Applied Mathematics and Statistics
https://www.sciltp.com/journals/ams

Article

Using the ECOC Algorithm to Classify Diabetes with Fuzzy-Mapped Decoding Method

Ying Bai [1,*] and Dali Wang [2]

[1] Computer Science, Engineering and Mathematics, Johnson C. Smith University, Charlotte, NC 28216, USA
[2] Physics and Computer Science, Christopher Newport University, Newport News, VA 23606, USA
* Correspondence: ybai@jcsu.edu

How To Cite: Bai, Y.; Wang, D. Using the ECOC Algorithm to Classify Diabetes with Fuzzy-Mapped Decoding Method. *Applied Mathematics and Statistics* **2026**, *3*(1), 1. https://doi.org/10.53941/ams.2026.100001

Received: 20 November 2025
Revised: 24 December 2025
Accepted: 25 December 2025
Published: 20 January 2026

Abstract: The Error Correcting Output Codes (ECOC) algorithm has played more and more important role in the artificial intelligence (AI) study and research fields in recent years due to its special properties and error correcting abilities. Various methods have been reported to improve its performance and classification accuracy. In this study, we developed a simplified ECOC encoding method to reduce the complexity in the encoding process and reduce its randomness in initializing the coding matrix. Furthermore, we also suggested a fuzzy mapping decoding (FMD) method used for the ECOC decoding process to improve its classification accuracy. By using this FMD, the classification accuracy can be improved up to 84% compared with the decoding method without using the FMD, which is 77%.

Keywords: ECOC classifications; fuzzy inference system; ECOC encoding and decoding; fuzzy mapping method

1. Introduction

The Error Correcting Output Codes (ECOC) algorithm has played a more and more important role in the artificial intelligence (AI) study and research fields in recent years due to its special properties and error correcting abilities. Different methods and processes have been reported to improve the ECOC's performance and its classification accuracy [1–3]. The main purpose of using ECOC is to simplify multiclass classification tasks into multiple binary classification jobs. By using either the One-vs-All (OvA) or One-vs-One (OvO) technique, some complicated multiclass classification tasks can be significantly reduced to multiple binary classification tasks.

Like all other AI models, an ECOC model establishment process needs two steps: encoding (training) and decoding (testing). One primary factor affecting the use of the ECOC algorithm is the encoding process, which is how to quickly and correctly establish a mapping between input classes and the binary encoding matrix, also called the coding matrix, that is composed of a collection of code words to represent each related class in the input data.

Many ECOC approaches [4–9] have been proposed to design a good coding matrix in recent years. Most Error correcting output codes (ECOC) algorithm constructs an ensemble or a group of binary base classifiers by randomly and independently assigning true/false pseudo labels (1 and 0 in the coding matrix) for each base task. The results of all the base learners are combined to make a classification or a prediction. Like all other supervised learning algorithms, ECOC also needs to be trained and tested based on the selected dataset. These processes consist of two main steps: (1) training or encoding process, (2) testing or decoding process. The most well-known coding strategy is the one-versus-all [1], where each class is discriminated against the rest of the classes.

In the encoding process, we create an encoding matrix to encode each class (row in the coding matrix) into a unique coding row that is as different as possible from the coding rows of the remaining classes. One example of the encoding matrix is illustrated in Figure 1 [10].

Class	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
0	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**
1	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**
2	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**
3	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
5	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**

Figure 1. A 10-bit error-correcting output code for a 6-class problem**.**

A row of the coding matrix represents the code of each class, while a column of the coding matrix represents the binary classes to be considered when learning a base classifier.

As shown in Figure 1, a 10-bit error-correcting output code for a 6-class problem is presented as a coding matrix. Each class can be presented as a unique binary string of length 10. This string can be called a codeword. For instance, class 3 has a codeword 1000010011. During the EEOC training process, one binary bit, also called a classifier, is learned for each column. For example, for the first column, we build a binary classifier to separate {0, 2, 3, 5} from {1, 4}. Thus, 10 binary classifiers are trained in this way. To classify a new testing data point x, all 10 binary classifiers are evaluated to obtain a 10-bit string or a codeword. Finally, we choose the class whose codeword is closest to x's output string as the predicted label.

