Solving Multiclass Classification Problems using Combining Complementary Neural Networks and Error-Correcting Output Codes

Somkid Amornsamankul, Jairaj Promrak, Pawalai Kraipeerapun

***Abstract*— This paper presented an innovative method, combining Complementary Neural Networks (CMTNN) and Error-Correcting Output Codes (ECOC), to solve multiclass classification problem. CMTNN consist of truth neural network and falsity neural network created based on truth and falsity information, respectively. Two forms of ECOC, exhaustive code and random ECOC, are considered to deal with k-class classification problem. Exhaustive code is applied to the problem with $3 \leq k \leq 7$ whereas random ECOC is used for $k > 7$. In the experiment, we deal with feed-forward backpropagation neural networks, trained using 10 fold cross-validation method and classified based on two decoding techniques: minimum distance and $T > F$. The proposed approach has been tested with six benchmark problems: balance, vehicle, nursery, Ecoli, yeast and vowel from the UCI machine learning repository. Three data sets: balance, vehicle and nursery are dealt with exhaustive code while random ECOC is applied for Ecoli, yeast and vowel. It was found that our approach provides better performance compared to the existing techniques considering on either CMTNN or ECOC.

***Keywords*— Multiclass classification problem, Neural network, Feed forward backpropagation, Complementary neural networks, Error-correcting output codes, Exhaustive codes

I. INTRODUCTION

CLASSIFICATION is the problem of mapping a vector of observed characteristics into a defined class. It can be separated into two types which are binary classification and multiclass classification. In binary classification, only two classes are involved whereas, in multiclass classification, there are more than two defined classes [1]. In order to solve multiclass classification problem, the most successful and popular method is Neural Networks (NNs) [2].

The significant advantages of neural networks can be represented in various aspects. They can be considered as data driven self-adaptive methods, universal functional approximators which use arbitrary accuracy to approximate any function, and also nonlinear models flexible in modeling real world complex relationships [3], [4], [5]. Moreover, neural networks have been successfully applied to many branches: signature recognition [6], [7], medicine [8], and business failure prediction [9], [10]. The performance between neural networks and other classifiers are widely studied as well. Kim presented comparison of decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size. In his study, artificial neural network was superior for two or more continuous and categorical independent variables [11]. Eskandarinia *et al.* studied two models based on neural networks and k-nearest neighbor in daily flow forecasting. They found that neural networks model provides better result than nearest neighbor method [12]. Razi and Athappilly presented comparison of prediction accuracy including nonlinear regression, neural networks (NNs), as well as Classification and Regression Tree (CART) models. In their study, NNs and CART models provided better result than nonlinear regression model in which applying NNs produced lower values of Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) than using CART [13]. Three models based on discriminant analysis, logistic regression and neural network used to determine sex from the upper femur are built and compared [14]. The consequence is that neural network can correctly classify more than discriminant analysis and logistic regression. These are why neural networks are applied in this work. Although several types of neural networks can be used for classification purposes [15], our focus is on the feed-forward neural network since it has been widely studied and used to solve the classification problem.

To improve the percentage of accuracy prediction for classification problems, Complementary Neural Networks (CMTNN) based on feedforward neural network have been proposed [16], [17]. This model uses both truth and falsity information as an input information while traditional neural network requires only truth information. More detail about CMTNN will be described in section II. Apart from this, some techniques which are codeword designs and decoding can also be used to increase the model's efficiency [4].

In designing code, Error-Correcting Output Codes (ECOC) are suggested to deal with multiclass classification problem since this method can reduce the variance of learning algorithm which leads to ability in correcting errors [18]. It was also found that ECOC can improve generalization capabilities of classification systems [19]. In section III, the ways to construct ECOC are explained.

In this research, we will apply both powerful techniques, ECOC and CMTNN, to solve multiclass classification problems which can be described as follows.

S. Amornsamankul is with Department of Mathematics, Faculty of Science, Mahidol University, and is with Centre of Excellence in Mathematics, CHE, Bangkok, Thailand (email: scsam@mahidol.ac.th)

J. Promrak is with Department of Mathematics, Faculty of Science, Mahidol University, Bangkok, Thailand (email: jpjairaj@gmail.com)

P. Kraipeerapun is with Department of Computer Science, Faculty of Science, Ramkhamhaeng University, Bangkok, Thailand (email: pawalai@yahoo.com)

II. Complementary Neural Networks

In order to solve the classification problem using neural network, on the whole, we deal with binary values, 0 and 1, representing the truth information. However, it is not exactly true. Hence, degrees of truth must be considered. For degree of truth, its value is in the set $[0, 1]$. Instead of considering only the truth information, the complement of the truth which is the falsity information should also be considered since the predicted output may not exactly true. Therefore, the truth neural network and the falsity neural network are created in order to predict the truth output and the falsity output, respectively. These two outputs are predicted in the sense that they should be complement to each other. If both output values are similar then it is an indicator that we may have to readjust parameters of neural networks. The combination of these two networks is called complementary neural networks (CMTNN). Both truth and falsity neural networks are created based on the implication rules shown in TABLE I.

