Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 3987 – 3998

doi:10.3906/elk-1411-36

Time series adapted supervised fuzzy discretization: an application to ECG signals

Umut ORHAN*
Department of Computer Engineering, Faculty of Engineering and Architecture, Çukurova University, Adana, Turkey

Received: 06.11.2014 • **Accepted/Published Online:** 28.06.2015 • **Final Version:** 20.06.2016

Abstract: In this study, a new method called supervised fuzzy discretization (SFD), which can be used without having expertise on data, is proposed for classifying time series datasets. Because an ECG signal has a partially stationary characteristic, its classification process is more difficult than it would be for completely stationary signals. On the other hand, because the method proposed can be used without having expertise on the data, comprehensive data like ECG signals are enough to introduce one such method. To prove the efficacy of the SFD, RR intervals selected from a common ECG database are used in the classification experiments. Some parameters, such as the coefficients of discretization, equal time slicing, learning rate, and momentum, are analyzed for the highest level of success in classification. A new mechanism called an inconsistency detector is suggested for increasing the level of success in supervised learning by adjusting the learning rate. The best results of the SFD method are compared with those of other studies in the same database, which hopefully establishes the proposed method as worth investigating in other areas because of its projected success.

Key words: Supervised fuzzy discretization, inconsistency detector, time series, electrocardiograph, congestive heart failure

1. Introduction

Time series are datasets with continuous and quantitative characteristics that are encountered in real-world applications in areas like medicine, biology, and economics [1–8]. These continuous attributes are recorded as discrete signals by means of digital systems. In the last decade particularly, many studies have focused on the discretization of already discrete signals for the discovery of knowledge [9–15]. There are two parameters that affect results in discretization: the number of cut points and their locations. Although there are several unsupervised algorithms in the literature focusing on the adjustment of these parameters [16–23], there is not an absolute answer for the question of which one is the best algorithm, because each one of those algorithms can attain the best result for different problems. However, this claim is not correct for a supervised technique, since it tries to adapt itself to input data.

Discretization methods are classified by various taxonomies according to point of view: supervised/unsupervised, crisp/fuzzy, static/dynamic, local/global, top-down/bottom-up, and direct/indirect [10,23]. The most frequently used methods, in contrast to the one used in this study, belong to the unsupervised and crisp groups, and the best known methods are equal width and equal frequency discretization methods [9,24,25]. Although many researchers have conducted studies in this area, there is no single method in the literature used for the

*Correspondence: uorhan@cu.edu.tr

classification of a time series. Before the classification procedure, a time series, especially in an automated system, is put through a number of steps, such as denoising, signal improvement-enhancement, and feature extraction. The discretization implementations applied to time series are intended to enhance the input signal. The effects of discretization methods on the improvement of time series are analyzed in terms of entropy in some studies [17,26]. Because the discretization procedure raises the possibility of repetition in the values of the time series, as a natural consequence, it reduces the entropy value of the signal. It increases intraclass similarity and thus the possibility of obtaining high classification success.

The frequencies of the amplitude values in time series are used as the features of the input signal in some studies [27,28]. In those studies aiming to extract the most meaningful features of a signal, a probability term is used to represent the likelihood of having an amplitude value in any discrete interval. On the other hand, the probability density values can also be considered as fuzzy memberships, indicating how much of a signal belongs to any discrete interval.

When the powerful and meaningful features of a time series are extracted, there are some studies showing that very powerful classifiers are not needed for classification [29]. However, the supervised discretization methods known in the literature [17,30–40] are not suitable for the classification of time series without using any classifier method together. Thus, this study aims to put forward a time series adapted neural network method (in other words, a new supervised discretization classifier) that does not need an extra classifier and adjusts discretization steps by using class labels of the input signals.

