NORMALIZED DATA BARRIER AMPLIFIER FOR FEED-FORWARD NEURAL NETWORK

P. Fuangkhon[*]

Abstract: A boundary vector generator is a data barrier amplifier that improves the distribution model of the samples to increase the classification accuracy of the feed-forward neural network. It generates new forms of samples, one for amplifying the barrier of their class (fundamental multi-class outpost vectors) and the other for increasing the barrier of the nearest class (additional multi-class outpost vectors). However, these sets of boundary vectors are enormous. The reduced boundary vector generators proposed three boundary vector reduction techniques that scale down fundamental multi-class outpost vectors and additional multi-class outpost vectors. Nevertheless, these techniques do not consider the interval of the attributes, causing some attributes to suppress over the other attributes on the Euclidean distance calculation. The motivation of this study is to explore whether six normalization techniques; min-max, Z-score, mean and mean absolute deviation, median and median absolute deviation, modified hyperbolic tangent, and hyperbolic tangent estimator, can improve the classification performance of the boundary vector generator and the reduced boundary vector generators for maximizing class boundary. Each normalization technique pre-processes the original training set before the boundary vector generator or each of the three reduced boundary vector generators will begin. The experimental results on the real-world datasets generally confirmed that (1) the final training set having only FF-AA reduced boundary vectors can be integrated with one of the normalization techniques effectively when the accuracy and precision are prioritized, (2) the final training set having only the boundary vectors can be integrated with one of the normalization techniques effectively when the recall and F1-score are prioritized, (3) the Z-score normalization can generally improve the accuracy and precision of all types of training sets, (4) the modified hyperbolic tangent normalization can generally improve the recall of all types of training sets, (5) the min-max normalization can generally improve the accuracy and F1-score of all types of training sets, and (6) the selection of the normalization techniques and the training set types depends on the key performance measure for the dataset.

Key words: *data mining, data normalization, data reduction, instance selection, neural network*

Received: November 18, 2018
Revised and accepted: April 30, 2021

DOI: 10.14311/NNW.2021.31.007

[*]Piyabute Fuangkhon; Department of Digital Business Management, Assumption University, 88 Moo 8 Bangna-Trad K.M.26, Bang Sao Thong, Samut Prakan 10540 Thailand, E-mail: piyabutefng@au.edu, piyabute@hotmail.com

©CTU FTS 2021

1. Introduction

Machine learning, a computerized learning method, automates analytical model building without following step-by-step algorithms. Supervised learning is one of the categories of machine learning tasks. A computing unit is presented with the inputs and the preset outputs. The goal is to learn a rule that maps the inputs to corresponding outputs. Example supervised learning models include artificial neural network, k-nearest neighbor algorithm [1], naïve Bayes classifier [2], and support vector machine [3]. Artificial neural network (ANN) is widely used in many real-world applications that involve pattern identification, regression analysis, classification, data processing. A feed-forward neural network (FFNN) is an ANN wherein connections in consecutive layers between the neurons never form a cycle. Factors influencing the accuracy of this class prediction model include the training parameter adjustment and the quality of the training set. Regarding the training parameter adjustment, the neural network repeatedly adjusts the weights between consecutive neurons to reduce the value of the error function by some small amount. Relating to the quality, the representativeness of the training set is a necessary condition for an excellent neural network generalization.

Many techniques were proposed to improve the distribution model of the data by assisting the neural network to classify the problem that is linearly separable non-linearly. Researches [4–6] are based on barrier detection. Researches [7, 8] are based on cluster centroid detection. Nevertheless, these techniques do not conserve the distribution model of the samples accurately. Multi-class contour preserving classification (MCOV generator) [9] is a data barrier amplifying technique based on Euclidean distance function that generates the boundary vectors to help preserve and improve the distribution model of the samples for the FFNN. It generates two forms of boundary vectors; one for amplifying the barrier of their class (fundamental multi-class outpost vectors) and the other for increasing the barrier of the nearest class (additional multi-class outpost vectors). Nevertheless, the number of generated fundamental multi-class outpost vectors and additional multi-class outpost vectors in the training set is tremendous, resulting in significantly prolonged training time. Reduced multi-class contour preserving classification (RMCOV generators) [10] adapted [9] by scaling down the sets of fundamental multi-class outpost vectors and additional multi-class outpost vectors. It proposes three reduced boundary vector generators to scale down these sets of boundary vectors; (1) FAF-AFA reduced boundary vector generator (FF-AA RMCOV), (2) FF-AF reduced boundary vector generator (FF-AA RMCOV), and (3) FA-AF reduced boundary vector generator (FAF-AFA RMCOV), that apply *one-to-rest* strategy.

When the boundary vector generator and the reduced boundary vector generators were applied to the real-world datasets, the problem regarding the attribute intervals and units was encountered. It is known that real-world datasets occasionally have multiple attributes. Some attributes have smaller intervals and units than others. Expressing the attributes in imbalanced intervals and measurement units may give some attributes a stronger weight or effect than the others on the Euclidean distance calculation. In other words, these attributes may suppress the other attributes on the Euclidean distance calculation. Therefore, the training set should be processed by altering all attributes to fall within a regular interval such as [0.0, 1.0] or [-1.0, 1.0] before applying the Euclidean distance function to avoid reliance on the choice of intervals and measurement units. This alteration, called *normalization*, helps prevent the attributes with initially large intervals (e.g., income, credit card limit) from overweighing or prevailing the attributes with initially smaller intervals (e.g., binary or ternary attributes). Therefore, it can alleviate the issue above. Besides, normalization is helpful for classification algorithms involving neural networks because it helps speed up the learning phase, resulting in an improved computational capability.

The objective of this research is to study the integration of each of the six normalization techniques; (1) min-max normalization, (2) Z-score normalization, (3) mean and mean absolute deviation normalization, (4) median and median absolute deviation normalization, (5) modified hyperbolic tangent normalization, and (6) hyperbolic tangent estimator normalization, with the boundary vector generator [9] and the reduced boundary vector generators [10] for maximizing class boundary. Its goal is to reveal whether any normalization techniques can improve the classification performance of the original data boundary amplifier for the feed-forward neural network with an emphasis on accuracy, precision, recall, and F1-score. The proposed methodology adds a data normalization as a new step before the data set will be reduced by the boundary vector generators and the reduced boundary vector generators.

This paper presents the following major contributions:

- The final training set having only FF-AA reduced boundary vectors can be integrated with one of the normalization techniques effectively when the accuracy and precision are prioritized.
- The final training set having only the boundary vectors can be integrated with one of the normalization techniques effectively when the recall and F1-score are prioritized.
- The Z-score normalization can generally improve the accuracy and precision of all types of training sets.
- The modified hyperbolic tangent normalization can generally improve the recall of all types of training sets.
- The min-max normalization can generally improve the accuracy and F1-score of all types of training sets.
- The selection of the normalization techniques and the training set types depends on the key performance measure for the dataset.

After this section, Section 2 summarizes the preliminaries. Section 3 explains the experimental methodology and parameters. Section 4 shows the experimental results carried out on the real-world datasets from the UCI database and the ELENA project and discussion. Section 5 presents the conclusions.

2. Preliminaries

This section strives to present the research works related to this study. It comprises the research on contour preserving classification [4], the research on boundary vector generator [9], and the research on reduced boundary vector generators [10] which are presented in Sections 2.1, 2.2, and 2.3, respectively. Besides, the six normalization techniques are briefly summarized in Sections 2.4, 2.5, 2.6, 2.7, 2.8, and 2.9.

2.1 Contour preserving classification

Contour preserving classification (COV) [4] increases the fault tolerance and robustness of the neural network. The technique assists the neural network to non-linearly classify the linearly solvable problem to supplement the leeway between the hyperplane and samples. The technique generates two new forms of boundary vectors; one is named fundamental outpost vector (defined by FOV), and the other is named additional outpost vector (defined by AOV), which are located at the center between the decision barrier of consecutive data of different classes for improving the distribution model of the samples having two classes. The FOV declares the decision barrier between a sample in one class and the nearest sample in another class. In comparison, the AOV declares the decision barrier between a sample belonging to one class and the paired vector of that sample. These new samples are generated at the middle between the decision barrier of consecutive data of different classes to improve the distribution model of the samples and aid the neural network to classify a linearly solvable problem non-linearly.

FOVs, AOVs, sample's barrier, and sample's counter barrier are depicted in Fig. 1 using a 2D two-class problem. Solid circles represent the decision barrier of a sample that clarifies the terrain of the sample for setting its FOV and the AOV of its pair vector. Dot circles represent the counter barrier of a paired vector that helps clarify the terrain of the paired vector versus the terrain of a sample in the reversed direction. The counter barrier can amplify clearance between the decision barrier of a paired vector and the decision barrier of a sample in the reversed direction only.

2.2 Multi-class contour preserving classification

One of the significant problems in contour preserving classification in Section 2.1 is its inability to cope with multi-class data. This section summarizes the multi-class outpost vector generation technique [9], a data barrier amplifying technique, which is a boundary vector generator that does conserve the distribution model of the data and aid the neural network to classify the linearly solvable problem non-linearly. Its goal is to supplement the leeway between the hyperplane and the samples. It improves [4] to make it support data with two or more classes. This method generates new forms of boundary vectors; one is named fundamental multi-class outpost vector (delineated as FMCOV), the other is named additional multi-class outpost vector (delineated as AMCOV). Both FMCOVs and AMCOVs are located at the decision barrier of data in different classes to improve the data

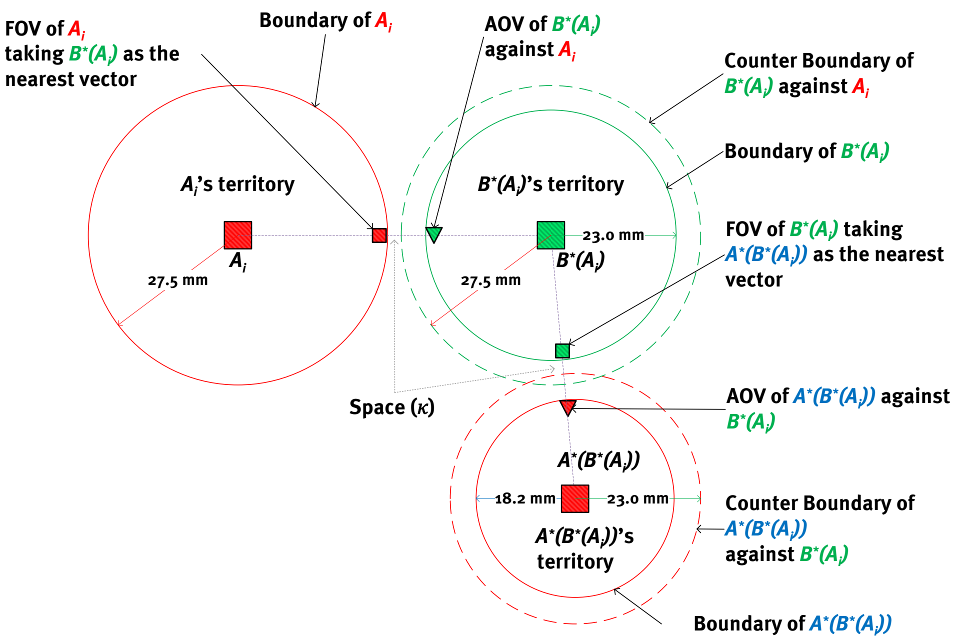


Fig. 1 *FOVs are represented with small rectangles. AOVs are represented with small triangles. Sample's barrier is represented by a solid circle. Sample's counter barrier is represented by dash circle.*

distribution model with two or more classes. The anatomies of both FMCOV and AMCOV generators are described as follows.

- **FMCOV generator** generates a new boundary vector to declare the decision barrier between a sample in one class and the nearest sample from another class. Assume A_i is a sample i in class A, B_j is a sample j in class B, and $A \cap B = \emptyset$. The terrain of A_i is established by spotting B_j that is nearest to A_i. B_j is a pair vector of A_i (defined by $\phi(i)$). Then, the terrain of A_i is declared in the middle between A_i and B_j. Then, the counter terrain of B_j is also declared in the middle between A_i and B_j too. Consequently, the radius of the terrain of A_i is set at the center of the distance between A_i and B_j. This radius assures that the distance from either A_i or B_j to the hyperplane will be at the maximum if B_j sets its terrain using the same radius. Finally, A_i's FMCOV (defined by $o(A_i)$ is generated at the decision barrier of A_i's in reversed direction to B_j. Space (defined by κ) is added at the decision barrier between the terrain of A_i and the counter terrain of B_j to give a little leeway between FMCOVs and AMCOVs in diverse classes.