TABLE I
Implication rules

Type of NN	Input	Target	Output (Inference)
Truth NN	True	True	True
Falsity NN	True	False	False

From the table, the logical implication "if X then Y ($X \rightarrow Y$)" is applied. If we know that X and Y are true, we then get its inference also true. On the other hand, if X is true but Y is false, then its inference is false. In the training phase of CMTNN, X and Y are considered as the input feature and the target value, respectively. The inference is considered as the predicted output.

Suppose that we have n patterns, each with m features, and we want to classify patterns into k classes. For each pattern, let x_i be the input pattern of i^{th} sample where $i = 1, 2, 3, ..., n$, $t(x_i)$ be the truth target value, $f(x_i)$ be the falsity target value, $T(x_i)$ be the truth output value, $F(x_i)$ be the falsity output value. Notice that the relationship between the truth target value and its complement which is the falsity value is

$$f(x_i) = 1 - t(x_i). \tag{1}$$

For example, if

$$t(x_i) = (10000, 01000, 00100, 00010, 00001),$$

then its complement is

$$f(x_i) = (01111, 10111, 11011, 11101, 11110).$$

The process of solving multiclass classification using complementary neural networks is shown in Fig. 1.

After the input patterns, the truth and falsity target values are entered, the truth and falsity neural networks are trained to predict degrees of truth and falsity output value separately. After that, both trained networks can be used to predict the unknown input pattern. The truth and falsity output obtained from both trained networks can be aggregated to form the final output which is then classified into one of the k classes.

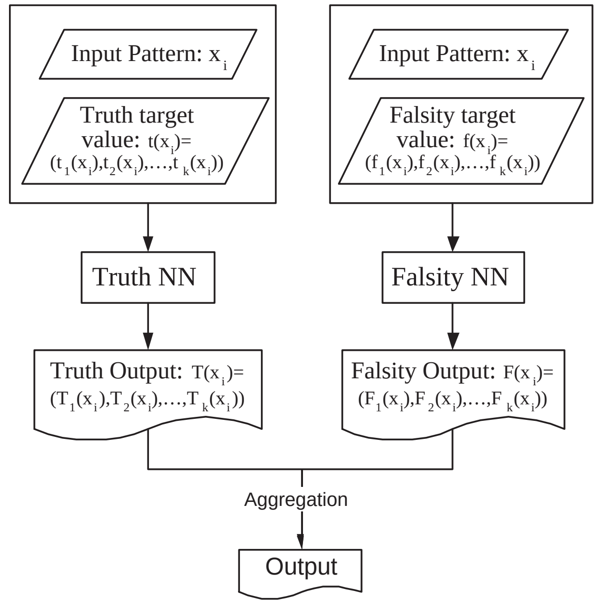

Fig. 1. Complementary neural networks model

III. Error Correcting Output Codes

ECOC is an information theoretic concept used to correct errors when transmitting data in communication tasks. The idea of these codes is adding some redundant cases which do not match with any acceptable solution in the output set. If one of these cases appears after data is transmitted, the system will realize occurence of the error. In classification process, vector of features (patterns) is transmitted into the set of defined classes where classes are represented by codewords. ECOC is believed to improve classification's performance by dividing a multiclass problem into binary-class sub-problems and correcting errors in the decision making stage as well. Dietterich and Bakiri presented three forms of ECOC based on the number of class (k) which are exhaustive codes for $3 \leq k \leq 7$, column selection from exhaustive code for $8 \leq k \leq 11$, and randomized hill climbing and Bose-Chaudhuri-Hocquengham (BCH) codes when $k > 11$ [20]. Besides, random generation of the codewords is a recommended method [20], [21].

In this work, we focus on exhaustive code and random ECOC, obtained from random generation method, which can be explained as follows.

