Research Journal of Applied Sciences, Engineering and Technology 11(12): 1336-1343, 2015
DOI: 10.19026/rjaset.11.2241
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: May 30, 2015 Accepted: August 5, 2015 Published: December 25, 2015

Research Article

Optimal Classifier Ensemble Design Based on Cooperative Game Theory

Jafar A. Alzubi
Al-Balqa Applied University, Jordan

Abstract: Classifier ensemble techniques have been an active area of machine learning research in recent years. The aim of combining classifier ensembles is to improve the accuracy of the ensemble compared to any individual classifier. An ensemble can overcome the weakness of an individual classifier if its base classifiers do not make simultaneous errors. In this study, a novel algorithm for optimal classifier ensemble design called Coalition-based Ensemble Design (CED) is proposed and studied in detail. The CED algorithm aims to reduce the size and the generalization error of a classifier ensemble while improving accuracy. The underlying theory is based on the formation of coalitions in cooperative game theory. The algorithm estimates the diversity of an ensemble using the Kappa Cohen measure for multi base classifiers and selects a coalition based on their contributions to overall diversity. The CED algorithm is compared empirically with several classical design methods, namely Classic ensemble, Clustering, Thinning and Most Diverse algorithms. Experimental results show that the CED algorithm is superior in creating the most diverse and accurate classifier ensembles.

Keywords: Classification, classifiers diversity, classifier ensemble, cooperative game theory, internet security, kappa cohen measure, machine learning

INTRODUCTION

Recently, classifier ensembles have attracted the attention of many researchers as they can in crease the accuracy of classification for many tasks, especially the complex ones (Banfield *et al*., 2003). The most important two factors when designing classifier ensembles are: how to select individual classifiers that are as diverse as possible and how to combine the different out puts in a way that enhances the ensemble accuracy (Alzubi, 2015).

In literature it has been shown theoretically and proved through numerical result that classifier ensembles are efficient if and only if the base classifiers are of relatively high accuracy and don't make simultaneous errors (Giacinto and Roli 2001a). Ji and Ma (1997) introduced an algorithm for combining weak classifiers (with an accuracy just little over 50%) in order to obtain a classification system whose generalization and efficiency are both good. The idea behind their algorithm is how to properly select the strength of the weak classifiers. Also a construction of multi version systems of artificial neural networks by calculating a diversity measure was discussed by Partridge and Yates (1995). A similar approach also has been adopted by Sharkey (1996) where an ensemble of artificial neural net works to be created in which nets exhibits no coincident errors. Furthermore, (Kuncheva *et al*., 2000) proved by artificial example that combining negatively dependent classifiers are preferable over independent ones and their combination will guarantee a better result.

The just mentioned theoretical and empirical results agree on the role of diversity in improving the accuracy and effectiveness of classifier ensembles. However, this agreement is a companied by the observation that constructing an ensemble that is accurate and diverse is not an easy task (Partridge and Yates, 1995; Sharkey, 1996). This difficulty emerges from the fact that in real applications, classifiers tend to make the same errors. In other words, the yaccurately classified simple tasks and identically misclassified complicated ones. This motivates the need for a new method that takes both accuracy and diversity into consideration at the ensemble design level (Giacinto and Roli, 2001b). However, much of the research effort in the field of classifier ensembles has focused on the combination methods and little attention paid to the design of classifier ensembles.

In this study, an approach to the automatic design of classifier ensemble formed by different types of classifiers is proposed. The aim of our approach is to choose the most effective subset of classifiers. The proposed approach is different from any previous work in this field as it uses cooperative game theory (in particular coalition formation) in the design stage of classifier ensembles. In addition, we will use Cohen's Kappa as a measure of diversity between base classifiers and will compute the diversity for the whole ensemble in order to estimate the contribution of each base classifier.

LITERATUR REVIEW

Many efforts have been made to find the optimal architecture of classifier ensembles. In the literature (Giacinto and Roli, 2001a; Sharkey, 1996), several methods have been suggested on how to create ensembles of neural networks in which they don' tmake the same errors. In general, the main idea behind such methods is to vary the parameters that specifically related to neural networks design and training. So, it is not possible to generalize these methods to different types of classifiers.