Many different encoding methods have been reported recently. Zhou et al. reported using a confusion matrix to find the correlation between each pair of input predictor classes to build a new encoding method for ECOC [11]. MengXin Sun et al. proposed a new method to build the ECOC coding matrix based on a hierarchical partition of the class space with the aim of minimizing data complexity (named as ECOC-MDC) [12]. To overcome the non-competence problem and better model the multi-class problem for reducing the classification cost, [13] embedded a reject option to ECOC and presented a new variant of the ECOC algorithm called Reject-Option-based Re-encoding ECOC (ROECOC). Kai-Jie Feng et al. proposed a variable-length code words-based ECOC (VL-ECOC), which generates longer codes for hard classes than those for easy classes. VL-ECOC consists of two phases: the overall-class phase and the hard-class phase [14]. The aim of the research [15] is to optimize the original matrix so as to better match the trained base classifiers. This is done by considering the performances of base classifiers over the individual classes, and changing the ECOC matrix whenever it is deemed beneficial, while taking the HD information into account.

P. Radeva et al. proposed a novel technique called ECOC-ONE to improve an initial ECOC configuration by including new dichotomies guided by the confusion matrix over exclusive training subsets. In this way, the initial coding represented by an optimal decision tree is extended by adding binary classifiers forming a network [16]. Wang, L.-N, et al. investigated the influence of deep learning on ECOCs or N-ary ECOC with three different parameter sharing strategies [17]. O. Pujol et al. presented a heuristic method for learning error correcting output code matrices based on a hierarchical partition of the class space that maximizes a discriminative criterion [18]. F. Masulli and G. Valentini presented an extensive experimental work for evaluating the dependence among output errors of the decomposition unit in ECOC learning machines. In particular, it applied measures based on mutual information to compare the dependence of ECOC multi-layer perceptron (ECOC MLP), made up by a single multi-input multi-output MLP, and ECOC ensembles made up by a set of independent and parallel dichotomizers (ECOC PND) [19]. S. A. A. Ahmed et al. analyzed an error correcting output coding (ECOC) framework for constructing ensembles of deep networks and proposed different design strategies to address the accuracy-complexity trade-off [20]. In [21], several commonly used artificial intelligence (AI) methods are used to design algorithms for the bitwise decoding of ECC. Jon-Lark Kim and Miseong Kim considered Hadamard matrices of various orders for the Hadamard ECOC to determine which orders of Hadamard matrices give a good performance compared to the number of classes [22].

However, all of those methods are either too time-consuming or too complicated to be implemented in actual applications. Another issue is that most of the recent research and studies on the ECOC algorithm only concentrate on the mapping or building coding matrix or the encoding process, but few of them paid attention to the decoding process.

Two contributions are involved in this study; first, we used a simple and easy way to build this coding matrix with the real and actual datasets as predictors. To make things clear and straightforward, we utilized a Diabetes dataset as an example to illustrate how to simplify this building or mapping between the actual predictor values (continuous values) in the dataset and the coding matrix represented in binary format. Second, to reduce some uncertainties or errors existing in the testing data used in the decoding process, a fuzzy mapping method is used to obtain better-matched binary classifiers to improve the classification accuracy.

A comparison study is performed to compare the classification accuracy between the standard mapping method (SMM) and fuzzy mapping method (FMM) for the testing data. The comparison result shows that FMM provided a better classification accuracy (84%) than SMM (77%).

The purpose of building a coding matrix is to map the actual physical or continuous predictors' values in the dataset to a sequence of binary classes to convert a multi-classification task to multiple binary classification tasks. The resulting coding matrix is composed of m rows, in which each of them is mapped to a class, and n columns, which are mapped to n features or n binary classifiers, represented by a binary value, either 1 (true) or 0 (false). Each row is called a code word. For a one-versus-all strategy, all binary bits in each row should be as different as possible from those in other rows. To make this possible, we need to have a closer look at our target dataset, Diabetes Dataset [23], which is located on the site: https://data.mendeley.com/datasets/wj9rwkp9c2/1 (accessed on 15 November 2025).

2. Simplified Method to Build Coding Matrix

2.1. Using a Simplified Method to Build an ECOC Matrix

As we mentioned, there are many studies that are related to creating and building various sophisticated coding matrices in recent years. However, most of them can be used as a general strategy to build the coding matrix, combined with different additional algorithms for general applications. Those methods are not appropriate for some actual datasets in real applications.

In this study, we proposed a simplified and straightforward method that is dataset-dependent and closely related to the actual applications. In other words, this method is flexible, and it is derived based on the different or individual application with actual implementation value.

The generation process of a coding matrix is a mapping from a set of discrete values or predictors in a dataset to a collection of binary classifiers in that matrix, $M(i,j)$. The goal of encoding is to construct a matrix $\mathbf{M} = (m_{ij})_{k \times l}$, $m_{ij} \in \{0, 1\}$, where rows hold the code words of the class and columns represent bipartitions for the dichotomizers.