- **AMCOV generator** generates a new boundary vector to declare the decision barrier between a sample in one class and the paired vector of that sample. Assume A_i is a sample i in class A, while B_j is A_i's pair vector. A_i's AMCOV (defined by $o'(i)$) is generated at the decision barrier of B_j

in reversed direction to A_i. Regarding the territories of A_i and B_j, the radius of the feature s of A_i's pair vector (defined by $\bar{r}(\phi(i)_s)$) is adapted for synthesizing feature s of A_i's AMCOV.

FMCOVs, AMCOVs, sample's barrier, and sample's counter barrier are depicted in Fig. 2 using a 2D three-class problem. Solid circles represent the decision barrier of a sample that clarifies the terrain of the sample for setting its pair vector's FMCOV and AMCOVs. Dot circles represent the paired vector's counter barrier that helps clarify the terrain of the paired vector versus the terrain of a sample in the reversed direction. The counter barrier can amplify the clearance between the decision barriers of a paired vector and a sample in the reversed direction only.

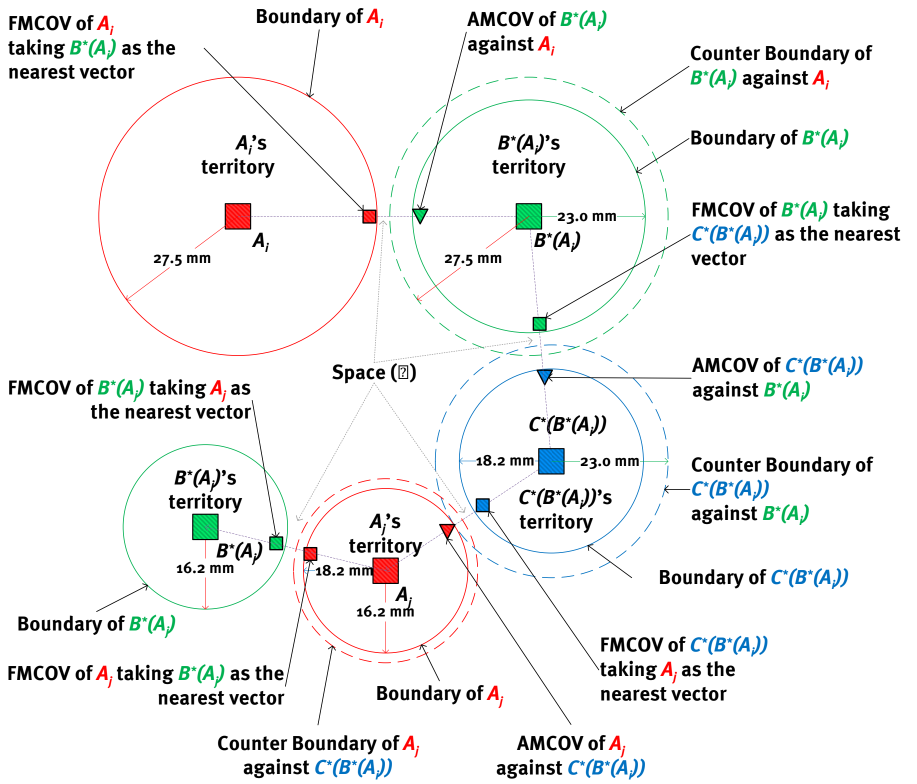


Fig. 2 *FMCOVs are represented with small rectangles. AMCOVs are represented with small triangles. Sample's barrier is represented by a solid circle. Sample's counter barrier is represented by dash circle.*

The experimental results on the real-world datasets from the UCI database confirmed that the boundary vector generator could improve the performance of the classification of the feed-forward neural network on most datasets [9]. In contrast, the report excluded analysis of meaningful performance measures such as precision, recall, and F1-score.

2.3 Reduced multi-class contour preserving classification

One of the significant problems in the boundary vector generator in Section 2.2 is the enormous size of the sets of boundary vectors. This section summarizes the reduced boundary vector generators [10] which present three boundary vector reduction methods for the boundary vectors generated by [9] with *one-to-rest* strategy. Section 2.3.1 presents the FF-AA reduced boundary vector generator (FF-AA RMCOV generator). Section 2.3.2 presents the FA-AF reduced boundary vector generator (FA-AF RMCOV generator). Section 2.3.3 presents the FAF-AFA reduced boundary vector generator (FAF-AFA RMCOV generator).

2.3.1 FF-AA reduced boundary vector generator

This method (FF-AA RMCOV generator) generates a set of reduced boundary vectors (RMCOVs) by FF RMCOV generator, which processes an FMCOV in a class versus all FMCOVs in other classes, and AA RMCOV generator, which processes an AMCOV in a class versus all AMCOVs in other classes. Fig. 3 highlights FF-AA RMCOV generator's framework. This method finds a minimum Euclidean distance between FMCOV in a class (defined by S) and FMCOVs in other classes

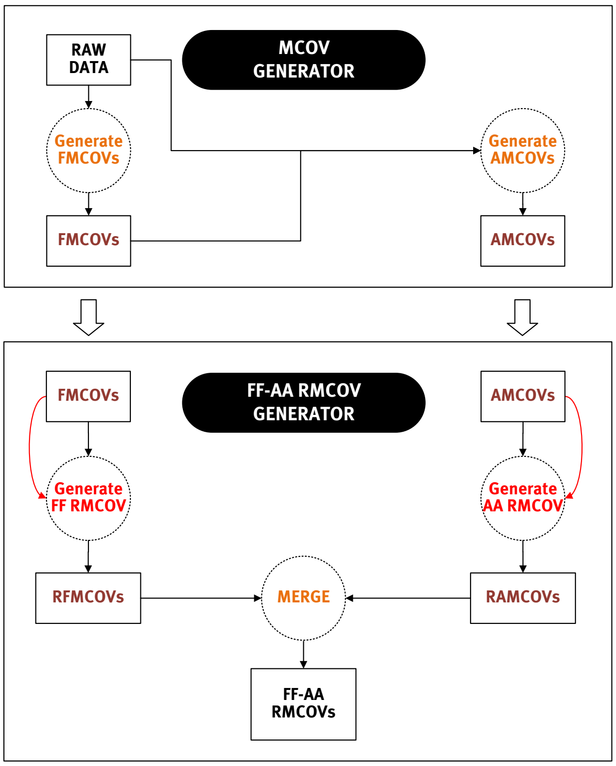


Fig. 3 *A framework of FF-AA reduced boundary vector generator.*

(defined by D). The AA RMCOV generator finds a minimum Euclidean distance between AMCOV in a class (defined by S) and AMCOVs in other classes (defined by D). Both FF RMCOV and AA RMCOV generators combine all Ds to form a set of RMCOVs.

1. **FF RMCOV generator** calculates a Euclidean distance between FMCOV in a class and FMCOVs in other classes. Assume $o(i)$ is i's FMCOV and $o(j)$ is j's FMCOV, $o(j)$ which has a minimum distance to $o(i)$ is denoted by RFMCOV (defined by $o_r(i)$).

2. **AA RMCOV generator** calculates a Euclidean distance between AMCOV in a class and AMCOVs in other classes. Assume $o'(i)$ is i's AMCOV and $o'(j)$ is j's AMCOV, $o'(j)$ which has a minimum distance to $o'(i)$ is denoted by RAMCOV (defined by $o'_r(i)$).

Both RMCOVs are uniquely consolidated to generate an FF-AA RMCOV set. The class of $o_r(i)$ equals the class of $o(j)$. The class of $o'_r(i)$ equals the class of $o'(j)$.

2.3.2 FA-AF reduced boundary vector generator

This method (FA-AF RMCOV generator) generates a set of reduced boundary vectors (RMCOVs) by FA RMCOV generator, which processes an FMCOV in a class versus all AMCOVs in other classes, and AF RMCOV generator, which processes an AMCOV in a class versus all FMCOVs in other classes. Fig. 4 highlights FA-AF RMCOV generator's framework. This method finds a minimum Euclidean distance between FMCOV in a class (defined by S) and AMCOVs in other classes (defined by D). The AF RMCOV generator finds a minimum Euclidean distance between AMCOV in a class (defined by S) and FMCOVs in other classes (defined by D). Both FA RMCOV and AF RMCOV generators combine all Ds to form a set of RMCOVs.

1. **FA RMCOV generator** calculates a Euclidean distance between FMCOV in a class and AMCOVs in other classes. Assume $o(i)$ is i's FMCOV and $o'(j)$ is j's AMCOV, $o'(j)$ which has a minimum distance to $o(i)$ is denoted by RAMCOV (defined by $o'_r(i)$).

2. **AF RMCOV generator** calculates a Euclidean distance between AMCOV in a class and FMCOVs in other classes. Assume $o'(i)$ is i's AMCOV and $o(j)$ is j's FMCOV, $o(j)$ which has a minimum distance to $o'(i)$ is denoted by RFMCOV (defined by $o_r(i)$).

Both RMCOVs are uniquely consolidated to generate an FA-AF RMCOV set. The class of $o'_r(i)$ equals the class of $o'(j)$. The class of $o_r(i)$ equals the class of $o(j)$.

2.3.3 FAF-AFA reduced boundary vector generator

This method (FAF-AFA RMCOV) generates reduced boundary vectors (RMCOVs) by FAF RMCOV generator, which processes an FMCOV in a class versus all

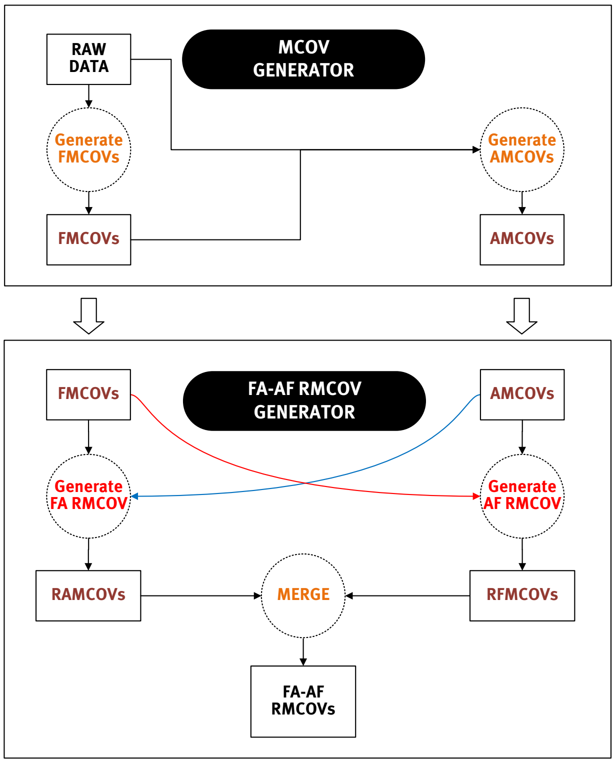


Fig. 4 *A framework of FA-AF reduced boundary vector generator.*

MCOVs in other classes, and AFA RMCOV generator, which processes an AMCOV in a class versus all MCOVs in other classes. Fig. 5 highlights FAF-AFA RMCOV generator's framework. This method finds a minimum Euclidean distance between FMCOV in a class (defined by S) and MCOVs in other classes (defined by D). The AFA RMCOV generator finds a minimum Euclidean distance between AMCOV in a class (defined by S) and MCOVs in other classes (defined by D). Both FAF RMCOV and AFA RMCOV generators combine all Ds to form a set of RMCOVs.

1. **FAF RMCOV generator** calculates a Euclidean distance between FMCOV in a class and AMCOVs and FMCOVs in other classes. Assume $o(i)$ is i's FMCOV, $o'(j)$ is j's AMCOV, $o(k)$ is FMCOV of k, either $o'(j)$ or $o(k)$ which has a minimum distance to $o(i)$ is denoted by RFMCOV (defined by $o_r(i)$).

2. **AFA RMCOV generator** calculates a Euclidean distance between AMCOV in a class and FMCOVs and AMCOVs in other classes. Assume $o'(i)$ is i's AMCOV, $o(j)$ is j's FMCOV, $o'(k)$ is AMCOV of k, either $o(j)$ or $o'(k)$ which has a minimum distance to $o'(i)$ is denoted by RAMCOV (defined by $o'_r(i)$).

Both RMCOVs are uniquely consolidated to generate a FAF-AFA RMCOV set.