It has been found by Partridge (1996) that in order to create a diverse ensemble of networks- that are making different errors three important parameters should be tuned: training set, weight-space and number of hidden nodes. Partridge (1996) establishes a basis for engineering a diversity approach which started to be investigated by networks design researchers. Most of the work in that area falls under one of the following two categories: the "direct" and the "overproduce and choose."

In the former one, the objective is to have ensembles containing error-diverse nets directly. An example of this category is work done by Opitz and Shavlik (1996) where they present an algorithm called "ADDEMUP" which explicitly search for the highly diverse set of accurate trained nets to form ensembles. On the other hand, the idea behind the "overproduce and choose" category is firstly to produce a pool or a large set of nets followed by step of selecting the most error-diverse and accurate subset. However, many studies advocate the previous approach. Aksela (2003) discussed several methods to be used in the selection stage of classifiers. He concluded that the best two methods are the one based on penalizing classifiers that make the same error and the one that uses exponential error count as criterion.

Many algorithms and methods have been introduced to achieve the just mentioned two approaches. It seems the main difference that distinguishes these algorithms is the way in which they calculate the diversity between the classifiers. In other words, the difference is based on the diversity measure used and how it is computed. Some of these algorithms apply a pair-wise diversity measure that only consider two classifiers at a time then average across all pairs to get a single diversity value. Other algorithms consider all classifiers in the ensemble and calculate one value for diversity.

Giacinto and Roli (2001a) created a diversity matrix to represent the pair-wise diversity for the classifiers pool then select the most diverse ones. They applied *double faultmeasure* and *Q statistics* (Roli *et al*., 2001) as shown in the following equations respectively (Kuncheva, 2004):

$$DF_{i,j} = d \quad (1)$$

$$Q_{i,j} = \frac{ad-bc}{ad+bc} \quad (2)$$

where, d represents the probability of both classifiers being incorrect.

The selection process aims to obtain a desired number of classifiers by applying a search method that considers two pairs of classifiers at a time and choose the least related classifiers. The just mentioned approach is called "the most diverse ensemble."

A new diversity measure has been proposed by Banfield *et al*. (2003) which is called Percentage Correct Diversity Measure (PCDM). It takes into account only the proportion of classifiers who correctly classify an object. They declare "uncertainty points" as the data points where the correct votes fall between 0.1 and 0.9. However, these points vary based on which base classifiers are selected to build the ensemble.

Also Banfield *et al*. (2003) introduced their algorithm "thinning the ensemble" which is based on their proposed measure PCDM. It basically follows the backward selection process where it starts with a pool of classifiers then removes the classifier that is most often incorrect on the uncertainty points. It repeats this procedure until the predetermined number of classifiers is reached. They claimed that "thinning the ensemble" algorithm reduces the size without compromising the accuracy of the ensemble.

Another different approach presented by Giacinto and Roli (2001a) which is "clustering and selection" method. It is obvious from the name that it consists of two main steps. The first one is forming clusters of classifiers and the second is choosing the most accurate classifier from each cluster. This approach uses the double fault measure of diversity to create the pair-wise diversity matrix which is called the *distance matrix.*

Our proposed approach is different from the above methods as it calculates the diversity for the entire ensemble and then computes the diversity contribution of each individual base classifier which makes it possible to select only the classifiers with the most contribution. An empirical comparison between our algorithm and the three aforementioned approaches is conducted and the results are presented later.

Coalition-based ensemble design algorithm: The rationale behind this algorithm is that we believe "direct" creation of ensemble from error-independent classifiers is a complicated task and can be computationally very expensive. This opinion appears to be shared by other researchers in the classifier ensembles field (Giacinto and Roli, 2001a; Partridge and Yates, 1995; Giacinto and Roli, 2001b):

Algorithm 1: Coalition Based Ensemble Design Algorithm
Input:
L: is set of classifiers
λ: Contribution value threshold

d: The maximal permutation size for calculating the contribution values
k: the number of classifiers selected in each phase
Output:
Optimal Coalition S where S ∈ L
1: initialize S: = φ
2: for each l ∈ L\S do
3: C_l: = Diversity_Contribution (l, S, d)
4: end for
5: if max C_l > λ then
6: S: = S ∪Classifiers_Selection ({C_f}, k, λ)
7: goto 2
8: else
9: return S
10: end if

In our algorithm, we will transform the game theory concepts into the arena of ensemble design (more precisely to the computation of contribution phase of the algorithm), in which one attempts to estimate the diversity contribution of each classifier in generating an ensemble. The players *N* are represented by the candidate classifiers and the "payoff" is mapped to a real-valued function ν(S), which measures the diversity of the ensemble generated using the set of classifiers *S*.