Let S = $\{(x_1, y_1), \ldots (x_m, y_m)\}$ be a set of m training examples where each instance x_i belongs to a domain X, and each label y_i is a label from the set $Y = \{l_1, l_2, \ldots l_k\}$. A multiclass classifier is a function F: $X \rightarrow Y$ that maps an instance x into an element y of Y. In the ECOC algorithm, a discrete output coding matrix M is a matrix of size $k \times l$ over $\{0, 1\}$ where each row of M corresponds to a class $y \in Y$. Each column of M defines a partition of Y or a binary classifier. The column j of the coding matrix M and row i that corresponds to class C_i is filled as follows: $M(i,j)$

$$M(i,j) = \begin{cases} 1 & if x_{ij} \in C_i \quad (i = 1, 2, \ldots k, j = 1, 2, \ldots l) \\ \\ 0 & if x_{ij} \notin C_i \quad (i = 1, 2, \ldots k, j = 1, 2, \ldots l) \end{cases} \tag{1}$$

Equation (1) is based on the One-vs-All (OvA) method, and it is adopted here.

The basic idea behind this Equation is that each class (row in the coding matrix) should be as different as possible from the coding rows of the other classes in this coding matrix $M(i,j)$.

A key issue is how to define the condition, *if* $x_{ij} \in C_i$? To simplify this definition, we divided the predictors x_{ij} into i classes based on the range of each class. For example, the dataset we used for this study is the Diabetes Dataset, which is a 1000 × 14 matrix listing all measured or tested parameters, such as Age, HbA1c, HDL, LDL, VLDL, and BMI, related to diabetes. We only adopt 11 columns or features with 30 rows from that dataset as our case. Three class labels, Y (Diabetes), N (Not Diabetes), and P (Pre Diabetes), are involved in this dataset.

Table 1 shows a selected partial dataset from our Diabetes_M.xlsx dataset. Only 10 rows or records in that dataset are selected as examples to illustrate how to use those data to derive the encoding process for the ECOC algorithm.

As the training process starts, each row of the coding design or the binary matrix corresponds to a distinct class, and each column corresponds to a binary classifier or learner. In a ternary coding design, for a particular column or binary classifier:

- Each row in the corresponding column containing 1 directs the binary classifier to group all observations in the corresponding class into a true class.

- Each row in the corresponding column containing 0 directs the binary classifier to group all observations in the corresponding class into a false class.

For our case, each binary classifier (CC1~CC10) is trained with a group of inputs in the related column to obtain the desired binary encoding string, or for three output classes, Y, N, and P. Based on Table 1, we can get the range for each class as shown in Table 5. These ranges are computed based on each feature's range in the dataset. Based on Table 5 and Equation (1), the encoding results for these 3 classes are:

- Y[1 0 0 1 0 0 0 0 0 1]
- N[1 1 1 1 1 0 0 0 1 0]
- P[1 0 1 0 0 1 0 1 0 0]

All of them come from the three highlighted rows in Tables 1–5 shown below.

A relation between the code length and the variables (Nc) is codelength ≥ $\text{Log}_2(\text{Nc})$. For our case, we have 1000 classes, so $\text{Log}_2(1000) = 9.96$, which is good for our study. More bits should be used if more variables are used.

Table 1. A selected partial dataset.

Age	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class
54	3.8	32	10.8	3.4	0.8	1.4	1.6	1.5	27	Y
53	3	48	9.5	7.2	2	1.9	2	0.7	32	Y
30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Y
50	2.6	106	4	6.3	4.4	1	3.6	1.7	24	Y
60	6	72	10.7	4.4	2.1	1.1	2.5	0.9	26	Y
60	4	63	12	3.6	5.1	0.9	2.5	0.9	30	Y
57	5.3	66	7.9	3.7	2.9	1.1	5.5	1.3	36	Y
50	3.5	59	6	4	2.1	1.4	1.9	0.9	25	P
30	3	45	4.1	4.9	1.3	1.2	3.2	0.5	22	N
61	5.9	56	8.2	4.9	2.1	1.1	2.5	0.9	28	Y
CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	

Table 2. ECOC for binary classification 1.

Age	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class
1	0	0	1	0	0	0	0	0	1	Y
0	0	1	0	0	1	0	1	0	0	P
1	1	1	1	1	0	0	0	1	0	N
CC1	**CC2**	**CC3**	**CC4**	**CC5**	**CC6**	**CC7**	**CC8**	**CC9**	**CC10**	

Table 3. ECOC for binary classification 2.