Although to say a proposed method may be applicable to every time series is possible in theory, to claim its efficacy requires proof obtained through serious experimentation in several fields. In this study, the method recommended for classification of time series is introduced in the use of ECG signals to detect a heart disease, congestive heart failure (CHF). CHF is often determined through waveform analysis on ECG segments. However, there is also a study with a time-slicing step that proposes the recognition of some special RR intervals in ECG signals in order to detect the disease [29]. Therefore, the signals used in this study were subjected to the preliminary step of time slicing.

In this study, RR intervals for three datasets (training, validation, and test) were prepared by random trimming from long-term ECG signals in the Physiobank archive [41], then aligned and normalized into the [0 1] region. The results of experiments are examined in order to determine the effects of some important parameters on classification success. Since there is no equivalent method in the literature, the results of the proposed method were compared with the results of the studies that previously classified the same signals. The rest of the study is organized as follows: the used dataset, the supervised fuzzy discretization (SFD) method, the results of the experiments performed, and lastly the conclusion. To understand the proposed approach, its general architecture is shown in Figure 1.

2. ECG dataset

This study used the Normal Sinus Rhythm (NSR) and the Congestive Heart Failure (CHF) databases from Physiobank [41], which were downloaded from its website. The main reason to use ECGs in the experiments is that the signal represents more general areas on account of its partial stationary characteristic. Long-term ECG recordings in the databases are composed of 15 in CHF at 250 Hz and 18 in NSR at 128 Hz. For preparing the dataset of RR intervals, segments with a length of 1 s were randomly chosen from each long-term ECG record. All of them were aligned to the maximum amplitude value (R peak) in the middle of the segment. According to the health situation (NSR or CHF) of the subject, the RR intervals chosen from the subject's record were classified without using the class information of the RR interval labeled by experts. In order to provide equal opportunity in comparison, all signals were resampled into 250 Hz and also normalized into [0 1] at amplitude axis. Thus, each segment was made independent from maximum and minimum values. Two different RR intervals selected from the dataset prepared by cutting from ECG segments are shown below.

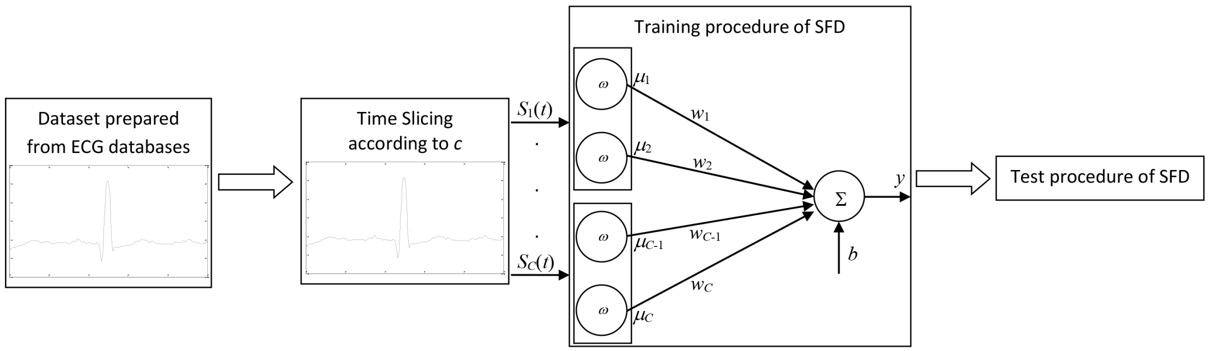

Figure 1. Architecture of the proposed approach.

In Figure 2, the difference between two RR intervals can be visually discerned without any prior knowledge, but physicians usually analyze ECG signals via their (P, Q, R, S, T) waves. Therefore, some parameters of these waves, such as the duration and the amplitude values, are important in decision making. Because a method independent from expertise on data is desired, more general features should be used, like equal time slicing [29]. In this study, the RR intervals were divided into some parts to simulate a classical waveform analysis, but to avoid determining the duration of each slice, the time axis was divided into slices with the same duration. Thus, each slice can be interpreted as independent from the waves of (P, Q, R, S, T). Then the segment in each slice was entered into the proposed method. In other words, equal time slicing is not a part of the SFD method. Instead, a time-frequency analysis could also be utilized.

