, pp. 335–344.

[11] M. Wozniak and K. Jackowski, "Some Remarks on Chosen Methods of Classifier Fusion Based on Weighted Voting," in *Hybrid Artificial Intelligence Systems*, ser. Lecture Notes in Computer Science, E. Corchado, X. Wu, E. Oja, . Herrero, and B. Baruque, Eds. Springer Berlin Heidelberg, 2009, vol. 5572, pp. 541–548.

[12] S. Bashir, U. Qamar, and F. H. Khan, "Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote based ensemble," *Quality & Quantity*, vol. 49, no. 5, pp. 2061–2076, 2015.



(a) (b)

Fig. 5. Box and whisker plots showing the classification performances achieved by DE-HEoC for 30 runs on heart disease prediction datasets considering the a) MCC and b) Accuracy score.

TABLE VII
COMPARISON OF BEST CLASSIFICATION PERFORMANCES BY BASHIR ET AL. (2015) AND THE AVERAGE PERFORMANCES OF DE-HEOC FOR HEART DISEASE PREDICTION DATASETS.

Dataset	Bashir et al. (2015) [33]			DE-HEoC		
	F-Measure	Accuracy(%)	MCC	F-Measure	Accuracy(%)	MCC
ERIC	78.51	75.19	0.62	78.33 ± 1.14	82.28 ± 0.67	0.64 ± 0.013
HeartDisease	82.17	81.82	0.66	87.10 ± 0.58	85.14 ± 0.84	0.70 ± 0.015
SPECT	77.15	80.75	0.35	78.52 ± 0.46	73.85 ± 0.66	0.45 ± 0.015
SPECTF	73.00	72.73	0.27	90.13 ± 1.84	95.00 ± 0.82	0.87 ± 0.021
Statlog	87.38	87.57	0.74	86.92 ± 0.60	84.89 ± 0.80	0.69 ± 0.016

[13] M. Wozniak, "Classifier Fusion Based on Weighted Voting - Analytical and Experimental Results," in Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on, vol. 2. IEEE, 2008, pp. 687–692.

[14] R. Valdovinos and J. Sánchez, "Combining Multiple Classifiers with Dynamic Weighted Voting," in Hybrid Artificial Intelligence Systems, ser. Lecture Notes in Computer Science, E. Corchado, X. Wu, E. Oja, A. Herrero, and B. Baruque, Eds. Springer Berlin Heidelberg, 2009, vol. 5572, pp. 510–516.

[15] L. Zhang and W.-D. Zhou, "Sparse ensembles using weighted combination methods based on linear programming," Pattern Recognition, vol. 44, no. 1, pp. 97–106, 2011.

[16] H. Georgiou, M. Mavroforakis, and S. Theodoridis, "A Game-Theoretic Approach to Weighted Majority Voting for Combining SVM Classifiers," in Artificial Neural Networks - ICANN 2006, ser. Lecture Notes in Computer Science, S. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Eds. Springer Berlin Heidelberg, 2006, vol. 4131, pp. 284–292.

[17] N. Liu, J. Cao, Z. Lin, P. P. Pek, Z. X. Koh, and M. E. H. Ong, "Evolutionary Voting-Based Extreme Learning Machines," Mathematical Problems in Engineering, vol. 2014, pp. 1–7, 2014.

[18] Y. Maghsoudi, A. Alimohammadi, M. V. Zoej, and B. Mojaradi, "Weighted combination of multiple classifiers for the classification of hyperspectral images using a genetic algorithm," in ISPRS Commission VII Mid-term Symposium on Remote Sensing: From Pixels to Processes, 2006.

[19] T. Bhadra, S. Bandyopadhyay, and U. Maulik, "Differential Evolution Based Optimization of SVM Parameters for Meta Classifier Design," Procedia Technology, vol. 4, pp. 50–57, 2012, 2nd International Conference on Computer, Communication, Control and Information Technology (C3IT-2012) on February 25-26, 2012.

[20] Y. Zhang, H. Zhang, J. Cai, and B. Yang, "A Weighted Voting Classifier Based on Differential Evolution," Abstract and Applied Analysis, vol. 2014, 2014.

[21] R. Storn and K. Price, "Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[22] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," Biochim. Biophys. Acta, vol. 405, no. 2, pp. 442–451, 1975.

[23] R. Dutt and A. Madan, "Predicting biological activity: Computational approach using novel distance based molecular descriptors ," Computers in Biology and Medicine, vol. 42, no. 10, pp. 1026–1041, 2012.

[24] K. V. Price, "An introduction to differential evolution," in New ideas in optimization, D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, Eds. Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999, pp. 79–108.

[25] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," Evolutionary Computation, IEEE Transactions on, vol. 15, no. 1, pp. 4–31, 2011.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations Newsletter, vol. 11, no. 1, pp. 10–18, 2009.

[27] G. Jurman, S. Riccadonna, and C. Furlanello, "A Comparison of MCC and CEN Error Measures in Multi-Class Prediction," PLoS ONE, vol. 7, no. 08, p. e41882, 2012.

[28] R. Sarker, S. M. Elsayed, T. Ray et al., "Differential evolution with dynamic parameters selection for optimization problems," Evolutionary Computation, IEEE Transactions on, vol. 18, no. 5, pp. 689–707, 2014.

[29] M. Lichman, "UCI Machine Learning Repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[30] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," Advances in Engineering Software, vol. 42, no. 10, pp. 760–771, 2011.

[31] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in Thirteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1996, pp. 148–156.

[32] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[33] S. Bashir, U. Qamar, and F. H. Khan, "A Multicriteria Weighted Vote-Based Classifier Ensemble for Heart Disease Prediction," Computational Intelligence, 2015.

[34] R. Rakotomalala, "Heart Disease Male," 2013, (Date last accessed on 9-Nov-2015). [Online]. Available: http://eric.univ-lyon2.fr/~ ricco/dataset/heart_disease_male.xls