Algorithm 1 presents the pseudo code of CED. It is obvious that CED has an iterative nature and it adopts a forward selection approach. In addition, it is clear that CED consists of two main processes which are represented by functions in our implementation code. The first one is the *diversity_contribution* which performs sorting of each candidate classifier according to its diversity contribution value. The second main process is the *Classifier_Selection* which mainly selects a specific number of classifiers k with the highest diversity contribution values. The process of computing the diversity contribution values will be repeated for the remaining candidate classifiers given those already were selected.

The process of selecting a new classifier also will be repeated as long as the candidate classifier's diversity contribution is larger than the threshold λ.

The first glance at algorithm 1 without further exploring the details of diversity contribution, one might get the impression that this algorithm is a generalization of filter method. However, the main idea of CED is that the *Diversity_contribution* function, unlike any other filter methods, returns a contribution value for each classifier according to its assistance in increasing the ensemble diversity and in conjunction with other classifiers.

The natural question to ask here is, how to calculate the contribution of each classifier to the coalition? Game theory provides the answer by constructing a value function, which assigns a real value for each classifier in the coalition. The value represents the contribution of the classifier in achieving a high payoff. Shapley value provides a way to calculate each classifier contribution.

Definition: A coalitional game (*N*, *v*), the Shapley value of player *i* is given by (Cohen *et al*., 2005):

$$\Phi_i = \frac{1}{N!}\sum_{S\subseteq N\{i\}} |S|\,!\,(\,|N| - |S| - 1)!\,[v(S \cup i) - v(S)] \quad (3)$$

Equation (3) computes the average marginal contribution of player *i*, where it averages over all the different permutations according to which the grand coalition might be built from the empty coalition. It is obvious from Eq. (3) that the calculation of Shapley value requires summing over all possible subset of player. However, if the number of players is larger than these calculations become computationally intractable and in this case an approximation is required. Keinan *et al*. (2004) proposed an estimator which uses uniformly sampled subsets instead of the full set of subsets. However, a further reduction on the size of the subsets suggested by Cohen *et al*. (2005) where they bounded the permutation size into some constant *d* where *d* is less than *N*. Now, the Shapley value calculation equation with d-bounded permutations will be:

$$\varphi_i = \frac{1}{|\Pi_d|}\sum_{S\subseteq N\{i\}} |S|\,!\,(\,|N| - |S| - 1)!\,[v(S \cup i) - v(S)] \quad (4)$$

Π_d Denotes the set of d-bounded permutation, where *d* represents the number of interactions to be considered between classifiers. For instance when *d* =1, it means classifiers to be considered independently and no interactions between them matter.

By applying the cooperative game theory notations, more specifically referring to Eq. (4), the function *Diversity_Contribution* computes the d-bounded estimated diversity contribution values.

It should be kept in mind here that the goal is to optimize the overall diversity level of the suggested ensemble. The *Diversity_Contribution* function performs its calculations of contribution values based on the following payoff function *v* (*S*):

- **Train:** Generate an ensemble and train on the training set of the data.
- **Validation:** Classifying the validation set data.
- Return the diversity level defined as *v* (*S*) which can be calculate by finding Kappa Cohen measure for multi base classifiers (Cohen, 1960; Berry and Mielke Jr., 1988; Landis and Koch, 1977).

Now, let us explain the effect of varying the values of the parameters that were utilized in the CED algorithm:

- In each iteration of CED the number of selected classifier s*k* for the *Classifiers_Selection* function controls the redundancies of the selected classifiers. If we would like to increase the chance of classifiers with similar diversity contribution to be selected then we will set k to a higher value and vice versa. In our experiments, we chose $k = 1$ which means that we minimize the redundancy dependencies of the classifiers. It is evident that k has a direct relationship with the convergence of the CED. Decreasing k will cause the CED's convergence to be slow.
- Another parameter that must be tuned is λ as it represents a trade-off between the number of selected classifiers and the overall diversity of the ensemble. A high value of λ has direct effect on the ensemble size which that CED selects a small sets of classifiers. Setting $\lambda = 0$ yields that CED will choose classifiers as long as there exist a classifier that is likely to increase the diversity of the ensemble, as λ increases a smaller set of classifier will be selected. For this study experiments, λ value has been set to zero which means that any base classifier that has positive contribution to ensemble diversity will be has the chance to be selected. It is obvious that the halting criterion of CED depends on λ.
- The maximal permutation size d plays an important role in determining the diversity contribution values of different classifiers. When choosing the value of d must take into considerations that different classifiers combinations are represented.