Age	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class
0	0	1	0	1	1	0	1	1	0	Y
0	1	0	1	1	0	1	0	1	1	P
1	1	1	1	1	0	0	0	1	0	N
CC1	**CC2**	**CC3**	**CC4**	**CC5**	**CC6**	**CC7**	**CC8**	**CC9**	**CC10**	

Table 4. ECOC for binary classification 3.

Age	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class
0	1	0	0	0	0	1	0	0	1	Y
1	0	1	0	0	1	0	1	0	0	P
0	0	0	0	0	0	0	0	0	1	N
CC1	**CC2**	**CC3**	**CC4**	**CC5**	**CC6**	**CC7**	**CC8**	**CC9**	**CC10**	

Table 5. The range for each feature.

Item	Range	Class N	Class P	Class Y
Age	20–76	20–40	41–50	51–76
Urea	2–11	2–3.7	3.8–5.5	5.6–11
CR	20–116	20–46	47–84	85–116
HbA1c	4.0–16	4.0–6.3	6.4–9.7	9.8–16
Chol	2–9	2.4–4.5	4.6–5.9	6.0–9.0
TG	0.5–14	0.5–4.0	4.1–9.4	9.5–14
HDL	0.2–10	0.2–4.5	4.6–8.0	8.1–10
LDL	0.2–10	0.2–4.5	4.6–8.0	8.1–10
VLDL	0–32	0–10	10–20	21–32
BMI	18–50	18–29	30–40	41–50

2.2. The Fuzzy Mapped Decoding Method for ECOC

Figure 2 shows a functional block diagram of the decoding process for the ECOC algorithm to classify multi-class objects with multiple binary classification algorithms. A testing dataset should be used to validate or decode the ECOC results.

As shown in Figure 2, both the standard decoding method and our new fuzzy mapping method are utilized for the decoding purpose. The reason we used both of them is to compare their decoding performances. The so-called standard decoding method is similar to the encoding process we discussed above. In which the true or false binary classifiers can be derived based on the range of the testing data. A more detailed discussion about our fuzzy mapping method is given below in this part.

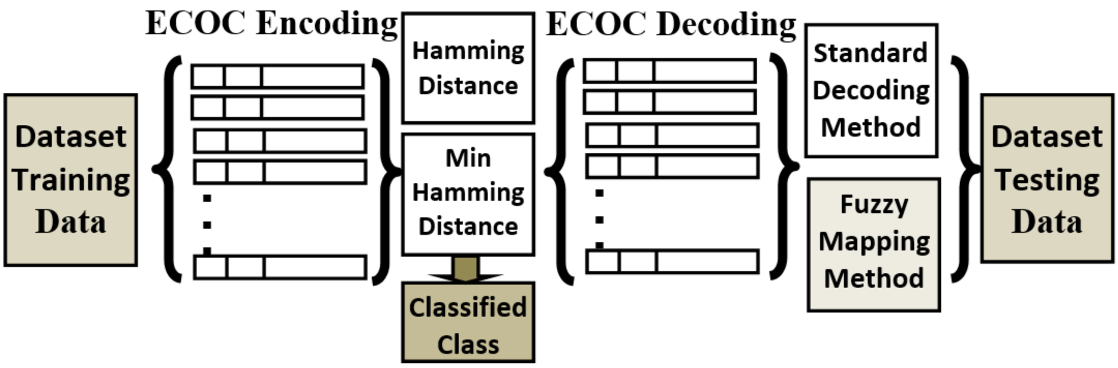


Figure 2. A functional block diagram of the decoding process for ECOC.

Due to the randomness property and possible overlapping in class ranges among either training or testing data, some ambiguous or inaccurate data may be involved in this standard decoding process. In order to reduce those ambiguities and errors in the data, a fuzzy mapping method is a good candidate to help solve them. As regular FIS design process, the testing data will be processed in three popular steps:

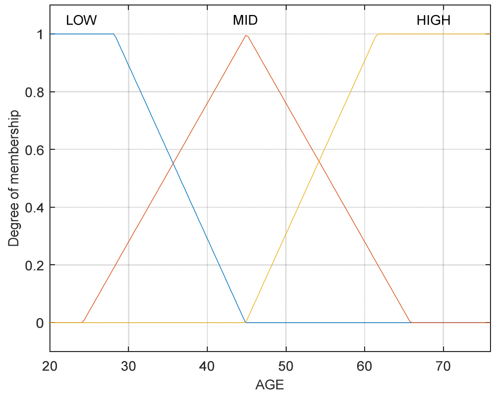
(1) Fuzzifications
(2) Design control rules
(3) Defuzzifications

Since we have 10 features for this example, we need to perform these 3 steps for each of them. To make things simple, we just use the Age feature as an example to illustrate these operations.