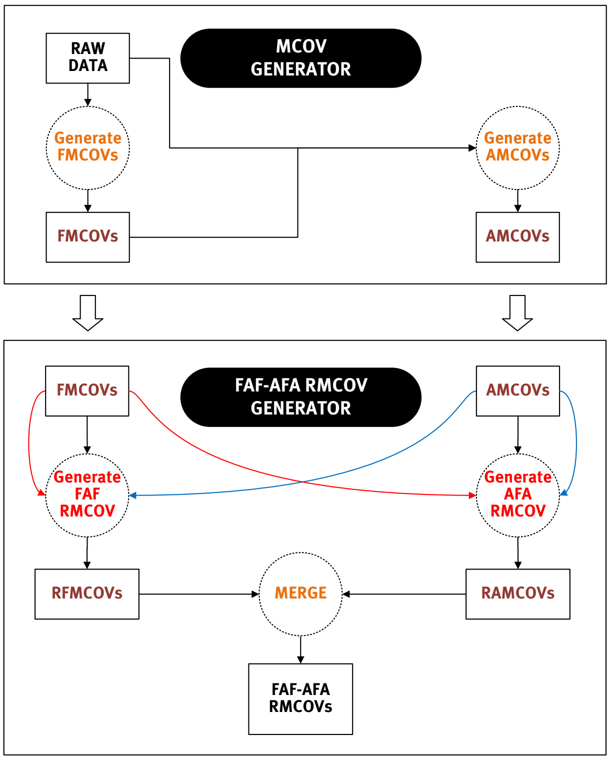


Fig. 5 *A framework of FAF-AFA reduced boundary vector generator.*

The class of $o_r(i)$ equals the class of either $o(j)$ or $o'(k)$ (depending on the sample). The class of $o'_r(i)$ equals the class of either $o'(j)$ or $o(k)$ (depending on the sample).

The complete results assured that all reduced boundary vector generators significantly reduced the set of fundamental multi-class outpost vectors and the set of additional multi-class outpost vectors [10].

2.4 Min-max normalization

Min-max normalization [11] is a data scaling technique that executes a linear transformation on the given samples while conserving the relationships among the sample values. As a result, the distribution model of the sample distribution will not change. Eq. (1) describes the min-max normalization.

$$v'_i = \left(\frac{v_i - \min_i}{\max_i - \min_i} \times (\max'_i - \min'_i) \right) + \min'_i. \quad (1)$$

Given that $\min_i$ and $\max_i$ are the minimum value and maximum value of an attribute i, the min-max normalization transforms a value v of an attribute i (denoted by v_i) to a value in the new interval $[\min_i', \max_i']$ (denoted by v_i') (e.g., from interval $[1, 100]$ to interval $[0.0, 1.0]$).

2.5 Z-score normalization

Z-score normalization or standard score normalization [11] is a data scaling technique that executes a linear transformation on the given samples based on the mean and standard deviation. Eq. (2) describes the Z-score normalization.

$$v_i' = \frac{v_i - \mu_i}{\sigma_i}, \tag{2}$$

where v_i is the value in attribute i of point v, μ_i is the mean value of an attribute i, and σ_i is the standard deviation value of an attribute i.

The standard score is helpful when the actual minimum value and the actual maximum value of attribute i are unknown or when some outliers prevail the min-max normalization.

2.6 Mean and mean absolute deviation normalization

Mean absolute deviation (MAD) measures the variability of a univariate sample of quantitative data. It is the average of the absolute value of the deviations from the mean. Mean MAD [12] is an absolute deviation technique that is insensitive to outliers where the points are at the extreme ends of the distribution. As a result, a normalization using mean MAD is robust. Eq. (3) describes the mean absolute deviation ($\mathrm{MAD}_{\mathrm{mean}_i}$) of an attribute i. Eq. (4) describes the mean and mean absolute deviation normalization.

$$\mathrm{MAD}_{\mathrm{mean}_i} = \frac{1}{n}\sum_{x=1}^{n} |v_x - \mu_i|, \tag{3}$$

$$v_i' = \frac{v_i - \mu_i}{\mathrm{MAD}_{\mathrm{mean}_i}}, \tag{4}$$

where v_i is the value in attribute i of point v, n is the number of points, and μ_i is the mean value of an attribute i.

2.7 Median and median absolute deviation normalization

Median absolute deviation (MedAD) measures the variability of a univariate sample of quantitative data. It is the middle of the absolute value of the deviations from the median. Median MedAD [12] is an absolute deviation technique that is also insensitive to outliers. As a result, a normalization using Median MedAD is robust. Eq. (5) describes the median absolute deviation ($\mathrm{MedAD}_{\mathrm{median}}$) of an attribute i. Eq. (6) describes the median and median absolute deviation normalization.

$$\mathrm{MedAD}_{\mathrm{median}_i} = \mathrm{median}(|v_i - \mathrm{median}_i|), \tag{5}$$

$$v_i' = \frac{v_i - \text{median}_i}{\text{MedAD}_{\text{median}_i}}, \tag{6}$$

where v_i is the value in attribute i of point v and median_i is the median value of an attribute i.

2.8 Hyperbolic tangent estimator normalization

Hyperbolic tangent estimator [13] has been used in normalization [14]. This technique is insensitive to outliers. Eq. (7) describes the hyperbolic tangent estimator normalization.

$$v_i' = \frac{1}{2} \times \left(\tanh \left(0.01 \times \frac{v_i - \mu_{GH}}{\sigma_{GH}} \right) + 1 \right). \tag{7}$$

μ_{GH} is the mean value of the genuine score distribution. σ_{GH} is the standard deviation value of the genuine score distribution.

2.9 Modified hyperbolic tangent normalization

Given the complexity of the genuine score calculation given by Hampel estimator [13], the hyperbolic tangent estimator was modified [12] to avoid this complicated calculation. Eq. (8) describes the modified hyperbolic tangent normalization that omits the Hampel calculation.

$$v_i' = \frac{1}{2} \times \left(\tanh \left(0.01 \times \frac{v_i - \mu_i}{\sigma_i} \right) + 1 \right), \tag{8}$$

where v_i is the value in attribute i of point v, μ_i is the mean value of an attribute i, and σ_i is the standard deviation value of an attribute i.

3. Methodology

This section presents the implementation of each of the six normalization techniques; min-max normalization, Z-score normalization, mean and mean absolute deviation, median and median absolute deviation, modified hyperbolic tangent normalization, and hyperbolic tangent estimator normalization, with the boundary vector generator [9] mentioned in Section 2.2 and the three reduced boundary vector generators [10] mentioned in Sections 2.3.1, 2.3.2, and 2.3.3. This experiment aims to improve the classification performance of the feed-forward neural network on real-world datasets after each of the six normalization techniques pre-processes the original training set of all boundary vector generators.

Regarding the data, ten real-world datasets were picked from the UCI database [15] and the ELENA Project database [16]. They consisted of Default of Credit Card Clients[1], Bank Marketing[2], Nursery[3], Optical Recognition of Handwritten

[1] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[2] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[3] https://archive.ics.uci.edu/ml/datasets/Nursery

Digits[4], Poker Hand[5], Pen-Based Recognition of Handwritten Digits[6], Statlog (Shuttle)[7], Statlog (Landsat Satellite)[8], Phoneme[9], and Texture[10]. Diverse numbers of classes, attributes, and class balance were considered during dataset selection. The components of these datasets are shown in Tab. I. Tabs. II and III present the class balance of the training set and the testing set of all datasets. Three datasets (D4, D5, D10) were a balanced class; while the others were imbalanced class. All nominal attributes were transformed into numeric attributes by MATLAB *grp2idx* function. The test sets were divided from the dataset by MATLAB *dividerand* function.

Symbol	Dataset	Attribute Characteristics	Classes	Features	Training Samples	Test Samples
D1	Bank Marketing	Integer	2	20	4,119	41,188
D2	Default of Credit Card	Integer	2	23	21,000	9,000
D3	Nursery	Categorical	5	8	9,072	3,888
D4	Optical Recognition	Integer	10	64	3,823	1,797
D5	Pen-Based Recognition	Integer	10	16	7,494	3,498
D6	Poker Hand	Integer/ Categorical	10	10	25,010	1,000,000
D7	Statlog (Landsat Sat.)	Integer	6	36	4,435	2,000
D8	Statlog (Shuttle)	Integer	7	9	43,500	14,500
D9	Phoneme	Real	2	5	4,053	1,351
D10	Texture	Real	11	40	5,500	1,375

Tab. I *Characteristics of the selected real-world datasets.*

3.1 Normalization

For each dataset, its original training set was normalized by each of the following six normalization techniques.

[4] https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[5] https://archive.ics.uci.edu/ml/datasets/Poker+Hand

[6] https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits

[7] https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)

[8] https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

[9] https://www.elen.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/phoneme/

[10] https://www.elen.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/texture/

1. N0 = No normalization
2. N1 = Min-max normalization
3. N2 = Z-score normalization
4. N3 = Mean and mean absolute deviation normalization
5. N4 = Median and median absolute deviation normalization
6. N5 = Modified hyperbolic tangent normalization
7. N6 = Hyperbolic tangent estimator normalization

The output of each normalization is called the normalized original training set (defined by NORG). To simplify the representation of the output of this step, the term "normalized original training set" is also used even when no normalization is applied (N0).

Class	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1	89.05	77.50	00.01	09.84	10.41	49.95	23.40	00.03	70.49	08.82
2	10.95	22.50	32.62	10.18	10.39	42.38	24.17	78.41	29.51	09.50
3			33.60	09.94	10.41	04.82	10.80	00.09		09.04
4			02.63	10.18	09.59	02.05	21.67	00.30		08.82
5			31.14	10.12	10.41	00.37	09.36	15.51		09.19
6				09.84	09.61	00.22	10.60	05.65		09.04
7				09.86	09.61	00.14		00.01		09.58
8				10.12	10.38	00.02				08.73
9				09.94	09.59	00.02				09.16
10				09.99	09.59	00.02				08.90
11										09.21

Tab. II *The class balance of the original training set of all datasets.*

Class	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1	88.73	78.78	00.03	09.91	10.38	50.12	23.50	00.01	71.13	09.89
2	11.27	21.22	33.62	10.13	10.41	42.25	23.05	79.16	28.87	07.85
3			32.72	09.85	10.41	04.76	11.20	00.09		09.24
4			02.29	10.18	09.61	02.11	19.85	00.27		09.89
5			31.35	10.07	10.41	00.39	10.55	14.86		08.80
6				10.13	09.58	00.20	11.85	05.58		09.24
7				10.07	09.61	00.14		00.03		07.64
8				09.96	10.41	00.02				10.18
9				09.68	09.61	00.00				08.87
10				10.02	09.61	00.00				09.67
11										08.73

Tab. III *The class balance of the test set of all datasets.*

3.2 Boundary vector generators

Each normalized original training set produced in Section 3.1 was processed to synthesize a set of boundary vectors (defined by OV1) and three sets of reduced boundary vectors (defined by OV2, OV3, OV4).

1. OV1 = a set of fundamental multi-class outpost vectors and additional multi-class outpost vectors ($\kappa = 5\%$).

2. OV2 = a set of FF-AA reduced boundary vectors (FF-AA RMCOVs) ($\kappa = 5\%$).

3. OV3 = a set of FA-AF reduced boundary vectors (FA-AF RMCOVs) ($\kappa = 5\%$).

4. OV4 = a set of FAF-AFA reduced boundary vectors (FAF-AFA RMCOVs) ($\kappa = 5\%$).

3.3 Final training set generation

The sets of boundary vectors (OV1) and reduced boundary vectors (OV2, OV3, OV4) in Section 3.2 were combined with the normalized original training set produced in Section 3.1 to construct eight final training sets (defined by T1, T2, T3, T4, T5, T6, T7, T8).

1. T1 = a set of the normalized original training set and all boundary vectors (MCOVs).

2. T2 = a set of all boundary vectors (MCOVs).

3. T3 = a set of the normalized training set and all FF-AA reduced boundary vectors (FF-AA RMCOVs).

4. T4 = a set of the normalized original training set and all FA-AF reduced boundary vectors (FA-AF RMCOVs).

5. T5 = a set of the normalized original training set and all FAF-AFA reduced boundary vectors (FAF-AFA RMCOVs).

6. T6 = a set of all FF-AA reduced boundary vectors (FF-AA RMCOVs).

7. T7 = a set of all FA-AF reduced boundary vectors (FA-AF RMCOVs).

8. T8 = a set of all FAF-AFA reduced boundary vectors (FAF-AFA RMCOVs).

Tab. IV presents the components of all final training sets produced by each of the six normalization techniques.