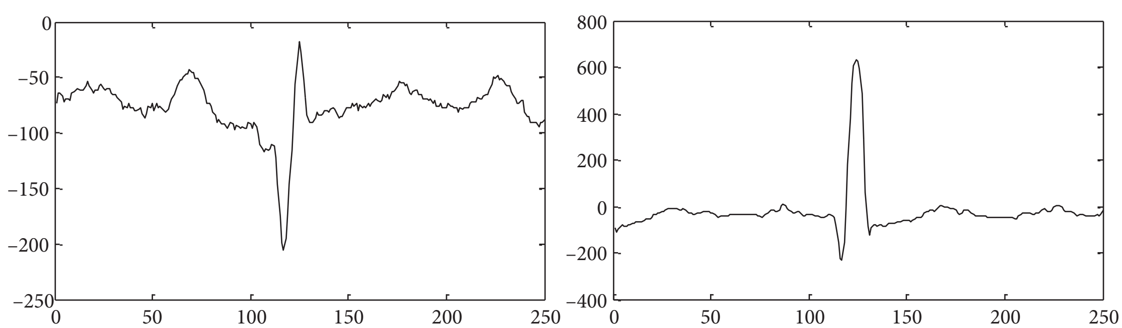

Figure 2. RR intervals selected from the datasets: (a) CHF, (b) NSR.

3. SFD method

The classification of the time series was mostly performed in two phases: feature extraction and classification. For both phases, there are many different possible methods. According to the selected problem area, the researcher should choose the most suitable method from the several that are available. In this study, a time series adapted neural network model, which can be used without having expertise on data, is proposed by combining a feature extraction method [27,28] and a conventional single layer perceptron classifier. This combination is a new entity like a chemical compound. Because each component (feature extraction and classifier) updates its own parameters by using knowledge from each other in training, as compounds they behave differently.

Fuzzy discretization is a method that converts a continuous signal into a discrete one, and it can represent a signal with some membership values. It can be implemented on discrete signals for acquiring certain types of information, but if it is used on a signal classification problem, it will be meaningless because of unsupervised discretization. Thus, it must be supervised during renovation. For this matter, the perceptron model is utilized, and discretization intervals are determined by using updated information from backpropagation. The proposed network has a simple architecture with two layered neurons, as shown in Figure 3.

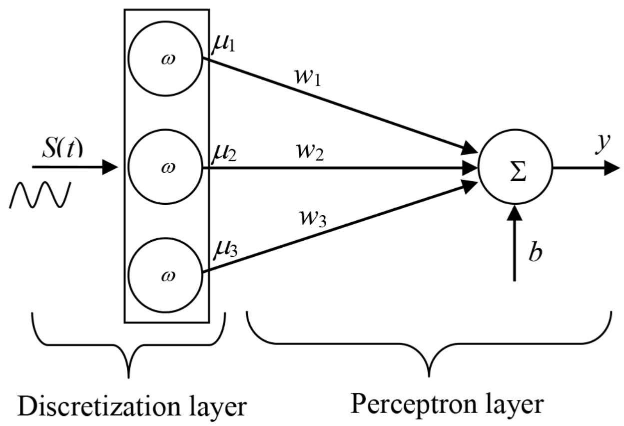

Figure 3. Symbolic representation of SFD with one output and three neurons in the discretization layer.