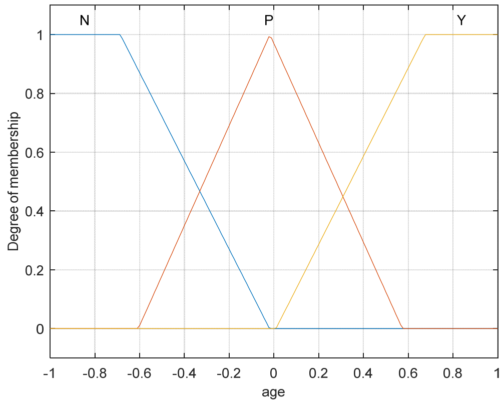
Three membership functions (MFs) are used for both input variables, such as Age, HbA1C, HDL, LDL, VLDL, and BMI (LOW, MID, and HIGH) and the output class (N, P, Y), including the Age. Figure 3 shows these MFs. Three fuzzy control rules are developed for these variables, a set of example rules for the Age are:

r1 = **"If AGE is LOW then age is N"**;
r2 = **"If AGE is MID then age is P"**;
r3 = **"If AGE is HIGH then age is Y"**;

where the AGE is input and the age is the output. All other variables have the similar rules.



(**a**) The input MFs.

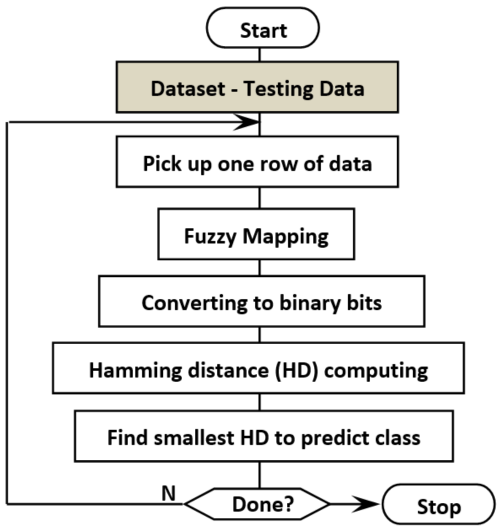


(**b**) The output MFs.

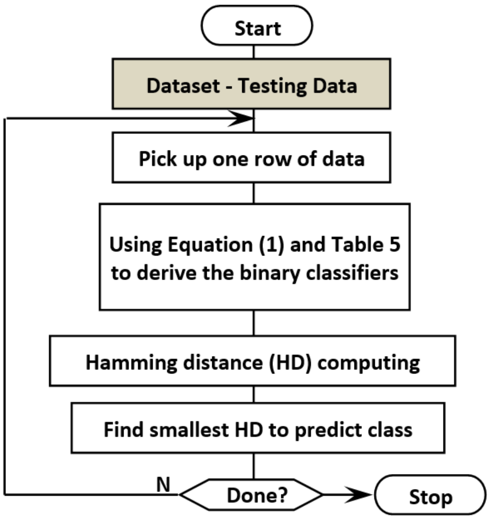
Figure 3. Membership functions for inputs and outputs.

After the defuzzification process, each piece of testing data or each binary classifier needs to be converted to a binary value, either 1 or 0. This conversion process needs to use some bounds to be determined by some actual testing for each related range applied to each feature. Then a sequence of Hamming distances will be calculated for each row, and the smallest distances will be selected and compared with the original class to check the classification results.

Figure 4 shows floating charts of using or not using this fuzzy mapping method for the ECOC decoding process. Figure 4a is for using this fuzzy mapping method for the ECOC decoding, but Figure 4b is for without using that fuzzy mapping process. The meaning of without using fuzzy mapping method is to use Equation (1) and Table 5 to calculate each classifier for each row, just as we did in Section 2.1.



(**a**) Using fuzzy mapping method.



(**b**) Without using the fuzzy mapping method.

Figure 4. Floating charts of ECOC with and without fuzzy mapping.

A comparison study is performed to compare the classification accuracy for these two decoding processes, one with the fuzzy mapping method, and the other is not. The detailed comparison result is discussed in next section.

3. Experimental Results

For comparison purposes, we used partial data in our modified Diabetes Dataset, randomly selected 30 records that are different from the training data from that dataset to test our ECOC algorithm with and without fuzzy mapping decoding method, and compare both methods to check and confirm the classification accuracy of both of them.

Table 6 shows the classification results of using the fuzzy mapping decoding method, and Table 7 shows similar results without using it.