Final Training Set	NORG	OV1	OV2	OV3	OV4
T1	✓	✓	-	-	-
T2	-	✓	-	-	-
T3	✓	-	✓	-	-
T4	✓	-	-	✓	-
T5	✓	-	-	-	✓
T6	-	-	✓	-	-
T7	-	-	-	✓	-
T8	-	-	-	-	✓

Tab. IV *Components of the eight final training sets.*

3.4 Training

Regarding the classifier, the experiments were implemented with:

- Application: MATLAB R2017a
- Toolboxes: Neural Network Toolbox, Signal Processing Toolbox, Statistics and Machine Learning Toolbox, Parallel Computing Toolbox
- Model: Feed-Forward Neural Network
- Hidden layer = 1
- Hidden neurons = 20, 23, 8, 64, 16, 10, 36, 9 (depend on the number of features of each dataset)
- Output neurons = 2, 2, 5, 10 10, 10, 6, 7 (depend on the number of classes of each dataset)
- Training function = Levenberg-Marquardt Backpropagation
- Transfer function = Log-sigmoid or Tan-sigmoid (depend on the type of the attributes of each dataset)
- Maximum epochs = 1,000
- Performance Goal = 0.01
- Minimum Gradient: 0.0000001
- Maximum Validation Checks: 100
- Mu: 0.001
- Mu Decrease Ratio: 0.1
- Mu Increase Ratio: 10
- Maximum mu: 10,000,000,000

The experimental results were evaluated with four performance measures [17]; *Accuracy*, *Recall*, *Precision*, and F1-score, to confirm the classification performance of all boundary vector generators when applied with and without each of the six normalization techniques. *Accuracy* is a comprehensive performance of the classifier (described in Eq. (9)). *Precision* is the actual ratio of the number of the correctly classified positive samples to the number of samples labeled as positive by the classifier. It is a good measure to determine when the costs of false-positive are high (described in Eq. (10)). *Recall* is the ratio of the number of correctly classified positive samples to the number of positive samples. It is a good measure to determine when there is a high cost associated with a false negative (described in Eq. (11)). F-score is the weighted harmonic mean of the recall and precision. It is a good measure to determine when a balance between Precision and Recall is needed, and there is an uneven class distribution (described in Eq. (12)). The β was initialized to one because the recall and precision were equally weighted, called F1-score.

$$\text{Accuracy} = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}, \tag{9}$$

$$\text{Precision} = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}, \tag{10}$$

$$\text{Recall} = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}, \tag{11}$$

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\beta \times \text{Precision} + \text{Recall}}, \tag{12}$$

where tp_i is the number of the positive samples classified in class i correctly, fp_i is the number of the positive samples classified in class i incorrectly, tn_i is the number of the negative samples classified in class i correctly, fn_i is the number of the negative samples classified in class i incorrectly, and l is the number of classes of data.

4. Experimental results and discussion

This section presents the classification performance of the boundary vector generator and the reduced boundary vector generators in Section 3 when the original training set was normalized with min-max normalization, Z-score normalization, mean and mean absolute deviation normalization, median and median absolute deviation normalization, modified hyperbolic tangent normalization, and hyperbolic tangent estimator normalization. Sections 4.1, 4.2, 4.3, and 4.4 report the experimental results in terms of accuracy, precision, recall, and F1-score, respectively.

For each final training set, the training and evaluation were conducted ten times to avoid the effect of the random initial weights of the feed-forward neural network. The experimental results present the averaged values of the accuracy, precision, recall, and F1-score.

4.1 Accuracy

Tab. V reports the accuracy of the boundary vector generator and the reduced boundary vector generators without normalization (N0) and with normalization (N1-N6) on all datasets (D1-D10). Bold values indicate that the specified normalization technique can improve the accuracy of the related boundary vector generator or the related reduced boundary vector generators.

Regarding the boundary vector generator (T1, T2), the normalization techniques that mostly improved the accuracy of the feed-forward neural network on:

- The T1 final training set (the final training set having the normalized original training set and the boundary vectors) included the min-max normalization (N1) and Z-score normalization (N2). They yielded a higher accuracy on 6 datasets than those without normalization.

- The T2 final training set (the final training set having only the boundary vectors) included the Z-score normalization (N2) and mean and mean absolute deviation normalization (N3). They yielded a higher accuracy on 7 datasets than those without normalization.

In sum, the Z-score normalization (N2) was the generalized technique that could best improve the accuracy of the boundary vector generator.

Regarding the reduced boundary vector generators (T3, T4, T5, T6, T7, T8), the normalization technique(s) that mostly improved the accuracy of the feed-forward neural network on:

- The T3 final training set (the final training set having the normalized original training set and the FF-AA reduced boundary vectors) included the min-max normalization (N1). It yielded a higher accuracy on 5 datasets than those without normalization.

- The T4 final training set (the final training set having the normalized original training set and the FA-AF reduced boundary vectors) included the mean and mean absolute deviation normalization (N3) and modified hyperbolic tangent normalization (N5). They yielded a higher accuracy on 5 datasets than those without normalization.

- The T5 final training set (the final training set having the normalized original training set and the FAF-AFA reduced boundary vectors) included the Z-score normalization (N2). It yielded a higher accuracy on 6 datasets than those without normalization.

- The T6 final training set (the final training set having only the FF-AA reduced boundary vectors) included the mean and mean absolute deviation normalization (N3). It yielded a higher accuracy on 8 datasets than those without normalization.

- The T7 final training set (the final training set having only the FA-AF reduced boundary vectors) included the min-max normalization (N1). It yielded a higher accuracy on 6 datasets than those without normalization.

- The T8 final training set (the final training set having only the FAF-AFA reduced boundary vectors) included the median and median absolute deviation normalization (N4). It yielded a higher accuracy on 6 datasets than those without normalization.

In sum, the min-max normalization (N1) and Z-score normalization (N2) were the generalized technique that could mainly improve the accuracy of all reduced boundary vector generators.

4.2 Precision

Tab. VI reports the precision of the boundary vector generator and the reduced boundary vector generators without normalization (N0) and with normalization (N1-N6) on all datasets (D1-D10). Bold values indicate that the specified normalization technique can improve the precision of the related boundary vector generator or the related reduced boundary vector generators.

Regarding the boundary vector generator (T1, T2), the normalization techniques that improved the precision of the feed-forward neural network on:

- The T1 final training set (the final training set having the normalized original training set and the boundary vectors) included the mean and mean absolute deviation normalization (N3). It yielded a higher precision on 6 datasets than those without normalization.
- The T2 final training set (the final training set having only the boundary vectors) included the min-max normalization (N1), Z-score normalization (N2), and mean and mean absolute deviation normalization (N3). They yielded a higher precision on 5 datasets than those without normalization.

In sum, the mean and mean absolute deviation normalization (N3) was the generalized technique that could best improve the precision of the boundary vector generator.

Regarding the reduced boundary vector generators (T3, T4, T5, T6, T7, T8), the normalization technique that improved the precision of the feed-forward neural network on:

- The T3 final training set (the final training set having the normalized original training set and the FF-AA reduced boundary vectors) included the min-max normalization (N1). It yielded a higher precision on 7 datasets than those without normalization.
- The T4 final training set (the final training set having the normalized original training set and the FA-AF reduced boundary vectors) included the Z-score normalization (N2) and mean and mean absolute deviation normalization (N3). They yielded a higher precision on 8 datasets than those without normalization.
- The T5 final training set (the final training set having the normalized original training set and the FAF-AFA reduced boundary vectors) included the Z-score normalization (N2). It yielded a higher precision on 7 datasets than those without normalization.

- The T6 final training set (the final training set having only the FF-AA reduced boundary vectors) included the mean and mean absolute deviation normalization (N3), modified hyperbolic tangent normalization (N5), and hyperbolic tangent estimator normalization (N6). They yielded a higher precision on 8 datasets than those without normalization.

- The T7 final training set (the final training set having only the FA-AF reduced boundary vectors) included the min-max normalization (N1). It yielded a higher precision on 6 datasets than those without normalization.

- The T8 final training set (the final training set having only the FAF-AFA reduced boundary vectors) included the Z-score normalization (N2). It yielded a higher precision on 6 datasets than those without normalization.

In sum, the Z-score normalization (N2) was the generalized technique that could mainly improve the precision of all reduced boundary vector generators.

4.3 Recall

Tab. VII reports the recall of the boundary vector generator and the reduced boundary vector generators without normalization (N0) and with normalization (N1-N6) on all datasets (D1-D10). Bold values indicate that the specified normalization technique can improve the recall of the related boundary vector generator or the related reduced boundary vector generators.

Regarding the boundary vector generator (T1, T2), the normalization techniques that improved the recall of the feed-forward neural network on:

- The T1 final training set (the final training set having the normalized original training set and the boundary vectors) included the Z-score normalization (N2). It yielded a higher recall on 6 datasets than those without normalization.

- The T2 final training set (the final training set having only the boundary vectors) included the mean and mean absolute deviation normalization (N3), median and median absolute deviation normalization (N4), modified hyperbolic tangent normalization (N5), and hyperbolic tangent estimator normalization (N6). They yielded a higher recall on 6 datasets than those without normalization.

In sum, the Z-score normalization (N2) was the generalized technique that could best improve the recall of the boundary vector generator.

Regarding the reduced boundary vector generators (T3, T4, T5, T6, T7, T8), the normalization technique(s) that improved the recall of the feed-forward neural network on:

- The T3 final training set (the final training set having the normalized original training set and the FF-AA reduced boundary vectors) included the min-max normalization (N1). It yielded a higher recall on 4 datasets than those without normalization.

- The T4 final training set (the final training set having the normalized original training set and the FA-AF reduced boundary vectors) included the mean absolute deviation normalization (N3). It yielded a higher recall on 5 datasets than those without normalization.

- The T5 final training set (the final training set having the normalized original training set and the FAF-AFA reduced boundary vectors) included the Z-score normalization (N2). It yielded a higher recall on 5 datasets than those without normalization.

- The T6 final training set (the final training set having only the FF-AA reduced boundary vectors) included the modified hyperbolic tangent normalization (N5). It yielded a higher recall on 7 datasets than those without normalization.

- The T7 final training set (the final training set having only the FA-AF reduced boundary vectors) included the Z-score normalization (N2), modified hyperbolic tangent normalization (N5), and hyperbolic tangent estimator normalization (N6). They yielded a higher recall on 4 datasets than those without normalization.

- The T8 final training set (the final training set having only the FAF-AFA reduced boundary vectors) included the median and median absolute deviation normalization (N4). It yielded a higher recall on 6 datasets than those without normalization.

In sum, the Z-score normalization (N2) was the generalized technique that could mainly improve the recall of all reduced boundary vector generators.

4.4 F1-score

Tab. VIII reports the F1-score of the boundary vector generator and the reduced boundary vector generators without normalization (N0) and with normalization (N1-N6) on all datasets (D1-D10). Bold values indicate that the specified normalization technique can improve the F1-score of the related boundary vector generator or the related reduced boundary vector generators.

Regarding the boundary vector generator (T1, T2), the normalization techniques that improved the F1-score of the feed-forward neural network on:

- The T1 final training set (the final training set having the normalized original training set and the boundary vectors) included the Z-score normalization (N2) and modified hyperbolic tangent normalization (N5). They yielded a higher F1-score on 6 datasets than those without normalization.

- The T2 final training set (the final training set having only the boundary vectors) included the Z-score normalization (N2). It yielded a higher F1-score on 8 datasets than those without normalization.

In sum, the Z-score normalization (N2) was the generalized technique that could best improve the recall of the boundary vector generator.

Regarding the reduced boundary vector generators (T3, T4, T5, T6, T7, T8), the normalization technique(s) that improved the F1-score of the feed-forward neural network on:

- The T3 final training set (the final training set having the normalized original training set and the FF-AA reduced boundary vectors) included the min-max normalization (N1). It yielded a higher F1-score on 6 datasets than those without normalization.
- The T4 final training set (the final training set having the normalized original training set and the FA-AF reduced boundary vectors) included the min-max normalization (N1). It yielded a higher F1-score on 5 datasets than those without normalization.
- The T5 final training set (the final training set having the normalized original training set and the FAF-AFA reduced boundary vectors) included the Z-score normalization (N2). It yielded a higher F1-score on 6 datasets than those without normalization.
- The T6 final training set (the final training set having only the FF-AA reduced boundary vectors) included mean absolute deviation normalization (N3), median and median absolute deviation normalization (N4), and hyperbolic tangent estimator normalization (N6). They yielded a higher F1-score on 6 datasets than those without normalization.
- The T7 final training set (the final training set having only the FA-AF reduced boundary vectors) included the min-max normalization (N1) and median and median absolute deviation normalization (N4). They yielded a higher F1-score on 6 datasets than those without normalization.
- The T8 final training set (the final training set having only the FAF-AFA reduced boundary vectors) included the median and median absolute deviation normalization (N4). It yielded a higher F1-score on 6 datasets than those without normalization.