A. Exhaustive Codes

For data set having k classes when $3 \leq k \leq 7$, a code of length $2^{k-1} - 1$ can be constructed where

class 1 : All strings are one,

class 2 : There are 2^{k-2} zeroes followed by $2^{k-2} - 1$ ones,

class 3 : There are 2^{k-3} zeroes, followed by 2^{k-3} ones, followed by $2^{k-3} - 1$ zeroed, followed by $2^{k-3} - 1$ ones.

$\vdots$ $\quad\quad\quad\quad$ $\vdots$

class i : There are alternating runs of 2^{k-i} zeroes and ones [20].
For example, when $k = 4$; length of code has $2^{4-1} - 1 = 7$ digits where class 1 is 1111111, class 2 has $2^{4-2} = 4$ zeroes and $2^{4-2} - 1 = 3$ ones, class 3 has $2^{4-3} = 2$ zeroes, $2^{4-3} = 2$ ones, $2^{4-3} = 2$ zeroes and $2^{4-3} - 1 = 1$ one, and class 4 has $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ ones, $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ one, $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ ones, and $2^{4-4} = 1$ zeroes; see TABLE II.

TABLE II
EXHAUSTIVE CODES WHEN $k = 4$

Class	Codeword
1	1 1 1 1 1 1 1
2	0 0 0 0 1 1 1
3	0 0 1 1 0 0 1
4	0 1 0 1 0 1 0

B. Random Error-Correcting Output Code

The constuction for the randomly generated ECOCs can be described as follow.

Let C be a codeword matrix with size $k \times L$ where k is the number of class and L is a length of codeword. Each element in the matrix is set to either 0 or 1 randomly. There are three main issues for design codeword matrix C including

- how many used columns or codeword length (L),
- distance between rows, and
- distance between columns.

On average, when the length is increased, Hamming distance between any pairs of codewords will increase in which longer codewords might provide almost optimal code. However, it was found that beneficial effect of optimal code is decreased comparing with random code when code's length is increasing [22]. For convenience in decoding, $L = 15$ is selected. Let H_{k_i} be the minimum distance of any pair of rows in the matrix C_i and H_{L_i} be the minimum distance of any pair of columns in the matrix C_i for $i = 1, 2, 3, \ldots$,500. From theory of error-correcting codes, the code matrix C_i can be corrected up to $\lfloor \frac{H_{k_i}-1}{2} \rfloor$ errors [18]. In other words, minimum Hamming distance determines error-correcting ability. For two code matrices with the same length, matrix having more error-correction ability performs better [22]. Thus, in order to avoid misclassifications, the Hamming distance between every pair of codewords should be as large as possible. Since distance between columns determines independence of base classifier, maximum Hamming distance between any pair of column is required.

In our experiment, the best codeword matrix from random 500 times is applied. The selected codeword matrix C must provide the maximum of the sum $H_k + H_L$ [23], i.e.,

$$C = \{C_j \mid H_{k_j} + H_{L_j} = \max_i \{H_{k_i} + H_{L_i}\}\} \qquad (2)$$

for $i = 1, 2, 3, \ldots, 500$. The examples of random ECOC for $k = 8$ is shown in the TABLE III.

TABLE III
RANDOM ECOC WITH $k = 8$, $L = 15$ AND $H_k + H_L = 17$

Class	Codeword
1	0 0 1 1 1 0 0 1 0 0 0 0 0 0 0
2	0 0 1 0 0 0 1 0 0 0 1 1 0 0 1
3	0 1 1 0 0 1 0 0 0 1 1 0 0 0 1
4	1 1 1 1 0 1 1 1 1 0 1 1 0 0 1
5	0 1 1 0 1 1 0 1 1 0 0 0 0 0 1
6	1 1 1 0 0 1 0 1 1 0 1 1 0 1 0
7	1 1 1 0 0 1 1 0 1 0 0 1 1 0 1
8	1 1 0 0 1 1 0 1 0 0 1 0 0 1 0

Moreover, another codeword form , which is One-Per-Class (OPC) code, is also considered and compared to the ECOC code. OPC code is a simple code. It can be described as follows.

For data set having k classes, the OPC code of class i is the codeword with k digits where i^{th} digit is one and other digits are zero for $i = 1, 2, 3, ..., k$. That is, OPC-code matrix is the identity matrix of size k. The examples of OPC code's designs for $k = 8$ are shown in TABLE IV.