In Figure 3, $S(t)$ is the input signal and y is the estimation (or class) value. The discretization layer determines fuzzy memberships (μ_i) from the input signal by discretizing and then these values are sent to the perceptron layer. The perceptron layer behaves similarly to a classical perceptron model. The neuron in the perceptron layer sums the incoming weighed inputs ($\mu_i \times w_i$) and sends the summation into its activation function. Three activation functions were used in this study: the step, the sigmoid, and the hyperbolic-tangent. Each neuron in the discretization layer includes an interval width value (ω_i), and the discretization layer produces some membership values according to the position of samples in the input signal and the interval width values in neurons. In fact, the number of the neurons in the discretization layer also represents the number of discretization coefficients (k). After the discretization, the amplitude axis is divided into k discrete intervals. The membership values are calculated by:

$$\mu_i = \frac{|S \cap D_i|}{|S|} i = 1, 2, \ldots k \tag{1}$$

where S and D_i represent the input signal and discrete intervals, respectively. $|S \cap D_i|$ is the number of samples of the S signal in the D_i interval, and $|S|$ is the number of samples of the S signal. For a signal divided into k intervals by SFD, the discrete intervals (D_i) and interval width values (ω_i) are shown in Figure 4.

Producing an output value by the network is called feedforward. In a feedforward operation, the output value of the network is calculated by:

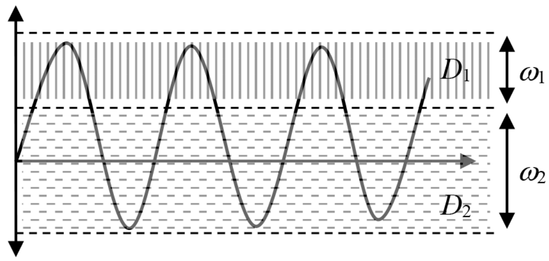

Figure 4. A discretization representation of a signal into two discrete intervals by SFD.

$$y=f\left(\sum_{i}\mu_i w_i+b\right) \tag{2}$$

where μ_i is the ith feature of the input signal, w_i is the weight of the ith feature, b is the bias, and f is an activation function in order to provide nonlinear solutions. The delta rule was preferred for this study, as it is the most frequently used training algorithm. In delta training, the procedure depends on the level of error in the output. According to the output error, the weights are updated by:

$$\Delta w_i(t)=\alpha\Delta w_i(t-1)-\eta\frac{\partial E(t)}{\partial w_i(t)} \tag{3}$$

where $E(t)$ is the energy of the output error, $\Delta w_i(t)$ is the updated amount of the weight, $\Delta w_i(t-1)$ is the updated amount of the weight in the previous iteration, α is the momentum coefficient, and η is the learning rate. Taking inspiration from the delta rule, the training phase of the SFD method, which is special to this study, is performed by the equation below.

$$\Delta\mu_i(t)=\alpha\Delta\mu_i(t-1)-\eta\frac{\partial E(t)}{\partial\mu_i(t)} \tag{4}$$

Because the essential aim is to update interval width values, the relations between ω_i and μ_i should be analyzed correctly. In the proposed method, while determining μ_i values, each amplitude value of the S signal is considered as an element of the interval in which it comes. Since a wide interval includes more elements (samples of the signal), interval width values and the membership values produced by the help of these intervals are in direct proportion. Also according to Eq. (1), the relationship between ω_i (width value of interval D_i) and μ_i is seen as a direct proportion again. Therefore, the updating amount in each μ_i shows the required updating amount in each ω_i value, and thus Eq. (4) is transformed as below.

$$\Delta\omega_i(t)=\alpha\Delta\omega_i(t-1)-\eta\frac{\partial E(t)}{\partial\mu_i(t)} \tag{5}$$

Then, since the sum of membership values must be one, the interval-width values (ω_i) are normalized after each iteration performed.

During the training procedure, the success ratio is expected to increase consistently. In some observations of experiments, however, oscillations in the success ratio were monitored, and it was deduced that this situation did not allow for the optimum parameters for the best chance of success. To prevent the oscillations, a mechanism called an "inconsistency estimator" was constructed. The mechanism analyzes two different values (difference of long–short time averages and difference of short–long time variances) in success on the validation set.