In sum, the min-max normalization (N1) and median and median absolute deviation normalization (N4) were the generalized technique that could mainly improve the F1-score of all reduced boundary vector generators.

4.5 Discussion

Tab. IX summarizes the classification performance of the six normalization techniques with the boundary vector generator and the reduced boundary vector generators based on the experimental results reported in Sections 4.1, 4.2, 4.3, and 4.4. Bold values indicate the number of datasets of each experimental setup that yielded the highest accuracy, precision, recall, or F1-score. It is seen that min-max normalization can be recommended as a default function when accuracy and F1-score are prioritized, i.e., the nearness of a measured value to the standard value and the test's accuracy. The Z-score normalization can be a default function when precision is prioritized, i.e., the relevance of selected items. The modified hyperbolic

FTS	Norm	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Total
T1	N0	0.742	0.736	0.824	0.861	0.787	0.846	0.859	0.503	0.681	0.906	0
T1	N1	**0.782**	0.716	0.716	0.860	**0.787**	**0.906**	**0.862**	**0.516**	**0.688**	0.904	**6**
T1	N2	**0.783**	0.708	0.719	0.856	**0.788**	**0.907**	**0.864**	**0.505**	0.679	**0.907**	**6**
T1	N3	**0.770**	0.701	0.795	0.857	0.786	**0.904**	0.854	**0.506**	**0.685**	**0.907**	5
T1	N4	**0.801**	0.691	0.099	0.860	0.784	0.091	0.855	**0.505**	**0.682**	**0.907**	4
T1	N5	**0.789**	0.699	0.719	0.858	0.773	**0.907**	**0.864**	**0.512**	0.672	0.906	4
T1	N6	**0.790**	0.708	0.721	0.856	0.786	**0.906**	0.856	**0.509**	0.673	0.906	3
T2	N0	0.641	0.497	0.876	0.845	0.768	0.894	0.802	0.457	0.581	0.905	0
T2	N1	**0.762**	0.489	0.164	0.842	0.765	0.879	**0.815**	**0.476**	**0.598**	**0.906**	5
T2	N2	**0.779**	**0.528**	0.157	0.845	0.767	**0.904**	**0.814**	**0.477**	**0.595**	**0.905**	**7**
T2	N3	**0.764**	**0.503**	0.185	0.841	0.766	**0.901**	**0.810**	**0.475**	**0.591**	**0.906**	**7**
T2	N4	**0.808**	**0.719**	0.176	0.839	0.767	0.091	0.788	**0.471**	**0.584**	**0.905**	5
T2	N5	**0.782**	0.489	0.171	0.844	0.767	**0.904**	0.798	**0.474**	0.571	**0.905**	4
T2	N6	**0.753**	**0.502**	0.099	0.843	0.765	0.812	**0.818**	**0.477**	**0.594**	**0.906**	6
T3	N0	0.819	0.738	0.873	0.869	0.808	0.906	0.869	0.504	0.752	0.907	0
T3	N1	0.814	**0.745**	0.717	**0.872**	**0.811**	**0.907**	0.868	**0.520**	0.752	0.906	**5**
T3	N2	0.817	**0.746**	0.716	0.869	0.807	**0.907**	0.869	**0.512**	0.751	0.906	3
T3	N3	0.782	0.716	0.794	0.865	0.802	0.900	0.859	**0.513**	0.728	0.886	1
T3	N4	0.798	0.689	0.098	0.866	0.796	0.090	0.853	**0.506**	0.735	0.895	1
T3	N5	0.747	0.720	0.718	0.865	0.804	0.903	0.867	0.500	0.728	0.884	0
T3	N6	0.817	0.697	0.723	0.867	0.779	**0.907**	0.852	**0.509**	0.724	0.896	2
T4	N0	0.819	0.734	0.795	0.862	0.797	0.816	0.852	0.497	0.725	0.896	0
T4	N1	0.754	0.714	0.717	**0.872**	0.793	**0.874**	**0.859**	**0.507**	0.721	0.888	4
T4	N2	0.793	0.704	0.716	**0.867**	0.795	**0.907**	**0.858**	**0.506**	0.718	0.895	4
T4	N3	0.755	0.719	0.791	**0.866**	0.782	**0.904**	**0.865**	**0.499**	**0.744**	0.892	**5**
T4	N4	0.805	0.674	0.107	**0.869**	0.786	0.067	0.850	**0.497**	0.724	0.895	2
T4	N5	0.814	0.713	0.721	**0.870**	0.788	**0.899**	**0.855**	**0.508**	**0.735**	0.890	**5**
T4	N6	0.767	0.710	0.713	**0.869**	0.783	**0.884**	**0.861**	**0.506**	0.709	0.894	4
T5	N0	0.779	0.728	0.782	0.850	0.741	0.815	0.829	0.458	0.569	0.892	0
T5	N1	0.770	0.499	0.722	0.844	**0.742**	**0.885**	**0.851**	**0.495**	**0.575**	0.892	5
T5	N2	0.725	0.469	0.711	0.839	**0.743**	**0.907**	**0.849**	**0.492**	**0.590**	**0.893**	**6**
T5	N3	0.697	0.487	0.782	0.826	0.714	**0.857**	**0.845**	**0.484**	0.538	0.883	3
T5	N4	0.773	0.661	0.104	0.821	0.711	0.021	0.827	**0.477**	0.544	0.881	1
T5	N5	0.719	0.471	0.713	0.822	0.699	**0.834**	**0.837**	**0.497**	0.553	0.884	3
T5	N6	0.681	0.517	0.724	0.828	0.710	**0.905**	**0.841**	**0.495**	0.551	0.886	3
T6	N0	0.381	0.532	0.843	0.818	0.689	0.820	0.753	0.432	0.472	0.740	0
T6	N1	**0.605**	0.453	0.081	0.815	0.681	0.804	**0.759**	**0.456**	0.402	**0.777**	4
T6	N2	**0.513**	0.505	0.199	**0.839**	**0.690**	**0.835**	0.736	**0.452**	0.455	**0.796**	6
T6	N3	**0.617**	0.493	0.165	**0.843**	**0.716**	**0.844**	**0.775**	**0.464**	**0.484**	**0.790**	**8**
T6	N4	**0.806**	**0.716**	0.176	**0.821**	**0.692**	0.091	0.723	**0.456**	**0.480**	**0.798**	7
T6	N5	**0.524**	0.529	0.160	**0.833**	**0.709**	**0.863**	0.730	**0.455**	0.447	**0.809**	6
T6	N6	**0.628**	**0.540**	0.122	**0.834**	**0.730**	0.801	0.738	**0.455**	**0.556**	**0.812**	7
T7	N0	0.533	0.504	0.838	0.828	0.667	0.878	0.754	0.422	0.523	0.754	0
T7	N1	**0.667**	0.478	0.170	0.825	**0.668**	0.696	**0.766**	**0.441**	**0.567**	**0.757**	**6**
T7	N2	0.507	0.369	0.151	**0.841**	0.636	0.780	**0.765**	**0.451**	0.479	**0.762**	4
T7	N3	0.528	0.420	0.179	**0.844**	0.626	0.832	**0.776**	**0.453**	0.476	**0.783**	4
T7	N4	**0.807**	**0.719**	0.186	0.825	0.658	0.091	0.738	**0.447**	0.481	**0.776**	4
T7	N5	**0.535**	**0.574**	0.166	**0.836**	0.617	0.788	0.740	**0.450**	0.484	**0.770**	5
T7	N6	0.517	0.440	0.085	**0.837**	0.643	0.810	**0.767**	**0.444**	0.493	**0.785**	4
T8	N0	0.201	0.270	0.747	0.666	0.315	0.520	0.633	0.389	0.216	0.744	0
T8	N1	**0.220**	0.223	0.130	0.624	**0.345**	0.342	**0.709**	0.388	**0.228**	0.714	4
T8	N2	0.179	0.225	0.180	0.632	**0.336**	0.410	**0.702**	0.388	**0.229**	0.708	3
T8	N3	0.159	0.224	0.170	0.635	**0.320**	**0.875**	**0.721**	0.388	**0.225**	0.729	4
T8	N4	**0.805**	**0.716**	0.174	**0.668**	0.306	0.091	**0.673**	0.383	**0.232**	**0.748**	**6**
T8	N5	0.152	0.224	0.170	0.634	**0.335**	0.494	**0.683**	0.387	**0.224**	0.720	3
T8	N6	0.163	0.222	0.072	0.623	**0.348**	**0.777**	**0.677**	0.386	**0.218**	0.713	4

* **FTS** stands for "Final Training Set". **Norm** stands for "Normalization". **Bold values** indicate the better accuracy than the original training set including discarded digits.

Tab. V *The accuracy of the feed-forward neural network on the final training sets T1-T4.*