For both Tables, the first 10 columns are features for each record, and the column Class contains the desired or the original class label for each row or record. The column, Y, N, and P, includes the Hamming distance calculated based on the fuzzy mapping for ECOC decoding on each testing record. The column, MIN HD, contained the smallest Hamming distance based on the Hamming distances included in the columns, Y, N, and P. The column Result indicates whether the predicted class derived from the smallest Hamming distance matches the desired class. A letter T indicates that both classes are a match, but an F means that both classes are not the same.

Table 7 shows the classification results without using fuzzy mapping method for the ECOC decoding process. All values in the first 11 columns in Table 7 are identical with those values shown in Table 6. But the values in the following 3 columns, Y, N and P, are derived without using the fuzzy mapping method.

A comparison can be performed between Tables 6 and 7 to check to find the classification accuracy for both methods. It can be found from Table 6 that the classification accuracy is about 84% when the fuzzy mapping method is used, but it is 77% for Table 7, in which no fuzzy mapping method is adopted.

Table 6. The classification results with the fuzzy mapping method for ecoc decoding.

AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class	Y	N	P	MIN HD	Result
59	4.7	55	7.0	3.4	1.8	1.1	1.8	0.8	26	Y	2	7	5	2	T
58	4.3	56	9.1	4.4	2	1.0	2.5	0.9	29	Y	4	7	6	4	T
52	2.6	30	10	5.4	2.1	0.9	0.9	2.0	37	Y	1	6	7	1	T
43	2.1	55	5.7	4.7	5.3	0.9	1.7	2.4	25	P	5	6	3	3	T
32	3.6	28	4.0	3.8	2.0	2.4	3.8	1.0	24	N	5	4	3	3	F
55	3.9	51	10.7	3.6	1.2	1.4	1.7	0.5	30	Y	2	7	5	2	T
55	4.3	56	7.0	3.8	2.6	1..0	2.1	1.6	30	Y	2	7	6	2	T
55	2.4	36	12.4	4.3	2.4	1.8	1.3	1.5	29	Y	1	6	6	1	T
33	2.0	54	5.4	3.7	1.3	0.8	2.4	0.6	22	N	3	6	3	3	F
54	4.3	44	7.6	5.4	2.0	1.0	3.5	0.9	30	Y	3	6	7	3	T
55	6.66	61	7.7	4.1	2.03	1.08	2.2	14	37.6	Y	3	8	6	3	T
49	4.8	46	5.8	4.8	1.1	1.7	2.6	0.5	24	P	5	6	5	5	T

Table 6. *Cont.*

AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class	Y	N	P	MIN HD	Result
73	4.3	79	6.9	5.3	1.4	1.5	3.2	0.6	28	Y	4	5	7	4	T
25	7.9	35	4.0	4.3	3.5	0.8	1.3	0.8	22	N	5	6	4	4	F
57	3.0	60	7.9	4.8	2.4	1.8	2.0	1.1	27	Y	2	7	6	2	T
63	8.0	93	9.0	5.9	2.2	1.2	3.7	1.0	29	Y	4	7	5	4	T
50	4.3	59	6.1	4.0	3.0	1.0	1.8	1.3	24	P	3	8	5	3	F
54	6.3	87	8.2	3.3	0.7	0.8	1.2	0.3	29	Y	2	7	6	2	T
56	5.0	61	9.2	4.6	1.0	1.5	1.45	0.7	30	Y	3	8	7	3	T
58	4.5	53	9.0	5.5	3.2	3.2	1.4	1.5	35	Y	4	7	8	4	T
56	4.1	51	11.5	5.9	1.7	1.7	1.04	0.7	31	Y	3	8	6	3	T
30	5.7	53	6.0	5.4	1.7	1.4	3.3	0.7	22	P	6	7	4	4	T
56	3.4	44	4.8	4.1	1.5	0.8	1.7	1.4	39	Y	1	4	5	1	T
50	5.5	97	5.4	3.2	1.8	1.6	0.9	0.8	21	N	4	7	5	4	F
61	6.5	78	9.2	6.5	2.4	1.7	1.8	1.5	37	Y	3	8	6	3	T
30	5.7	53	6.0	5.4	1.7	1.4	3.3	0.7	22	P	6	7	4	4	T
52	7.6	84	8.1	4.2	2.0	0.7	2.6	0.9	36	Y	4	7	5	4	T
55	4.4	48	7.7	3.4	0.7	1.0	2.1	0.4	29	Y	1	6	7	1	T
57	3.2	31	11.9	6.1	2.6	0.9	4.1	1.1	30.1	Y	2	5	5	2	T
73	4.3	79	6.0	5.3	1.4	1.5	3.2	0.6	27	Y	4	7	5	4	T