TABLE IV
ONE PER CLASS CODES WITH $k = 8$

Class	Codeword
1	1 0 0 0 0 0 0 0
2	0 1 0 0 0 0 0 0
3	0 0 1 0 0 0 0 0
4	0 0 0 1 0 0 0 0
5	0 0 0 0 1 0 0 0
6	0 0 0 0 0 1 0 0
7	0 0 0 0 0 0 1 0
8	0 0 0 0 0 0 0 1

IV. EXPERIMENTS

A. Data Sets

Six data sets used in this experiment which are balance, vehicle, nursery, Ecoli and yeast are selected from UCI machine learning repository [24]. The characteristics of each data set are summarized in TABLE V. All data set is of type classification and has more than two classes.

B. Experimental Methodology and Results

There are five main steps to solve multiclass classification problem:

1) *Preparing the data*

After data sets are selected, each raw data set is normalized to numeric form in the set of $[0, 1]$. Let $X = \{x_1, x_2, x_3, \ldots, x_n\}$ be a set of numeric data with n samples.

TABLE V
UCI DATA SETS USED IN THIS STUDY

Data	No. of classes	No. of features	Sample set size
Balance	3	4	625
Vehicle	4	18	846
Nursery	5	8	12,960
Ecoli	8	7	336
Yeast	10	8	1,484
Vowel	11	10	990

Each x_i is mapped to be $x_i^{normal} \in [0,1]$ using following formula.

$$x_i^{normal} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3}$$

where

$$x_{min} = \min_i \{x_i \mid x_i \in X\},$$

$$x_{max} = \max_i \{x_i \mid x_i \in X\},$$

$i = 1, 2, 3, \ldots, n$. Then, data set will be separated into two parts for training phase and testing phase using 10-fold average method.

2) *Code design*

One-per-class codes and two forms of ECOC, exhaustive code and random ECOC, are constructed based on the concept described in section III. Exhaustive code will be applied to balance, vehicle and nursery having number of classes from three to seven whereas random ECOC is applied to Ecoli, yeast and vowel having more than seven classes.

3) *Classification*

NN and CMTNN models are constructed and compared. The classification algorithm is separated into two parts: training phase and testing phase in which each phase can be explained as follows.

Let x be a vector of m features, C_t be a set of codewords for truth information, C_f be a set of codewords for falsity information, D_t be a truth training set and D_f be a falsity training set which are denoted by

$x = (x_1, x_2, x_3, ..., x_m) \in X$,
$C_t = \{c_{t_1}, c_{t_2}, c_{t_3}, ..., c_{t_k}\}$,
$C_f = \{c_{f_1}, c_{f_2}, c_{f_3}, ..., c_{f_k}\}$,
$D_t = \{< x, c_t > \in X \times C_t\}$ and
$D_f = \{< x, c_f > \in X \times C_f\}$, respectively.

In training phase of NN model, only truth target values are used to train neural networks to predict truth output values; see Fig. 2.

For CMTNN, input information D_t including feature vectors and their truth target codewords are trained using feed-forward backpropagation method to obtain the classifier function $L_t : X \rightarrow C_t$. By the same way, the function $L_f : X \rightarrow C_f$ is obtained using input set D_f constituting of feature vectors and their falsity target codewords. The process of training phase of CMTNN model is shown in Fig. 3. CMTNN model consists of truth neural network (Truth NN) and falsity neural network (Falsity NN) where architectures and properties of Truth NN and Falsity NN are the same. However, the truth NN is trained from target codeword vectors to predict truth output but falsity NN is trained from complementary of target codeword vectors to predict falsity output.

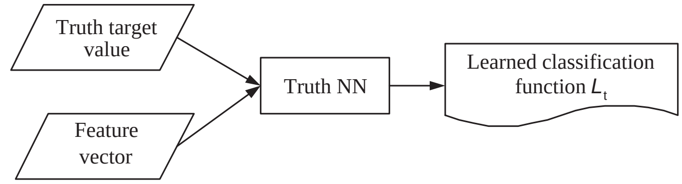

Fig. 2. Training Phase of neural network model

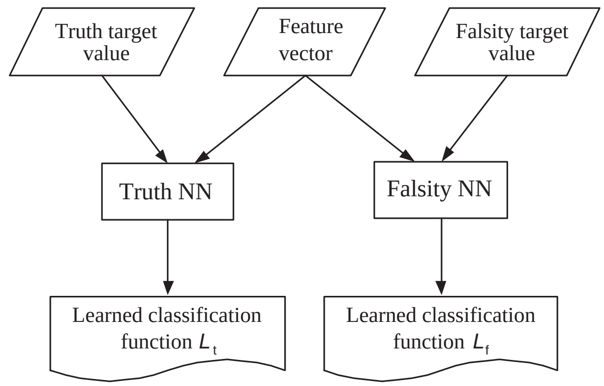

Fig. 3. Training Phase of complementary neural network model

In testing phase, the rest data set will be used. Features of a new sample are transferred through the truth NN and falsity NN basing on the learned classification function from truth NN and falsity NN, respectively. Then the truth and falsity outputs, $T(x)$ and $F(x)$, are obtained. The testing phase algorithm for CMTNN is shown in Fig. 4.