EXPERIMENTAL DATA

Performance of the algorithms (CED, Classic and Clustering) is evaluated by running experiments on 15 representative data sets from the UCI repository (Newman *et al*., 1998). These data sets were used in similarstudies (Webb, 2000; Quinlan, 1996).

Table 1 presents a summary of these data sets. When choosing these data sets we took into consideration that a binary and multi class data sets to be included. In addition, a variation in the number of the attributes and examples (data items) are also considered.

Two sets of experiments were conducted in order to compare the performance of *CED* algorithm with clustering algorithm and the classic ensemble. The results of the latter one served as the baseline in our experiments. In each set of the of the experiments *CED*, Clustering and classic ensemble design were compared on 15 data sets using the MATLAB implementation of these algorithms.

In classic ensemble design all base classifiers are included in the ensemble, whereas in the Clustering ensemble the target ensemble size must be predetermined in advance. In our experiments the target ensemble size for these algorithms was set to 5 which we find out experimentally to be the most accurate ensemble size. Our algorithm (CED) bypasses this requirement by setting the desired ensemble diversity value instead of setting the target size of the ensemble. CED may terminate with a smaller ensemble size if the number of iterations generates a diverse ensemble that exceeds the specified value. However, CED could be easily adjusted to return a predetermined number of classifiers if we desire. This is one of the advantages of our approach which will be demonstrated latter in the results section of this study.

Table 1: Summary of data sets

Name	Examples	Classes	Attributes
Blog spam	56000	2	547
Breast cancer	699	2	9
Letter recognition	20000	26	16
Iris	150	9	4
Segment	2310	7	19
Ionospere	315	2	34
Statle (vehicle silhouetts)	946	4	18
Haberman's survival	946	2	3
Contraceptive method choice	1473	2	3
Isolet	1559	26	617
Glass	214	6	9
Colic	368	2	22
Heart-c	303	2	13
Splice	3190	3	62
Anneal	898	6	38

Table 2: Summary of base classifiers

Name of classifies	Abbreviation
k-nearest neighbor classifier	knnc
Binary decision tree classifiers	treec
Naïve bayes classifier	naivebc
Support vector classifier	svc
Normal densities based linear classifier	ldc
Linear perceptron	perlc
Normal densities based quadratic classifier	qdc
Logistic linear classifier	loglc
Train neural network classifier by back-propagation	bpxnc
Nearest mean classifier	nmc
Train radial basis neural network classifier	rbnc
k-centers clustering	kcentres
Radial basis SV classifier	rbsvc
Parzen classifier	parzenc
Minimum least square linear classifier	fisherc

To ascertain that no algorithm was being disadvantaged by small number of base classifiers, we ran our experiments with 15 classifiers for all algorithms. In addition, the type of classifiers used in all experiments was the same. Table 2 shows the list of the classifiers that were implemented (Duin *et al*., 2007).

For the purpose of comparing *CED* with other algorithms across all domains, we implemented statistics used in (Webb, 2000), specifically the win/draw/loss record and the geometric mean error ratio. The simple win/draw/loss record computed by calculating the number of data sets for which *CED* obtained better, equal, or worse performance than any of the other algorithm with respect to the ensemble

classification accuracy. In addition to that, we computed another record representing the statistically significant win/draw/loss, according to this record win/loss is only computed if the difference between two values is greater than 0.05 level which was determined to be significant by computing the student paired t-test.

EXPERIMENTAL RESULTS

Table 3 and show the performance results of the *CED* compared to classic ensemble and Clustering ensemble designs. In our experiments, the training set sizes are varied from10 to 40% of the total available data. Those points on the learning curve are chosen because it is expected that the most difference in the performance of the algorithms will occur at these points.