FTS	**Norm**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**	**Total**
T1	N0	0.592	0.587	0.823	0.861	0.771	0.496	0.676	0.122	0.621	0.907	0
T1	N1	0.580	0.576	0.794	0.860	**0.772**	**0.600**	0.664	**0.123**	**0.647**	0.904	4
T1	N2	**0.605**	0.578	0.798	0.857	**0.776**	**0.614**	0.669	**0.124**	0.607	**0.907**	5
T1	N3	**0.596**	0.578	**0.873**	0.857	**0.772**	0.418	0.669	**0.129**	**0.624**	**0.907**	**6**
T1	N4	0.473	0.460	0.100	0.860	0.769	0.180	**0.700**	**0.128**	**0.645**	**0.907**	4
T1	N5	0.585	0.569	0.797	0.859	0.762	**0.638**	**0.681**	**0.127**	**0.633**	0.907	4
T1	N6	**0.593**	0.566	0.800	0.856	0.771	**0.506**	0.651	**0.128**	0.612	0.907	3
T2	N0	0.600	0.539	0.877	0.846	0.755	0.497	0.642	0.114	0.561	0.905	0
T2	N1	**0.621**	0.497	0.169	0.842	0.753	**0.595**	**0.644**	0.111	**0.580**	**0.906**	**5**
T2	N2	**0.616**	0.498	0.156	0.845	**0.756**	**0.598**	0.632	0.112	**0.577**	**0.906**	**5**
T2	N3	**0.619**	0.497	0.188	0.841	0.754	0.404	**0.645**	**0.115**	**0.572**	**0.906**	**5**
T2	N4	0.469	0.468	0.177	0.840	0.753	0.160	0.621	0.111	**0.582**	**0.906**	2
T2	N5	**0.620**	0.498	0.173	0.845	0.754	**0.613**	0.637	0.107	**0.576**	**0.905**	4
T2	N6	0.583	0.489	0.101	0.843	0.753	0.433	0.629	0.108	**0.571**	**0.906**	2
T3	N0	0.571	0.588	0.873	0.870	0.790	0.522	0.663	0.117	0.719	0.907	0
T3	N1	**0.582**	**0.597**	0.796	**0.872**	**0.794**	**0.529**	**0.675**	**0.120**	0.718	0.906	**7**
T3	N2	**0.583**	**0.596**	0.795	0.870	**0.790**	**0.591**	0.653	**0.126**	0.714	0.906	5
T3	N3	**0.582**	0.584	0.872	0.866	0.783	0.458	0.647	**0.117**	0.688	0.886	2
T3	N4	0.465	0.458	0.099	0.867	0.781	0.189	0.659	**0.119**	0.703	0.895	1
T3	N5	**0.580**	**0.590**	0.797	0.865	0.786	**0.549**	0.639	0.111	0.699	0.884	3
T3	N6	**0.575**	0.580	0.801	0.868	0.768	**0.568**	**0.668**	**0.123**	0.686	0.896	4
T4	N0	0.574	0.575	0.795	0.862	0.776	0.450	0.605	0.116	0.685	0.896	0
T4	N1	0.568	**0.583**	0.795	**0.872**	**0.778**	**0.605**	**0.646**	**0.118**	**0.693**	0.888	7
T4	N2	**0.609**	**0.582**	0.794	**0.868**	**0.777**	**0.629**	**0.640**	**0.124**	**0.685**	0.895	**8**
T4	N3	**0.576**	**0.585**	**0.868**	**0.866**	0.769	**0.485**	**0.629**	**0.124**	**0.704**	0.892	**8**
T4	N4	0.469	0.453	0.110	**0.870**	0.775	0.165	**0.654**	**0.118**	0.675	0.895	3
T4	N5	0.572	**0.579**	**0.800**	**0.870**	0.771	**0.600**	**0.652**	**0.117**	**0.702**	0.891	7
T4	N6	0.572	**0.580**	0.791	**0.869**	0.766	**0.539**	**0.653**	**0.119**	0.676	0.894	5
T5	N0	0.565	0.578	0.783	0.850	0.731	0.584	0.691	0.114	0.539	0.892	0
T5	N1	**0.579**	0.487	**0.801**	0.845	0.724	**0.618**	**0.708**	**0.115**	**0.565**	0.892	6
T5	N2	**0.581**	0.486	**0.789**	0.840	0.729	**0.664**	**0.696**	**0.117**	**0.573**	**0.893**	**7**
T5	N3	0.555	0.493	**0.860**	0.826	0.704	0.469	**0.708**	**0.120**	0.521	0.883	3
T5	N4	0.462	0.440	0.107	0.821	0.699	0.171	**0.756**	**0.122**	0.528	0.881	2
T5	N5	0.554	0.461	**0.792**	0.822	0.689	**0.686**	**0.702**	**0.118**	**0.544**	0.885	5
T5	N6	0.531	0.494	**0.802**	0.828	0.700	0.576	0.643	**0.120**	0.523	0.886	2
T6	N0	0.426	0.514	0.844	0.817	0.673	0.385	0.563	0.099	0.460	0.739	0
T6	N1	**0.516**	0.457	0.080	0.814	0.667	**0.556**	**0.607**	**0.103**	0.414	**0.775**	5
T6	N2	**0.486**	0.468	0.205	**0.838**	**0.678**	**0.575**	**0.589**	**0.107**	0.429	**0.796**	7
T6	N3	**0.540**	0.466	0.167	**0.843**	**0.695**	**0.391**	**0.592**	**0.105**	**0.477**	**0.790**	**8**
T6	N4	**0.464**	0.469	0.177	**0.822**	**0.686**	0.177	0.555	**0.103**	0.430	**0.797**	5
T6	N5	**0.525**	0.468	0.160	**0.832**	**0.678**	**0.596**	**0.589**	**0.109**	**0.481**	**0.808**	**8**
T6	N6	**0.490**	0.467	0.125	**0.833**	**0.711**	**0.452**	**0.602**	**0.104**	**0.485**	**0.811**	**8**
T7	N0	0.525	0.512	0.837	0.828	0.622	0.433	0.556	0.104	0.444	0.753	0
T7	N1	**0.578**	0.461	0.172	0.825	**0.634**	**0.511**	**0.595**	0.102	**0.494**	**0.756**	**6**
T7	N2	0.494	0.460	0.146	**0.840**	0.604	**0.562**	**0.606**	0.103	**0.469**	**0.762**	5
T7	N3	0.520	0.473	0.182	**0.844**	0.586	0.429	**0.637**	**0.104**	0.413	**0.784**	4
T7	N4	0.469	0.467	0.190	0.825	**0.637**	0.191	0.553	**0.105**	**0.452**	**0.777**	4
T7	N5	0.507	0.473	0.166	**0.836**	0.583	**0.580**	**0.613**	**0.105**	0.428	**0.770**	5
T7	N6	0.474	0.463	0.087	**0.837**	0.600	0.403	**0.580**	0.099	0.411	**0.784**	3
T8	N0	0.364	0.451	0.747	0.664	0.324	0.422	0.552	0.098	0.249	0.743	0
T8	N1	**0.447**	0.337	0.125	0.622	**0.354**	**0.469**	**0.598**	0.096	0.249	0.715	4
T8	N2	**0.412**	0.343	0.183	0.631	**0.346**	**0.488**	**0.614**	**0.100**	**0.271**	0.708	**6**
T8	N3	**0.385**	0.337	0.168	0.633	**0.328**	0.389	**0.624**	0.098	**0.258**	0.730	4
T8	N4	**0.469**	**0.466**	0.175	**0.668**	0.322	0.192	**0.613**	0.097	0.249	**0.749**	5
T8	N5	**0.402**	0.341	0.172	0.633	**0.345**	**0.502**	**0.584**	0.096	**0.252**	0.721	5
T8	N6	0.350	0.329	0.069	0.622	**0.354**	0.372	**0.579**	0.095	0.238	0.715	2

* **FTS** stands for "Final Training Set". **Norm** stands for "Normalization". **Bold values** indicate the better precision than the original training set including discarded digits.

Tab. VI *The precision of the feed-forward neural network on the final training sets T1-T4.*

FTS	Norm	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Total
T1	N0	0.586	0.655	0.822	0.862	0.773	0.668	0.786	0.313	0.664	0.906	0
T1	N1	**0.605**	0.631	0.797	0.861	**0.773**	**0.693**	**0.789**	**0.325**	0.649	0.904	5
T1	N2	**0.609**	0.618	0.800	0.858	**0.774**	**0.741**	**0.787**	**0.318**	0.639	**0.907**	**6**
T1	N3	**0.592**	0.620	**0.874**	0.858	0.772	**0.762**	**0.789**	0.306	0.648	**0.907**	5
T1	N4	**0.784**	**0.693**	0.122	0.861	0.770	0.081	0.784	0.306	0.654	**0.907**	3
T1	N5	**0.612**	0.614	0.796	0.859	0.763	**0.733**	**0.808**	**0.323**	0.645	**0.907**	5
T1	N6	**0.612**	0.628	0.804	0.857	0.772	**0.825**	**0.796**	**0.333**	0.642	0.906	4
T2	N0	0.532	0.521	0.877	0.847	0.754	0.595	0.716	0.293	0.579	0.905	0
T2	N1	**0.595**	0.492	0.184	0.842	0.753	**0.646**	0.716	**0.333**	**0.581**	**0.906**	5
T2	N2	**0.606**	0.499	0.140	0.846	**0.755**	**0.674**	0.707	0.287	**0.581**	**0.905**	5
T2	N3	**0.596**	0.495	0.268	0.843	0.754	**0.756**	**0.718**	**0.320**	**0.583**	**0.906**	**6**
T2	N4	**0.795**	**0.712**	0.169	0.841	**0.755**	0.088	0.699	**0.300**	**0.594**	**0.905**	**6**
T2	N5	**0.615**	0.499	0.177	0.846	**0.755**	**0.689**	0.708	**0.322**	**0.585**	**0.905**	**6**
T2	N6	**0.590**	0.496	0.080	0.845	0.753	**0.674**	**0.742**	**0.329**	**0.585**	**0.906**	**6**
T3	N0	0.708	0.654	0.874	0.871	0.798	0.800	0.815	0.334	0.718	0.907	0
T3	N1	0.686	**0.668**	0.796	**0.873**	**0.801**	0.731	0.809	**0.365**	0.718	0.906	**4**
T3	N2	0.688	**0.672**	0.796	0.870	0.794	0.781	0.790	**0.338**	0.717	0.906	2
T3	N3	0.675	0.652	0.873	0.867	0.793	0.785	0.781	**0.337**	0.688	0.880	1
T3	N4	**0.778**	**0.692**	0.120	0.868	0.787	0.057	0.794	**0.341**	0.703	0.897	3
T3	N5	0.674	0.652	0.795	0.867	0.793	0.721	0.798	**0.373**	0.699	0.889	1
T3	N6	0.707	0.647	0.799	0.868	0.770	0.763	0.787	0.290	0.687	0.900	0
T4	N0	0.718	0.655	0.803	0.863	0.788	0.750	0.814	0.333	0.684	0.899	0
T4	N1	0.658	0.647	0.794	**0.872**	0.786	0.711	0.800	**0.364**	**0.689**	0.886	3
T4	N2	0.667	0.646	0.790	**0.868**	0.785	**0.785**	0.795	0.303	**0.689**	0.897	3
T4	N3	0.667	0.649	**0.869**	**0.867**	0.783	**0.815**	0.802	**0.336**	**0.708**	0.895	**5**
T4	N4	**0.787**	**0.688**	0.138	**0.870**	0.780	0.050	0.800	0.319	0.684	0.890	3
T4	N5	0.672	0.643	0.794	**0.871**	0.777	**0.751**	0.793	**0.356**	**0.700**	0.894	4
T4	N6	0.669	0.650	0.798	**0.870**	0.780	**0.793**	**0.822**	**0.334**	0.682	0.889	4
T5	N0	0.612	0.643	0.778	0.851	0.738	0.669	0.704	0.313	0.535	0.896	0
T5	N1	0.608	0.502	**0.794**	0.846	0.732	0.668	**0.769**	0.307	0.531	0.887	2
T5	N2	0.580	0.476	**0.788**	0.841	0.737	**0.758**	**0.750**	0.312	**0.565**	**0.896**	**5**
T5	N3	0.542	0.479	**0.863**	0.830	0.709	**0.771**	**0.708**	0.283	0.522	0.889	3
T5	N4	**0.775**	**0.669**	0.110	0.822	0.710	0.035	**0.716**	0.281	0.533	0.878	3
T5	N5	0.565	0.474	**0.795**	0.823	0.699	**0.731**	**0.764**	0.276	0.534	0.891	3
T5	N6	0.531	0.472	**0.798**	0.829	0.714	**0.789**	**0.752**	**0.330**	0.517	0.883	4
T6	N0	0.467	0.496	0.847	0.822	0.691	0.480	0.689	0.358	0.468	0.771	0
T6	N1	**0.510**	0.421	0.102	0.819	**0.695**	**0.495**	0.652	0.347	0.427	**0.785**	4
T6	N2	**0.487**	0.478	0.178	**0.841**	**0.702**	**0.527**	0.641	0.274	0.447	**0.801**	5
T6	N3	**0.526**	0.476	0.142	**0.846**	**0.709**	**0.607**	0.683	0.352	0.461	**0.791**	5
T6	N4	**0.788**	**0.706**	0.169	0.817	**0.703**	0.070	0.658	0.352	0.459	**0.796**	4
T6	N5	**0.481**	**0.498**	0.178	**0.835**	**0.702**	**0.581**	0.651	0.321	**0.482**	**0.801**	**7**
T6	N6	**0.531**	0.491	0.097	**0.837**	**0.724**	**0.606**	0.648	0.332	**0.489**	**0.803**	6
T7	N0	0.536	0.502	0.844	0.832	0.679	0.517	0.702	0.328	0.459	0.766	0
T7	N1	**0.545**	0.494	0.162	0.831	0.673	**0.537**	0.648	0.287	**0.531**	0.756	3
T7	N2	0.493	0.485	0.143	**0.842**	0.658	0.484	0.643	**0.349**	**0.459**	**0.780**	**4**
T7	N3	0.504	0.447	0.195	**0.846**	0.638	**0.543**	0.674	0.310	0.444	**0.793**	3
T7	N4	**0.790**	**0.712**	0.217	0.827	0.671	0.071	0.687	0.321	0.438	**0.790**	3
T7	N5	0.496	**0.533**	0.268	**0.839**	0.648	**0.549**	0.645	0.324	0.406	**0.773**	**4**
T7	N6	0.525	0.485	0.128	**0.839**	0.664	**0.704**	0.679	**0.341**	0.409	**0.788**	**4**
T8	N0	0.306	0.425	0.762	0.672	0.405	0.389	0.559	0.262	0.245	0.758	0
T8	N1	**0.311**	0.285	0.130	0.631	**0.433**	**0.447**	**0.641**	0.252	**0.261**	0.721	5
T8	N2	0.253	0.288	0.181	0.638	**0.427**	0.377	**0.598**	0.223	**0.257**	0.720	3
T8	N3	0.232	0.288	0.137	0.641	**0.412**	**0.584**	**0.588**	0.237	**0.253**	0.742	4
T8	N4	**0.787**	**0.706**	0.146	0.671	**0.415**	0.061	**0.586**	0.229	**0.266**	**0.766**	**6**
T8	N5	0.212	0.286	0.194	0.641	**0.435**	**0.435**	**0.618**	0.241	**0.253**	0.727	4
T8	N6	0.256	0.294	0.032	0.633	**0.439**	**0.661**	**0.594**	0.238	0.245	0.738	3

* **FTS** stands for "Final Training Set". **Norm** stands for "Normalization". **Bold values** indicate the better recall than the original training set including discarded digits.