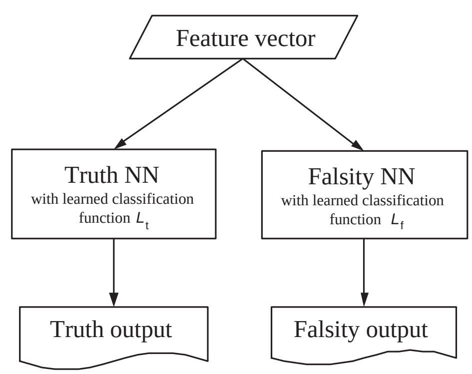

Fig. 4. Testing Phase of complementary neural network model

Since the predicted codeword $P(x)$, aggregating from the truth and falsity output, may not perfectly match to the predefined class, the decoding method is required for mapping $P(x) \rightarrow C_t$.

TABLE VI
AVERAGE CLASSIFICATION ACCURACY OBTAINED FROM THE TEST DATA

	Method	%correct					
		Balance	Vehicle	Nursery	Ecoli	Yeast	Vowel
NN	OPC:						
	min. dis.	83.72	56.59	87.95	74.56	0	16.36
	ECOC:						
	min. dis.	89.45	60.17	89.91	83.04	24.47	33.03
CMTNN	OPC:						
	min. dis.	91.68	68.09	88.04	79.33	24.47	31.01
	$T > F$	91.68	68.09	88.04	58.53	47.91	39.19
	ECOC:						
	min. dis.	92.00	82.62	91.00	84.81	52.10	41.92
	$T > F$	91.69	82.50	82.48	84.81	51.83	41.92

4) *Decoding*

Let $c_t = c_t^1 c_t^2 c_t^3 ... c_t^L$, $c_f = c_f^1 c_f^2 c_f^3 ... c_f^L$, $T_1 T_2 T_3 ... T_L \in T(x)$ and $F_1 F_2 F_3 ... F_L \in F(x)$ where L is length of the codeword. The decoding technique basing on minimum distance and $T > F$ are applied and described below.

• Minimum distance

For fixed x, find the distance d between output and each predefined class for CMTNN model as following:

$$d = \sum_{i=1}^{L} |T_i - c_t^i| + \sum_{i=1}^{L} |F_i - c_f^i|. \quad (4)$$

Note that if we decode on the neural network dealing only with the truth output, there are only one term in the right hand side. Then the class with minimum distance is chosen.

• $T > F$

Since this technique requires both truth and falsity outputs, it can be applied only for CMTNN model. For each i^{th} sample, if the truth output value is greater than the falsity output value $T_i > F_i$, then 1 is returned. Otherwise, 0 is returned. Then, binary values of L digits are obtained. If these values form to be one of defined codewords, class is assigned. Otherwise, minimum distance technique is applied.

After the predicted class is obtained from decoding, it will be compared to the truth target codeword to evaluate an efficiency of the proposed method which is shown in the next step.

5) *Evaluation*

The percentage of accuracy prediction can be calculated using following formula.

$$\%\ correct = \frac{\#\ correct\ predicted\ samples}{\#\ all\ samples} \times 100. \quad (5)$$

To obtain percentage of improvement, the original value and new value are used; see equation 6.

$$\%\ improvement = \frac{new\ value - original\ value}{original\ value} \times 100. \quad (6)$$

From the experiment, the accurate percentages of classification are obtained showing in TABLE VI.

Results shown in TABLE VI can be explained in three parts: results obtained from NN, CMTNN, and the comparison between these two models.

• Results obtained from NN model

The first part of TABLE VI shows the results obtained from NN model using OPC code and ECOC based on minimum distance. We found that all data sets give better results when applying ECOC.

• Results obtained from CMTNN model

The second part of TABLE VI shows the results obtained from CMTNN in which two types of codeword are also compared. For OPC code, using $T > F$ as the decoding technique provides more accuracy prediction for yeast and vowel, worse results for Ecoli, and similar results for balance, vehicle and nursery than the minimum distance technique. For ECOC, applying minimum distance gives a bit better results for balance, vehicle and yeast, and similar results for nursery, Ecoli and vowel. Considering results obtained from CMTNN model based on minimum distance, it was found that ECOC performs better results than OPC. Also, ECOC provides better results than OPC when using CMTNN based on $T > F$. On average, combining CMTNN and ECOC provides better performance when compared to combining CMTNN and OPC code.