In each table, the row represents the data set and the column represents the percentage of the data that is used for training. Each cell entry contains two numbers; the number on the left refers to the accuracy of CED while the number on the right refers to the accuracy of the algorithm under consideration. If the difference is statistically significant, which is determined by calculating the student paired t-test and the confidence intervals at the 0.05 level and then the larger of the two is shown in bold. The bottom of each table contains a summary of the statistics for the various points on the learning curve.

Figure 1 and 2 are scatter-plots representing the results showed in the Table 3. Each plot shows a comparison between *CED* and another algorithm for one point on the learning curve. Each one of the 15 data sets is denoted by a point on the scatter plots. If the point is located above the diagonal then it means that the accuracy of CED is higher than the accuracy of the algorithm under comparison, a point on the diagonal indicates that both accuracies are equal, otherwise the accuracy of *CED* is lower.

The general trend that is noticed when performing these experiments is the fact that the accuracy of any ensemble design outperforms the best individual base classifier.

The results in Table 3 confirm the fact that the ensemble accuracy is correlated with its diversity. So increasing the diversity between base classifiers yields an increase in the ensemble diversity. The table shows that *CED* outperforms the Classic Ensemble at all data sets using various training sizes and it also illustrated by the scatter-plot in Fig. 1 where it shows all points are above the diagonal.

Table 3: CED Vs. classic ensemble

Date set	10%	20%	30%	40%
Breast cancer	90.26/85.31	94.48/90.93	95.36/92.28	95.73/93.43
Lonosphere	88.64/82.40	89.55/82.94	89.91/83.57	90.20/86.51
Auto (statlog)	70.20/59.21	75.74/68.10	77.84/69.63	77.17/71.78
Haberman's survival	67.23/60.04	76.77/67.00	80.00/69.24	81.17/73.68
Contraceptive	45.89/37.40	55.20/46.95	62.86/51.23	67.77/55.45
Blog spam	91.83/85.37	93.88/88.69	94.94/90.50	96.47/91.27
Letter recognition	86.53/74.23	88.20/75.49	89.20/80.20	89.73/82.95
Iris	86.53/74.33	88.24/81.33	90.47/83.73	91.84/85.01
Segment	85.48/79.39	86.70/80.22	87.81/82.16	85.02/80.52
Isolet	81.54/74.20	83.33/76.16	84.47/79.68	85.02/80.52
Glass	53.40/44.52	59.77/5016	64.01/55.23	66.07/57.88
Colic	74.05/61.94	76.66/65.71	77.34/67.33	78.09/69.12
Heart-c	71.06/61.17	83.14/69.54	86.27/72.10	88.22/78.38
Splice	75.42/64.41	84.09/70.13	87.62/77.06	90.46/81.57
Anneal	87.09/74.89	88.38/78.51	88.68/79.04	89.22/80.95
Win/draw/loss	15/0/0	15/0/0	15/0/0	15/0/0
sig. W/D/L	15/0/0	15/0/0	15/0/0	15/0/0
GM error ration	0.6799	0.6365	0.6253	0.6405

Table 4: CED Vs. clustering algorithm

Date set	10%	20%	30%	40%
Breast cancer	90.26/87.87	94.48/93.67	95.36/94.89	95.73/95.34
Lonosphere	89.64/86.21	89.55/87.09	89.91/87.36	90.20/87.85
Auto (statlog)	70.20/67.21	75.74/69.79	77.84/72.58	77.17/73.53
Haberman's survival	67.23/61.77	76.77/67.07	80.00/71.68	81.17/72.17
Contraceptive	45.89/44.22	55.20/53.53	62.86/56.92	67.77/62.68
Blog spam	91.83/88.34	93.88/91.20	94.94/92.37	96.47/94.34
Letter recognition	86.53/79.33	88.20/86.33	89.20/87.73	89.73/87.81
Iris	86.55/86.58	88.24/88.24	90.47/90.93	91.84/91.97
Segment	85.48/84.25	86.70/84.61	87.81/84.63	85.02/85.35
Isolet	81.54/81.94	83.33/84.97	84.47/85.41	85.02/85.55
Glass	53.40/44.62	59.77/52.16	64.01/58.63	66.07/59.38
Colic	74.05/61.94	76.66/72.44	77.34/73.61	78.09/73.78
Heart-c	71.06/61.17	83.14/72.49	86.27/79.59	88.22/82.78
Splice	75.42/64.41	84.09/74.59	87.62/79.84	90.46/84.11
Anneal	87.09/74.89	88.38/84.49	88.68/85.59	89.22/86.41
Win/draw/loss sig.	13/0/2	13/1/1	13/0/2	14/0/2
W/D/L	11/4/0	12/2/1	12/2/0	12/3/0
GM error ration	0.8386	0.8137	0.8020	0.8006