Tab. VII *The recall of the feed-forward neural network on the final training sets T1-T4.*

FTS	Norm	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Total
T1	N0	0.588	0.626	0.822	0.861	0.772	0.680	0.810	0.235	0.609	0.907	0
T1	N1	0.582	0.600	0.797	0.860	**0.772**	**0.772**	**0.813**	**0.237**	**0.633**	0.905	5
T1	N2	**0.599**	0.587	0.800	0.856	**0.775**	**0.785**	**0.812**	**0.236**	0.604	**0.907**	**6**
T1	N3	0.585	0.583	**0.873**	0.857	0.771	**0.777**	0.805	**0.237**	**0.615**	**0.907**	5
T1	N4	**0.829**	**0.764**	0.114	0.860	0.769	0.081	0.794	0.231	**0.631**	**0.907**	4
T1	N5	**0.590**	0.586	0.801	0.858	0.767	**0.796**	**0.817**	**0.242**	**0.625**	**0.907**	**6**
T1	N6	**0.597**	0.599	0.805	0.855	0.771	**0.768**	0.807	0.231	0.604	0.907	2
T2	N0	0.516	0.464	0.877	0.845	0.754	0.659	0.739	0.224	0.517	0.905	0
T2	N1	**0.596**	**0.466**	0.178	0.844	0.752	**0.747**	**0.741**	**0.258**	**0.539**	**0.906**	7
T2	N2	**0.604**	**0.468**	0.163	0.844	**0.754**	**0.755**	**0.741**	**0.230**	**0.536**	**0.906**	**8**
T2	N3	**0.595**	0.453	0.224	0.841	0.753	**0.832**	**0.745**	**0.245**	**0.529**	**0.906**	6
T2	N4	**0.832**	**0.783**	0.151	0.839	0.753	0.078	0.721	**0.224**	**0.528**	**0.906**	5
T2	N5	**0.608**	0.449	0.204	0.844	0.753	**0.760**	0.732	**0.240**	**0.522**	**0.905**	5
T2	N6	**0.599**	**0.477**	0.123	0.843	0.752	**0.663**	**0.758**	**0.253**	**0.522**	**0.906**	7
T3	N0	0.603	0.606	0.873	0.870	0.793	0.724	0.807	0.270	0.718	0.907	0
T3	N1	**0.611**	**0.618**	0.803	**0.872**	**0.796**	**0.732**	0.795	**0.283**	0.718	0.906	**6**
T3	N2	**0.619**	**0.617**	0.873	**0.870**	0.792	**0.729**	**0.821**	0.239	0.718	0.906	5
T3	N3	0.600	0.596	0.872	0.865	0.786	**0.775**	0.806	**0.277**	0.688	0.886	2
T3	N4	**0.822**	**0.763**	0.106	0.866	0.782	0.071	0.799	0.248	0.701	0.899	2
T3	N5	0.570	0.602	0.797	0.864	0.788	0.702	0.800	0.252	0.696	0.893	0
T3	N6	**0.643**	0.581	0.807	0.867	0.772	**0.750**	0.801	0.248	0.686	0.901	2
T4	N0	0.643	0.626	0.794	0.861	0.778	0.648	0.799	0.335	0.683	0.900	0
T4	N1	0.574	0.597	0.794	**0.872**	**0.779**	**0.681**	**0.814**	0.265	**0.689**	0.898	**5**
T4	N2	0.617	0.589	0.791	**0.867**	**0.779**	**0.725**	**0.804**	0.241	0.682	0.899	4
T4	N3	0.580	0.599	**0.868**	**0.866**	0.769	**0.762**	0.799	0.235	**0.706**	0.896	4
T4	N4	**0.830**	**0.750**	0.098	**0.870**	**0.781**	0.054	0.787	0.236	0.679	0.899	4
T4	N5	0.598	0.595	**0.803**	**0.870**	0.767	**0.694**	0.786	0.230	**0.700**	0.894	4
T4	N6	0.587	0.591	0.791	**0.869**	0.774	**0.723**	**0.804**	0.226	0.675	0.898	3
T5	N0	0.603	0.626	0.781	0.850	0.728	0.639	0.753	0.226	0.503	0.897	0
T5	N1	0.575	0.474	**0.801**	0.844	0.722	**0.789**	**0.795**	**0.244**	**0.527**	0.896	5
T5	N2	0.555	0.430	**0.788**	0.839	0.726	**0.761**	**0.792**	**0.233**	**0.539**	**0.898**	**6**
T5	N3	0.523	0.437	**0.860**	0.825	0.703	**0.695**	**0.782**	0.223	0.491	0.884	3
T5	N4	**0.809**	**0.737**	0.115	0.820	0.696	0.022	**0.760**	0.205	0.498	0.888	3
T5	N5	0.526	0.475	**0.795**	0.821	0.682	**0.765**	**0.786**	0.206	**0.510**	0.879	4
T5	N6	0.496	0.454	**0.808**	0.827	0.698	**0.754**	**0.796**	0.219	**0.504**	0.887	4
T6	N0	0.332	0.476	0.844	0.816	0.675	0.571	0.681	0.245	0.424	0.762	0
T6	N1	**0.459**	0.379	0.089	**0.818**	0.667	**0.574**	0.666	0.230	0.351	**0.826**	4
T6	N2	**0.404**	0.406	0.201	**0.838**	0.671	**0.622**	0.657	0.219	0.398	**0.834**	4
T6	N3	**0.476**	0.414	0.176	**0.843**	**0.692**	**0.723**	**0.703**	0.242	0.422	**0.826**	**6**
T6	N4	**0.826**	**0.782**	0.151	**0.826**	**0.679**	0.084	0.679	**0.265**	0.414	**0.835**	**6**
T6	N5	**0.408**	0.467	0.177	**0.833**	0.666	**0.634**	0.663	0.225	**0.435**	**0.850**	5
T6	N6	**0.481**	0.455	0.132	**0.838**	**0.709**	**0.609**	0.650	0.236	**0.453**	**0.851**	**6**
T7	N0	0.456	0.460	0.838	0.827	0.604	0.627	0.705	0.204	0.434	0.785	0
T7	N1	**0.523**	0.442	0.189	**0.829**	**0.640**	0.532	0.665	**0.240**	**0.513**	**0.796**	**6**
T7	N2	0.394	0.329	0.159	**0.840**	0.588	0.511	0.671	**0.244**	**0.453**	**0.788**	4
T7	N3	0.402	0.388	0.188	**0.844**	0.562	**0.702**	0.678	**0.237**	0.431	**0.817**	4
T7	N4	**0.831**	**0.781**	0.191	0.824	**0.634**	0.084	0.684	**0.224**	**0.450**	**0.812**	**6**
T7	N5	0.423	**0.507**	0.167	**0.835**	0.559	0.581	0.661	**0.229**	0.428	**0.808**	4
T7	N6	**0.459**	0.416	0.074	**0.837**	0.580	**0.708**	0.677	**0.237**	0.405	**0.820**	5
T8	N0	0.175	0.246	0.747	0.660	0.292	0.361	0.570	0.157	0.213	0.768	0
T8	N1	**0.183**	0.205	0.132	0.618	**0.328**	0.326	**0.637**	**0.177**	0.212	0.725	4
T8	N2	0.156	0.208	0.197	0.629	**0.314**	0.335	**0.628**	0.154	**0.225**	0.712	3
T8	N3	0.138	0.205	0.192	0.631	**0.302**	**0.689**	**0.645**	**0.162**	**0.222**	0.743	5
T8	N4	**0.830**	**0.780**	0.153	**0.664**	0.283	0.074	**0.596**	**0.166**	0.212	**0.770**	**6**
T8	N5	0.132	0.206	0.163	0.633	**0.317**	**0.412**	**0.617**	**0.159**	**0.219**	0.743	5
T8	N6	0.164	0.226	0.078	0.622	**0.330**	**0.641**	**0.609**	0.156	0.203	0.748	3

* **FTS** stands for "Final Training Set". **Norm** stands for "Normalization". **Bold values** indicate the better f1-score than the original training set including discarded digits.

Tab. VIII *The F1-score of the feed-forward neural network on the final training sets T1-T4.*

Performance Measure	Normalization	T1 ORG+MCOV	T2 MCOV	T3 ORG+FF-AA MCOV	T4 ORG+FA-AF MCOV	T5 ORG+FAF-AFA MCOV	T6 FF-AA MCOV	T7 FA-AF MCOV	T8 FAF-AFA MCOV	Total
Accuracy	N1 MIN-MAX	**6**	5	**5**	4	5	4	**6**	4	**39**
	N2 Z-SCORE	**6**	**7**	3	4	**6**	6	4	3	**39**
	N3 MEAN MAD	5	**7**	1	**5**	3	**8**	4	4	37
	N4 MEDIAN MedAD	4	5	1	2	1	7	4	**6**	30
	N5 TANH (MODIFIED)	4	4	0	**5**	3	6	5	3	30
	N6 TANH HAMPEL	3	6	2	4	3	7	4	4	33
	Total	28	34	12	24	21	**38**	27	24	
Precision	N1 MIN-MAX	4	**5**	**7**	7	6	5	**6**	4	44
	N2 Z-SCORE	5	**5**	5	**8**	**7**	7	5	**6**	**48**
	N3 MEAN MAD	**6**	**5**	2	**8**	3	**8**	4	4	40
	N4 MEDIAN MedAD	4	2	1	3	2	5	4	5	26
	N5 TANH (MODIFIED)	4	4	3	7	5	**8**	5	5	41
	N6 TANH HAMPEL	3	2	4	5	2	**8**	3	2	29
	Total	26	23	22	38	25	**41**	27	26	
Recall	N1 MIN-MAX	5	5	**4**	3	2	4	3	5	31
	N2 Z-SCORE	**6**	5	2	3	**5**	5	**4**	3	33
	N3 MEAN MAD	5	**6**	1	**5**	3	5	3	4	32
	N4 MEDIAN MedAD	3	**6**	3	3	3	4	3	**6**	31
	N5 TANH (MODIFIED)	5	**6**	1	4	3	**7**	**4**	4	**34**
	N6 TANH HAMPEL	4	**6**	0	4	4	6	**4**	3	31
	Total	28	**34**	11	22	20	31	21	25	
F1-score	N1 MIN-MAX	5	7	**6**	**5**	5	4	**6**	4	**42**
	N2 Z-SCORE	**6**	**8**	5	4	**6**	4	4	3	40
	N3 MEAN MAD	5	6	2	4	3	**6**	4	5	35
	N4 MEDIAN MedAD	4	5	2	4	3	**6**	**6**	**6**	36
	N5 TANH (MODIFIED)	**6**	5	0	4	4	5	4	5	33
	N6 TANH HAMPEL	2	7	2	3	4	**6**	5	3	32
	Total	28	**38**	17	24	25	31	29	26	

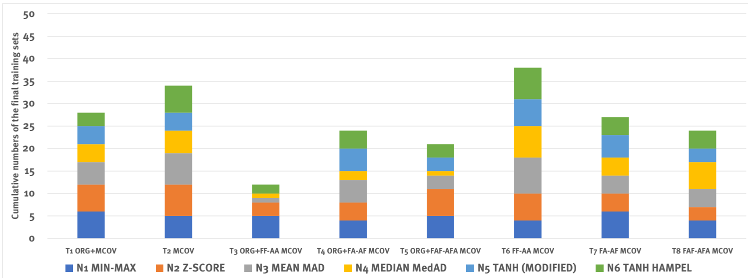
* **Bold values** indicate the number of datasets of each experimental setup that yielded the highest number of accuracy, precision, recall, or F1-score.

Tab. IX *The number of data sets that a specified normalization technique (N1-N6) improved the performance measure of the related final training sets (T1-T8).*

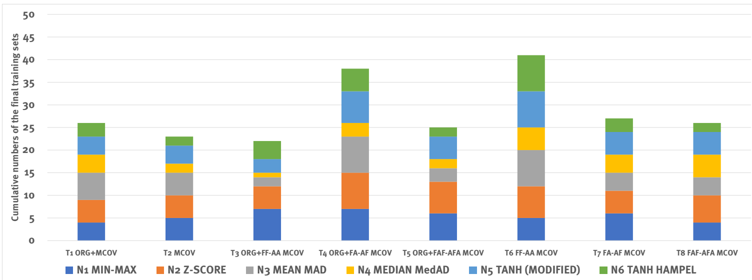
tangent normalization can be a default function when a recall is prioritized, i.e., selecting relevant items. In addition, the T6 final training set (the final training set having only the FF-AA reduced boundary vectors) can be a default training set type when accuracy and precision are prioritized. The T2 final training set (the final training set having only the boundary vectors) can be a default training set type when recall and F1-score are prioritized.