• Comparison results between NN and CMTNN models

In the OPC code with minimum distance, CMTNN model improves the classification performance as compared to NN model for all data sets which is the same trend when using ECOC code with minimum distance. Therefore, we can conclude that CMTNN gives better results than NN. It can also be noted that, CMTNN and ECOC based on minimum distance provides the highest accuracy percentage of prediction for all data sets.

In each data set, results obtained from six techniques are compared and shown in Fig. 5 to Fig. 10 including balance, vehicle, nursery, Ecoli, yeast and vowel, respectively.

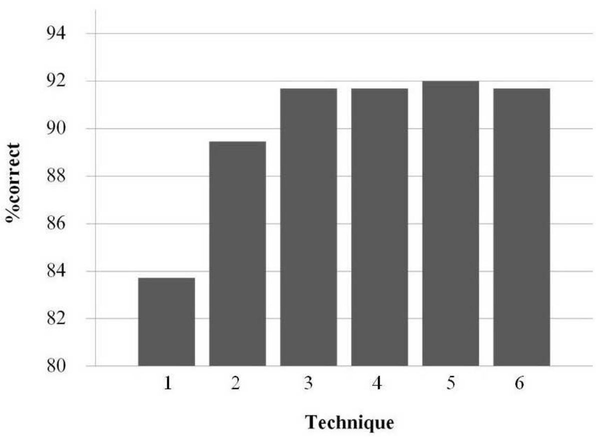

Fig. 5. Comparison results of balance data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

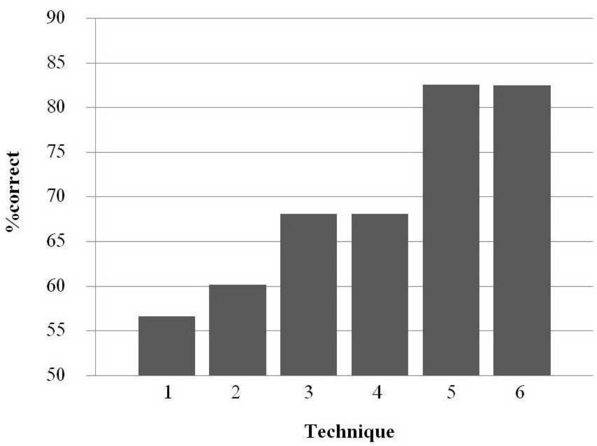

Fig. 6. Comparison results of vehicle data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

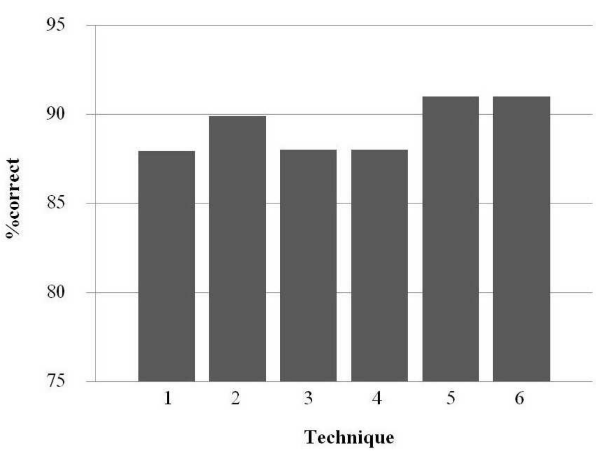

Fig. 7. Comparison results of nursery data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

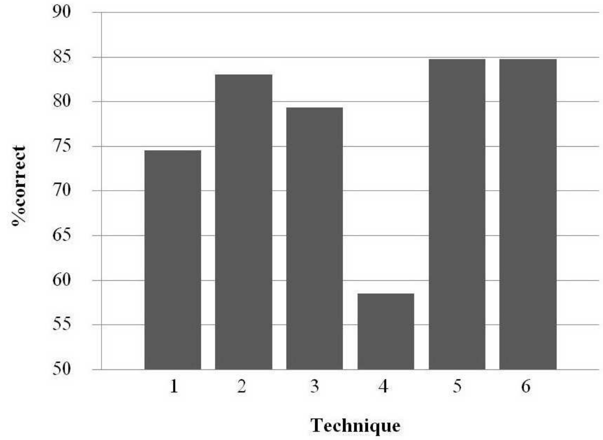

Fig. 8. Comparison results of ecolii data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