The results in III are expected based on the fact that *CED* algorithm selects the optimal number of classifiers with the highest diversity contribution. These results are encouraging as the classic or traditional Ensemble constitutes the base line or reference ensemble for our further comparisons. The results also affirm the scalability of *CED* design concept.

Again by looking at Table 3, we could see that the CED outperforms the classical Ensemble in every entry of the table.

This means that the gain in accuracy is always statistically significant. The statistics at the bottom of the table further confirm these results. In particular, the geometric mean error ratio values show the higher magnitude of the gain in all percentages of training data sets sizes.

Now the experiment moves into comparing the *CED* with the clustering algorithm. From Table 4 it is clear that the *CED* outperforms the Clustering algorithm on almost all data sets and on most training sizes. Moreover, the gain in accuracy is significant in most of the training sizes, in particular, when training sizes and the accuracy are low and that is supported by the values of the geometric mean error ratio. This is stemming from the fact that training data sizes are high and the base classifier errors are low yielding less diverse ensembles.

The scatter plot in Fig. 2 represents the comparisons between *CED* and Clustering algorithm over various sizes of training data sets. Figure 2 shows few points are slightly under the diagonal line which suggests that even in cases where Clustering beats *CED* the gain is significantly less than *CED's* gain over Clustering on the rest of the data sets and training data sizes.

Again, regarding Clustering Ensembles case, the number of selected classifiers by the *CED* across all datasets and all domains is always less than or equal to the number selected by the Clustering Ensemble. This is very imperative advantage that facilitates the reliability of the *CED* design in ensuring high performance and suitability to the data set in question and application used. Interestingly, the performance gain difference is observed to be higher when the performance of the Clustering Ensemble is in the low accuracy ranges. For example, in the Glass data set at 10 and 20 training sizes, the CED performance gain is 8.8 and 7.61, respectively compared to 3.49 and 2.68