Alternatively, one of many ways to help decide which type of the final training set should be generally selected first is to consider the cumulative numbers of the datasets that the implementation of each of the six normalization techniques yielded higher accuracy, precision, recall, or F1-score than the other normalization techniques, which is summarized in Figs. 6 and 7. Regarding the boundary vector generator (T1, T2), the T2 MCOV training set gained the highest accuracy, recall, and F1-score improvement. The T1 ORG+MCOV training set gained the highest precision improvement. Regarding the reduced boundary vector generators (T3, T4, T5, T6, T7, T8), the T6 FF-AA MCOV training set gained the highest accuracy, precision, recall, and F1-score improvement. These training set types should be generally selected first.

Besides, another way to help decide which normalization technique should be generally selected first is to consider the cumulative numbers of the datasets that each normalization technique yielded higher accuracy, precision, recall, and F1-score than the other techniques, which is summarized in Figs. 8 and 9. It is noticeable that the min-max (N1) normalization and Z-score (N2) normalization techniques generally improved the accuracy of all types of training sets. The Z-score (N2) normalization technique generally improved the precision of all types of training sets. The modified hyperbolic tangent (N5) normalization technique generally improved the recall of all types of training sets. The min-max (N1) normalization technique generally improved the F1-score of all types of training sets.

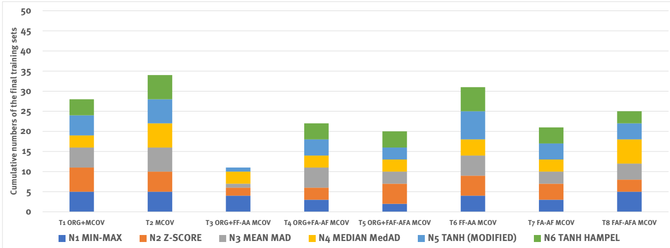


(a)

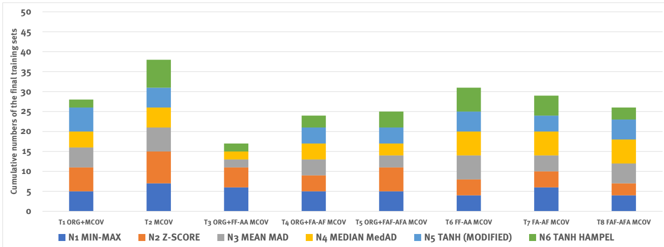


(b)

Fig. 6 *The cumulative numbers of the final training sets that the implementation of any normalization techniques yielded higher (a) accuracy and (b) precision than without the implementation of any normalization techniques.*

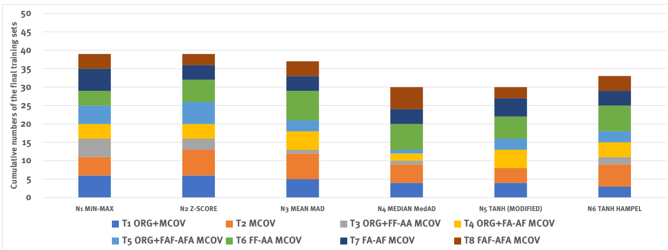


(a)

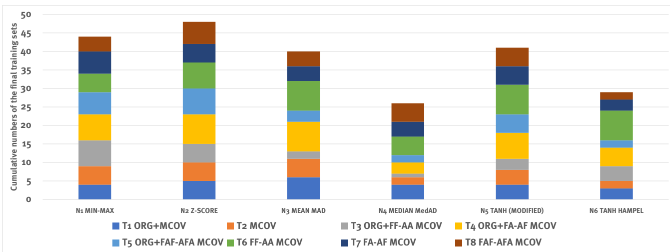


(b)

Fig. 7 *The cumulative numbers of the final training sets that the implementation of any normalization techniques yielded higher (a) recall and (b) F1-score than without the implementation of any normalization techniques.*

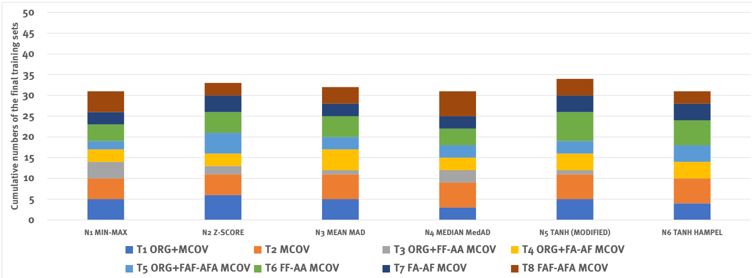


(a)

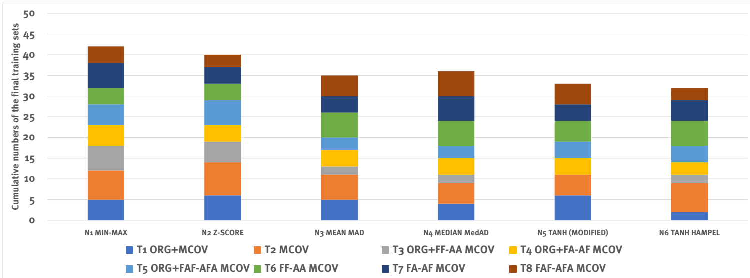


(b)

Fig. 8 *The cumulative numbers of the final training sets that each normalization technique yielded higher (a) accuracy and (b) precision than the same type of the final training set without normalization.*



(a)



(b)

Fig. 9 *The cumulative numbers of the final training sets that each normalization technique yielded higher (a) recall and (b) F1-score than the same type of the final training set without normalization.*

5. Conclusions

The feed-forward neural network is a prediction model widely used in a significant number of real-world implementations that involve pattern identification and classification. Factors influencing the accuracy of this class prediction model include the training parameter adjustment and the quality of the training set. The boundary vector generator is a data barrier amplifying technique that improves the distribution model of the samples to increase the classification performance of the feed-forward neural network. This method generates new forms of boundary vectors; one is named fundamental multi-class outpost vector, the other is named additional multi-class outpost vector. They are used to amplify the strength of the class barrier. However, these sets of boundary vectors are enormous. The reduced boundary vector generators present three techniques to reduce the number of fundamental multi-class outpost vectors and additional multi-class outpost vectors. Nevertheless, these techniques do not consider the interval of the attributes, causing some attributes to suppress the other attributes on the Euclidean distance calculation. This paper studies whether six normalization techniques; min-max, Z-score, mean and mean absolute deviation, median and median absolute deviation, modified hyperbolic tangent, and hyperbolic tangent estimator, can improve the classification performance on four primary performance measures of the boundary vector generator and the reduced boundary vector generators for maximizing class boundary; accuracy, precision, recall, and F1-score. Each normalization technique pre-processes the original training set before the boundary vector generator, or each of the three reduced boundary vector generators will begin. The experimental results on the real-world datasets from the UCI database and the ELENA project confirmed that

(1) the final training set having only FF-AA reduced boundary vectors (T6) can be integrated with one of the normalization techniques effectively when the accuracy and precision are prioritized,

(2) the final training set having only the boundary vectors (T2) can be integrated with one of the normalization techniques effectively when the recall and F1-score are prioritized,

(3) the Z-score normalization can generally improve the accuracy and precision of all types of training sets,

(4) the modified hyperbolic tangent normalization can generally improve the recall of all types of training sets,

(5) the min-max normalization can generally improve the accuracy and F1-score of all types of training sets, and

(6) the selection of the normalization techniques and the training set types depends on the key performance measure for the dataset.

List of abbreviations

- AA (additional-to-additional outpost vector)
- AF (additional-to-fundamental outpost vector)
- AFA (additional-to-fundamental/additional outpost vector)
- COV (contour preserving classification)
- FA (fundamental-to-additional outpost vector)
- FAF (fundamental-to-additional/fundamental outpost vector)
- FF (fundamental-to-fundamental outpost vector)
- FF-AA RMCOV (FF-AA reduced boundary vector)
- FF-AA RMCOV generator (FF-AA reduced boundary vector generator)
- FA-AF RMCOV (FA-AF reduced boundary vector)
- FA-AF RMCOV generator (FA-AF reduced boundary vector generator)
- FAF-AFA RMCOV (FAF-AFA reduced boundary vector)
- FAF-AFA RMCOV generator (FAF-AFA reduced boundary vector generator)

- FFNN (Feed-Forward Neural Network)
- MCOV (multi-class outpost vector / boundary vector)
- MCOV generator (boundary vector generator)
- RMCOV (reduced boundary vector)
- RMCOV generator (reduced boundary vector generator)
- MAD (mean absolute deviation)
- MedAD (median absolute deviation)

References

[1] COVER T., HART P. Nearest neighbor pattern classification. *IEEE Trans Inf Theory*, 1967, 13(1), pp. 21–27, doi: 10.1109/TIT.1967.1053964.

[2] RUSSEL S., NORVING P. *Artificial Intelligence A Modern Approach (2nd ed.)*. Pearson Education, Delhi, India, 2004.

[3] CORTES C., VAPNIK V. Support-vector networks. *Machine Learning*, 1995, 20(3), pp. 273–297, doi: 10.1007/BF00994018.

[4] TANPRASERT T., TANPRASERT C., LURSINSAP C. Contour preserving classification for maximal reliability. In: *Proceedings of the 1998 International Joint Conference on Neural Networks*, Anchorage, Alaska, 1998, doi: 10.1109/IJCNN.1998.685930.

[5] MONGKONSIRIVATANA J. Neural network reliability enhancement with boundary detection contour preserving training. In: *Proceedings of the International Conference on Intelligent Technology*, Bangkok, Thailand, 2005, pp. 234–239.

[6] TANPRASERT T., FUANGKHON P., TANPRASERT C. An improved technique for retraining neural networks in adaptive environment. In: *Proceedings of the International Conference on Intelligent Technology*, Samui, Thailand, 2008, pp. 77–80.

[7] TANPRASERT T., KRIPRUKSAWAN T . An approach to control aging rate of neural networks under adaptation to gradually changing context. In: *Proceedings of the 9th International Conference on Neural Information Processing*, Singapore, 2002, pp. 174–178.

[8] KAITIKUNKAJORN S., TANPRASERT T. Improving synthesis process of decayed prior sampling technique. In: *Proceedings of the 6th International Conference on Intelligent Technology*, Samut Prakan, Thailand, 2005, pp. 240–244.

[9] FUANGKHON P., TANPRASERT T. Multi-class Contour Preserving Classification. In: *Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2012)*, Natal, Brazil, 2012, pp. 35–42, doi: 10.1007/978-3-642-32639-4_5.

[10] FUANGKHON P., TANPRASERT T. Reduced multi-class contour preserving classification. *Neural Process Lett*, 2016, 43(3), pp. 759–804, doi: 10.1007/s11063-015-9446-1.

[11] HAN J., KAMBER M., PEI J. Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann, Waktham, MA, 2012.

[12] LATHA L., THANGASAMY S. Efficient approach to normalization of multimodal biometric scores. *Int J of Comp Appl*, 2011, 32(10), pp. 57–64, doi: 10.5120/3952-5530.

[13] ISO 13528:2015. Statistical methods for use in proficiency testing by interlaboratory comparison. 2015 [viewed 2017-05-01]. Available from: https://www.iso.org/standard/56125.html.

[14] JAIN A., NANDAKUMAR K., ROSS A. Score normalization in multimodal biometric systems. *Pattern Recognit*, 2005, 38(12), pp. 2270–2285, doi: 10.1016/j.patcog.2005.01.012.

[15] UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. 2017 [viewed 2017-05-01]. Available from: https://archive.ics.uci.edu/ml.

[16] ELENA Project. Louvain-la-Neuve, Belgium: Université catholique de Louvain, Laboratoire de Microélectronique. 2017 [viewed 2017-05-01]. Available from: https://www.elen.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm.

[17] SOKOLOVAA M., LAPALME G. A systematic analysis of performance measures for classification tasks. *Inf Process Manag*, 2009, 45(4), pp. 427–437, doi: 10.1016/j.ipm.2009.03.002.