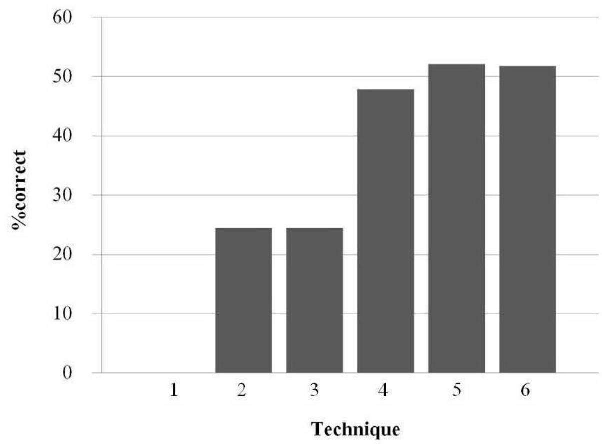

Fig. 9. Comparison results of yeast data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

We observe from TABLE VI that percentages of classification accuracy obtained from data set having less number of classes are higher than accurate percentages obtained from data set having more number of classes. It might be because of the more defined classes the more error probability.

TABLE VII shows the increased accuracy percentage of each method comparing to our proposed method. Each value is calculated using (6). However, there are some problems when applying this equation to NN model with OPC code for the yeast data set because its denominator is equal to zero. That is why we cannot calculate the percentage of improvement for this method. Since there are two decoding techniques applied to our method, averaging of these two values can be used as a representative for the comparison. It can be seen that our method much affects results for balance, nursery, and Ecoli and quite affects results for vehicle, yeast, and vowel considering from high percentages of improvement.

Therefore, we can conclude empirically that the CMTNN model provides better performance than NN model and ECOC

TABLE VII

THE PERCENT IMPROVEMENT OF THE PROPOSED TECHNIQUE (CMTNN+ECOC) COMPARED TO OTHER TECHNIQUES.

	Method	%improvement of CMTNN+ECOC on average					
		Balance	Vehicle	Nursery	Ecoli	Yeast	Vowel
NN	OPC :						
	min. dis.	9.70	45.89	3.47	13.75	-	156.23
	ECOC :						
	min. dis.	2.68	37.21	1.21	2.13	112.36	26.91
CMTNN	OPC:						
	min. dis.	0.18	21.25	3.36	6.91	112.36	35.18
	$T > F$	0.18	21.25	3.36	44.9	8.46	6.97

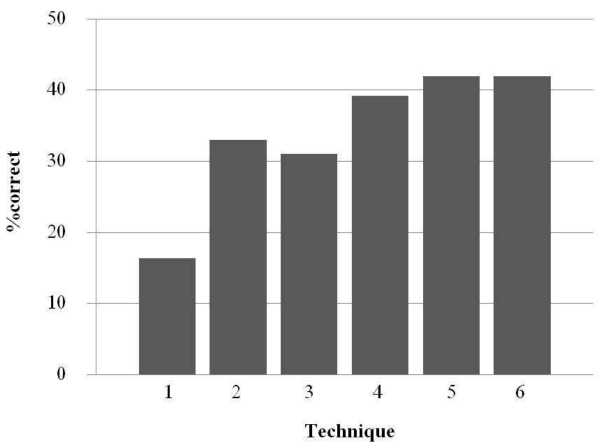

Fig. 10. Comparison results of vowel data set from various techniques: (1) NN+OPC, (2) NN+ECOC, (3) CMTNN+OPC+minimum distance, (4) CMTNN+OPC+$T > F$, (5) CMTNN+ECOC+minimum distance, and (6) CMTNN+ECOC+$T > F$

technique gives better results than OPC code. Moreover, as we desire, the results obtained from the proposed technique, the combination between CMTNN model and ECOC method, outperforms the existing techniques.

V. CONCLUSION

In this paper, combining CMTNN and ECOC based on minimum distance or $T > F$ are applied to solve multiclass classification problems. The proposed methods are tested using six data sets from UCI machine learning repository database which are balance, vehicle, nursery, Ecoli, yeast and vowel. The results show that our method provides better accuracy percentage than traditional techniques based on only CMTNN or ECOC in which using CMTNN and ECOC with minimum distance gives highest accuracy percentage of the prediction (best results for balance, vehicle, and yeast data sets and same results when compared to $T > F$ technique for nursery, Ecoli, and vowel). In the future, we will apply this technique to real world applications such as credit rating prediction, bankruptcy prediction and bond rating problem.