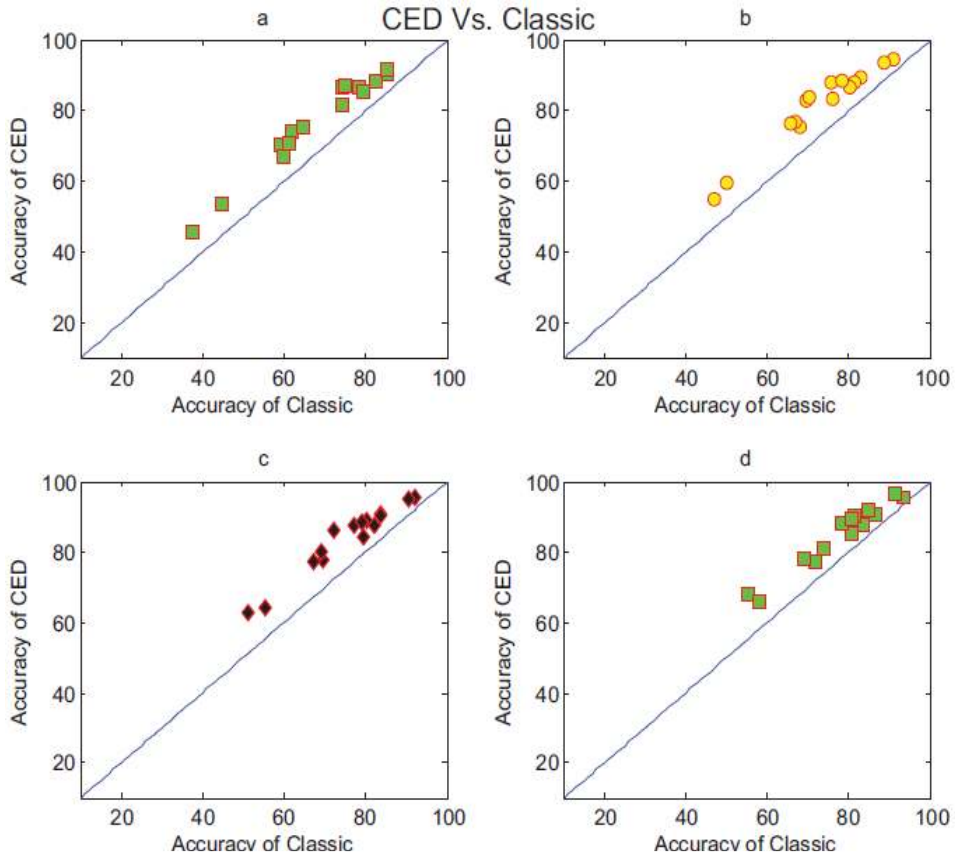

Fig. 1: Computing the performance of CED with classic ensemble on 15 data sets given: a) 10% b) 20% c) 30% d) 40% of the data for training

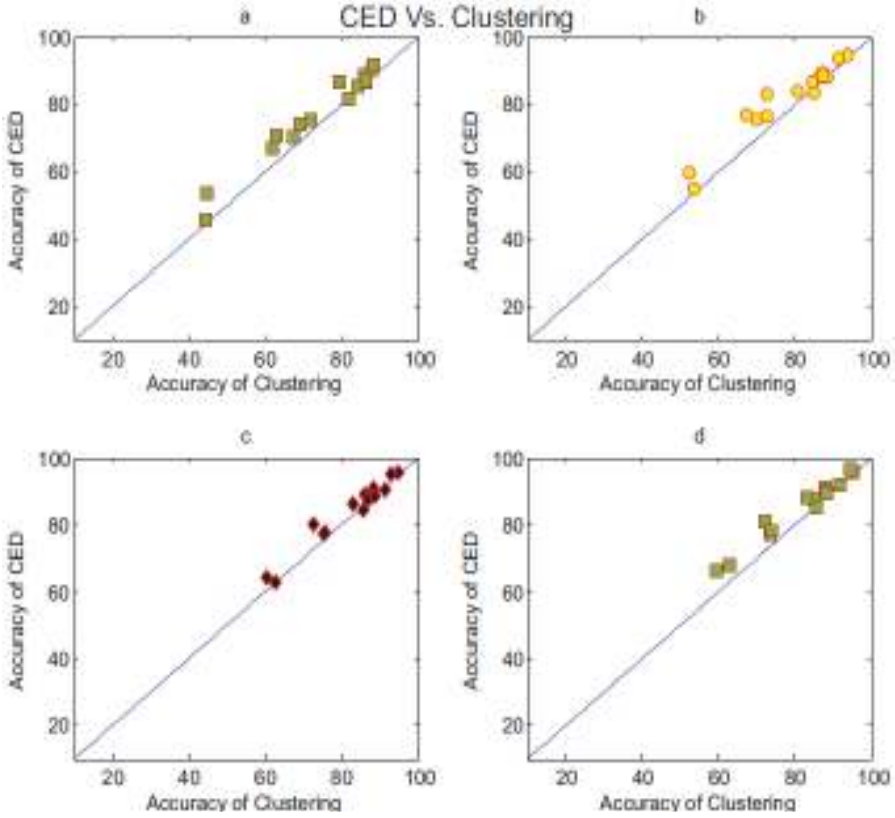

Fig. 2: Computing the performance of CED with clustering algorithm on 15 data sets given: a) 10% b) 20% c) 30% d) 40% of the data for training

for the Clustering Ensemble for the blog Spam data set at the same training sizes. Looking at other data sets, the same trend can be observed.

CONCLUSION

It is worth mentioning here that it was not our aim throughout this study to establish a theoretical link between diversity and overall accuracy of ensemble in the classification tasks, nor to prove this relationship as it has been established and validated by many previous studies in this field. Our goal was to develop a technique that takes advantage of these facts and optimizes its applicability in the design process of ensembles.

In this study we proposed *CED*, a new algorithm for creating classifier ensembles. *CED* differs from other ensemble methods such as bagging and boosting in that it explicitly tries to foster ensemble diversity. There have been many approaches that used diversity to guide the ensemble creation. Here we compared *CED* with Clustering algorithm.