Table 7. The classification results without the fuzzy mapping method for ecoc decoding**.**

AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	Class	Y	N	P	MIN HD	Result
59	4.7	55	7.0	3.4	1.8	1.1	1.8	0.8	26	Y	3	6	9	3	T
58	4.3	56	9.1	4.4	2	1.0	2.5	0.9	29	Y	1	5	2	1	T
52	2.6	30	10	5.4	2.1	0.9	0.9	2.0	37	Y	1	5	4	1	T
43	2.1	55	5.7	4.7	5.3	0.9	1.7	2.4	25	P	4	5	5	4	F
32	3.6	28	4.0	3.8	2.0	2.4	3.8	1.0	24	N	6	0	4	0	T
55	3.9	51	10.7	3.6	1.2	1.4	1.7	0.5	30	Y	0	6	4	0	T
55	4.3	56	7.0	3.8	2.6	1..0	2.1	1.6	30	Y	2	6	4	2	T
55	2.4	36	12.4	4.3	2.4	1.8	1.3	1.5	29	Y	1	3	4	1	T
33	2.0	54	5.4	3.7	1.3	0.8	2.4	0.6	22	N	3	3	0	0	F
54	4.3	44	7.6	5.4	2.0	1.0	3.5	0.9	30	Y	3	5	6	3	T
55	6.66	61	7.7	4.1	2.03	1.08	2.2	14	37.6	Y	4	7	3	3	F
49	4.8	46	5.8	4.8	1.1	1.7	2.6	0.5	24	P	3	4	7	3	F
73	4.3	79	6.9	5.3	1.4	1.5	3.2	0.6	28	Y	3	5	6	3	T
25	7.9	35	4.0	4.3	3.5	0.8	1.3	0.8	22	N	5	4	3	3	F
57	3.0	60	7.9	4.8	2.4	1.8	2.0	1.1	27	Y	3	5	6	3	T
63	8.0	93	9.0	5.9	2.2	1.2	3.7	1.0	29	Y	4	6	4	4	T
50	4.3	59	6.1	4.0	3.0	1.0	1.8	1.3	24	P	4	5	5	4	F
54	6.3	87	8.2	3.3	0.7	0.8	1.2	0.3	29	Y	3	7	4	3	T
56	5.0	61	9.2	4.6	1.0	1.5	1.45	0.7	30	Y	0	7	5	0	T
58	4.5	53	9.0	5.5	3.2	3.2	1.4	1.5	35	Y	2	6	4	2	T
56	4.1	51	11.5	5.9	1.7	1.7	1.04	0.7	31	Y	1	6	5	1	T
30	5.7	53	6.0	5.4	1.7	1.4	3.3	0.7	22	P	6	4	4	4	T
56	3.4	44	4.8	4.1	1.5	0.8	1.7	1.4	39	Y	1	3	2	1	T
50	5.5	97	5.4	3.2	1.8	1.6	0.9	0.8	21	N	5	5	6	5	T
61	6.5	78	9.2	6.5	2.4	1.7	1.8	1.5	37	Y	4	6	4	4	T
30	5.7	53	6.0	5.4	1.7	1.4	3.3	0.7	22	P	6	4	4	4	T
52	7.6	84	8.1	4.2	2.0	0.7	2.6	0.9	36	Y	4	5	3	3	F
55	4.4	48	7.7	3.4	0.7	1.0	2.1	0.4	29	Y	1	7	4	1	T
57	3.2	31	11.9	6.1	2.6	0.9	4.1	1.1	30.1	Y	3	4	3	3	T
73	4.3	79	6.0	5.3	1.4	1.5	3.2	0.6	27	Y	4	4	6	4	T

4. Conclusions

A simplified encoding method for the ECOC classification process is reported in this study. To improve the classification accuracy, a fuzzy mapping method is adopted and used for the ECOC decoding process. The experimental results show that better classification accuracy can be achieved after using the fuzzy mapping method for the ECOC decoding process.

Author Contributions

Dr. Dali Wang developed some parts on section 2 with equations, Dr. Bai also provided some contributions on section 2. Dr. Bai provided training (encoding) and testing (decoding) with MATLAB codes in section 3 and other sections.

Funding

This research received no external funding

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.”