ACKNOWLEDGMENT

This research project is supported by Faculty of Science, Mahidol University and in the case of the second author, it was also supported by the Development and Promotion of Science and Technology talents project (DPST).

REFERENCES

[1] S. Har-Peled, D. Roth, and D. Zimak, “Constraint classification for multiclass classification and ranking,” in *Proc.* 16^{th} *Annual Conference on Neural Information Processing Systems (NIPS 2002)*. MIT Press, 2003.

[2] C. M. Bishop, *Neural Network for Pattern Recognition*. Oxford, 2005.

[3] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Contr. Signals Syst.*, vol. 2, pp. 303–314, 1989.

[4] S. Haykin, *A Comprehensive Foundation*. New York: Macmillan, 1994, ch. Neural Networks.

[5] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, pp. 359–366, 1989.

[6] D. Bhattacharyya and T. Kim, “Signature recognition using artificial neural network,” in *Proc.* 9^{th} *WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS’10)*, 2010.

[7] B. Kovari, A. Horvath, B. Toth, and H. Charaf, “Local feature based off-line signature verification using neural network classifiers,” in *Proc.* 11^{th} *WSEAS International Conference on Mathematical methods, computational techniques and intelligent systems (MAMECTIS’09)*, 2009.

[8] R. I. Mubark, H. A. Keshk, and M. I. Eladawy, “Prediction of hemoglobin structure from dna sequence through neural network and hidden markov model,” in *Proc.* 7^{th} *WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS’08)*, 2008.

[9] B. Ahn, S. Cho, and C. Kim, “The integrated methodology of rough set theory and artificial neural network for business failure prediction,” *Expert Systems with Applications*, vol. 18, pp. 65–74, 2000.

[10] T. Tang and L. Chi, “Neural networks analysis in business failure prediction of chinese importers: A between-countries approach,” *Expert Systems with Applications*, vol. 29, pp. 244–255, 2005.

[11] Y. S. Kim, “Comparison of the decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size,” *Expert Systems with Applications*, vol. 34, pp. 1227–1234, 2008.

[12] A. Eskandarinia, H. Nazarpour, M. Teimouri, and M. Z. Ahmadi, “Comparison of neural network and k-nearest neighbor methods in daily flow forcasting,” *Applied Science*, vol. 11, pp. 1006–1010, 2010.

[13] M. A. Razi and K. Athappilly, “A comparative predictive analysis of neural networks (nns), nonlinear regression and classification and regression tree (cart) models,” *Expert Systems with Applications*, vol. 29, pp. 65–74, 2005.

[14] P. du Jardin, J. Ponsaille, V. Alunni-Perret, and G. Quatrehomme, “A comparison between neural network and other metric methods to determine sex from the upper femur in a modern french population,” *Forensic Science International*, vol. 192, pp. 127.e1–127.e6, 2009.

[15] R. Lippmann, "Pattern classification using neural networks," *IEEE Commun. Mag.*, vol. 27, pp. 47–64, 1989.

[16] P. Kraipeerapun, C. Fung, and K. Wong, "Uncertainty assessment using neural networks and interval neutrosophic sets for multiclass classification problems," *WSEAS Transactions on Computers*, vol. 6, pp. 463–470, 2007.

[17] P. Kraipeerapun and C. Fung, "Binary classification using ensemble neural networks and interval neutrosophic sets," *Neurocomputing*, vol. 72, pp. 2845–2856, 2009.

[18] E. Kong and T. G. Dietterich, "Error-correcting output coding correct bias and variance," in *The XII International Conference on Machine Learning*, San Francisco, 1995, pp. 313–321.

[19] T. Dietterich and G. Bakiri, "Error - correcting output codes: A general method for improving multiclass inductive learning programs," in *Proceedings of AAAI-91*. AAAI Press / MIT Press, 1991, pp. 572–577.

[20] ——, "Solving multiclass learning problem via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[21] R. Schapire, "Using output codes to boost multiclass learning problems," in *Machine Learning: Proceedings of the Fourteenth International Conference*, 1997, pp. 313–321.

[22] T. Windeatt and R. Ghaderi, "Binary labelling and decision level fusion," *Information Fusion*, vol. 2, p. 103112, 2001.

[23] L. I. Kuncheva, *Combining Pattern Classifiers Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.

[24] D. Newman, S. Hettich, C. Blake, and C. Merz, "Uci repository of machine learning databases," online, November 2009, http://archive.ics.uci.edu/ml/datasets.html.