The effectiveness of the *CED* algorithm was investigated and exploited as a type of ensemble designing technique that encourages diversity explicitly. The goal of our algorithm was to find the optimal coalition in the set of base classifiers in the ensemble based on their value of contribution in the overall ensemble diversity.

Extensive experiments with a wide variety of data sets have demonstrated that CED effectively captures the diversity contribution of base classifiers and particularly allows us to identify the base classifiers that significantly improve the ensemble diversity. In particular, we showed that when CED is used, it produces substantial improvements over using other ensemble approaches such as the ones mentioned above.

ACKNOWLEDGMENT

The author of this study would like to deeply thank Dr.Omar Alzubi at Al-Balqa Applied University and Professor Thomas Chen at City University London for their in valuable advices.

REFERENCES

Aksela, M., 2003. Comparison of classifier selection methods for improving committee performance. Proceeding of the 4th International Conference on Multiple Classifier Systems, (MCS'03), pp: 84-93.

Alzubi, J., 2015. Diversity based improved bagging algorithm. Proceeding of the International Conference on Engineering and MIS 2015, Istanbul-Turkey.

Banfield, R.E., L.O. Hall, K.W. Bowyer and W.P. Kegelmeyer, 2003. A new ensemble diversity measure applied to thinning ensembles. Proceeding of the 4th International Workshop on Multiple Classifier Systems, pp: 306-316.

Berry, K.J. and P.W. Mielke Jr., 1988. A generalization of Cohen's Kappa agreement measure to interval measurement and multiple raters. Educ. Psychol. Meas., 48(4): 921-933.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas., 20(1): 37.

Cohen, S.B., G. Dror and E. Ruppin, 2005. Feature selection based on the Shapley value. Proceeding of the IJCAI.

Duin, R., P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax and S. Verzakov, 2007. PR-Tools4.1, a matlab toolbox for pattern recognition. Delft University of Technology, 2007.

Giacinto, G. and F. Roli, 2001a. An approach to the automatic design of multiple classifier systems. Pattern Recogn. Lett., 22: 25-33.

Giacinto, G. and F. Roli, 2001b. Design of effective neural network ensembles for image classification purposes. Image Vision Comput., 19(910): 699-707.

Ji, C. and S. Ma, 1997. Combinations of weak classifiers. Trans. Neur. Netw., 8(1): 32-42.

Keinan, A., B. Sandbank, C.C. Hilgetag, I. Meilijson and E. Ruppin, 2004. Fair attribution of functional contribution in artificial and biological networks. Neural. Comput. 16(9): 1887-915.

Kuncheva, L., C. Whitaker, C. Shipp and R. Duin, 2000. Is independence good for combining classifiers? Proceeding of the 15th International Conference on Pattern Recognition.,

Kuncheva, L.I., 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, Hoboken.

Landis, J.R. and G.G. Koch, 1977. A one-way components of variance model for categorical data. Biometrics, 33(4): 671-679.

Newman, D., S. Hettich, C. Blake and C. Merz, 1998. UCI repository of machine learning databases. Department of Computer Science, University of California, Irvine. Retrieved from: http://www.ics.uci.edu/mlearn/MLRepository.html.

Opitz, D. and J. Shavlik, 1996. Actively searching for an effective neural-network ensemble. Connect. Sci., 8(3): 337-353.

Partridge, D. and W.B. Yates, 1995. Engineering multiversion neural-net systems. Neural Comput., 8: 869-893.

Partridge, D., 1996. Network generalization differences quantified. Neural Networks, 9(2): 263-271.

Quinlan, J.R., 1996. Bagging, boosting and C 4.5. Proceeding of the 13th National Conference on Artificial Intelligence, AAAI Press, pp: 725-730.

Roli, F., G. Giacinto and G. Vernazza, 2001. Methods for designing multiple classifier systems. Proceeding of the 2nd International Workshop on Multiple Classifier Systems (MCS '01). Springer-Verla, London, UK, pp: 78-87.

Sharkey, A.J.C., 1996. On combining artificial neural nets. Connect. Sci., 8: 299-313.

Webb, G.I., 2000. Multiboosting: A technique for combining boosting and wagging. Mach. Learn., 40(2): 159-196.