References

1. Crammer, K.; Singer, Y. On the learnability and design of output codes for multiclass problems. *Mach. Learn.* **2002**, *47*, 201–233.
2. Bagheri, M.A.; Montazer, G.A.; Kabir, E. A subspace approach to error correcting output codes. *Pattern Recognit. Lett.* **2013**, *34*, 176–184.
3. Escalera, S.; Pujol, O.; Radeva, P. On the decoding process in ternary error correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 120–134.
4. Bai, X.; Niwas, S.I.; Lin, W.; et al. Learning ECOC Code Matrix for Multiclass Classification with Application to Glaucoma Diagnosis. *J. Med. Syst.* **2016**, *40*, 78.
5. Allwein, E.; Schapire, R.; Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. *J. Mach. Learn. Res.* **2002**, *1*, 113–141.
6. Dietterich, T.; Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell.Res.* **1995**, *2*, 263–286.
7. Frank, E.; Kramer, S. Ensembles of nested dichotomies for multiclass problems. In Proceedings of the 21st international Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; pp. 305–312.
8. Hastie, T.; Tibshirani, R. Classification by pairwise grouping. *Ann. Stat.* **1998**, *26*, 451–471.
9. Ghani, R. Using error-correcting codes for text classification. In Proceedings of the ICML ’00: Seventeenth International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; pp. 303–310.
10. Available online: https://www.khoury.northeastern.edu/home/vip/teach/MLcourse/4_boosting/lecture_notes/ecoc/ecoc.pdf (accessed on 15 November 2025).
11. Zhou, J.; Yang, Y.; Zhang, M.; et al. Constructing ECOC based on confusion matrix for multiclass learning problems. *Sci. China Inf. Sci.* **2016**, *59*, 1–14. https://doi.org/10.1007/s11432-015-5321-y.
12. Sun, M.; Liu, K.; Wu, Q.; et al. A novel ECOC algorithm for multiclass microarray data classification based on data complexity analysis, *Pattern Recognit.* **2019**, *90*, 346–362. https://doi.org/10.1016/j.patcog.2019.01.047.
13. Lei, L.; Song, Y.; Luo, X. A new re-encoding ECOC using reject option. *Appl. Intell.* **2020**, *50*, 3090–3100. https://doi.org/10.1007/s10489-020-01642-2.
14. Feng, K.-J.; Liong, S.-T.; Liu, K.-H. The design of variable-length coding matrix for improving error correcting output codes, *Inf. Sci.* **2020**, *534*, 192–217. https://doi.org/10.1016/j.ins.2020.04.021.
15. Zor, C.; Yanikoglu, B.; Windeatt, T.; et al. FLIP-ECOC: A Greedy Optimization of the ECOC Matrix. In *Computer and Information Sciences*; Lecture Notes in Electrical Engineering; Gelenbe, E., Lent, R., Sakellari, G., Sacan, A., Toroslu, H., Yazici, A. Eds.; Springer: Dordrecht, The Netherlands, 2011. https://doi.org/10.1007/978-90-481-9794-1_30.
16. Radeva, P.; Pujol, O.; Escalera, S.; et al. ECOC-ONE: A Novel Coding and Decoding Strategy. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR’06), Hong Kong, China, 20–24 August 2006; pp. 578–581. https://doi.org/10.1109/ICPR.2006.469.
17. Wang, L.-N.; Wei, H.; Zheng, Y.; et al. Deep Error-Correcting Output Codes. *Algorithms* **2023**, *16*, 555. https://doi.org/10.3390/a16120555.

18. Pujol, O.; Radeva, P.; Vitria, J. Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1001–1007.
19. Masulli, F.; Valentini, G. An experimental analysis of the dependence among codeword bit errors in ECOC learning machines. *Neurocomputing* **2004**, *57*, 189–214. https://doi.org/10.1016/j.neucom.2003.09.011.
20. Ahmed, S.A.A.; Zor, C.; Awais, M.; et al. Deep Convolutional Neural Network Ensembles Using ECOC. *IEEE Access* **2021**, *9*, 86083–86095. https://doi.org/10.1109/ACCESS.2021.3088717.
21. Sharma, E.; Shakeel, I.; Deo, R.C.; et al. A Comparative Study of Artificial Intelligence-Based Algorithms for Bitwise Decoding of Error Correction Codes. In Proceedings of the 22nd International Symposium on Communications and Information Technologies (ISCIT), Sydney, Australia, 16–18 October 2023; pp. 37–42. https://doi.org/10.1109/ISCIT57293.2023.10376079.
22. Kim, J.-L.; Kim, M. Error-correcting output codes for multi-class classification based on Hadamard matrices and a CNN model. *Procedia Comput. Sci.* **2023**, *222*, 252–261. https://doi.org/10.1016/j.procs.2023.08.163.
23. Available online: https://data.mendeley.com/datasets/wj9rwkp9c2/1 (accessed on 15 November 2025).