- Mean(L–S): Difference of long–short time averages
- Var(S–L): Difference of short–long time variances

If the values of both Mean(L–S) and Var(S–L) are equal, the mechanism gives an inconsistency alarm. When the inconsistency alarm is given, the gradient descent coefficient (η) is multiplied by 0.5. Each time, the situation of the existing inconsistency is tested, and the training continues. When the inconsistency is still maintained in sequential fifth controls, the training is stopped. The fifth check represents a long period for inconsistency. Both training and validation sets are analyzed for this control: the training set for updating weights, and the validation set for adjusting parameters. Thus, the training is maintained for a longer time in consistency, and the optimization of parameters presenting the best chance of success is guaranteed. To understand the proposed model easily, its algorithm steps are described below.

1. Normalization: Signal datasets (training, validation, and test) are prepared by normalizing all of them in [0 1] and separating them into subsegments by equal time slicing coefficient (c). According to the entered discretization coefficient (k), all interval-width values (ω_i) are set to $1/k$ initial value ($i = 1, 2, , k$).

2. Discretization: Each segment in the training set is discretized by fuzzy discretization with ω_i and μ_i values, which are calculated by Eq. (1).

3. Feedforward: By using Eq. (2), the network output (y) is computed for each segment in the training set. The network error is then determined, and also classification success is computed on the validation set.

4. Update: According to Eq. (3), the weights in the network are updated, and the required update amounts in interval widths are determined by using Eq. (5). After updating the widths, all widths are normalized so that the sum of all widths equals 1.

5. Inconsistency test: According to classification success computed on the validation set, inconsistency analysis begins. Until the inconsistency estimator gives a fifth sequential alarm or the iteration number reaches the maximum value, go to step 2; otherwise, stop the algorithm and compute the success of the method on the test set.

In addition to its algorithm, training in the SFD approach is shown in Figure 5.

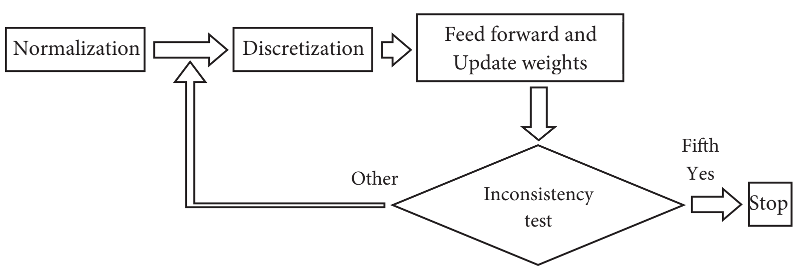

Figure 5. Training procedure of SFD approach.

The optimum value of the learning rate (η) supplies a stable increase in the classification success ratio as well as a decrease in entropy level. To measure the quality of discretization, an entropy analysis of the signal is necessary. The entropy value of the signal s is calculated by the Shannon equation below:

$$E(s) = -\sum_{i} P(s_i).log_2 P(s_i) \tag{6}$$

where s_i is the value of the ith amplitude in signal s and $P(s_i)$ is the existing frequency of the value s_i in a signal.

According to the characteristics of the studied signal, the signal may also be subjected to time slicing or time-frequency decomposition. The proposed method can be applied to every new signal obtained as described above. As a result of fragmentation (time slicing or decomposition), the structure of the SFD network applied to a signal is divided into two segments as S_1 and S_2 as shown below.

As shown in Figure 6, the architecture of the proposed method is applied as one discretization layer for each segment in the case of maintaining only one perceptron layer. In applications performed using this architecture, the memberships are normalized in the same discretization layer. Therefore, the summation of memberships in the same layer should equal 1. In this study, a novel method concerning the disintegration of the input signal by time slicing or time-frequency decomposition into subsegments was not recommended, and RR intervals in training, validation, and test sets were divided into subsegments by equal-width time slicing.

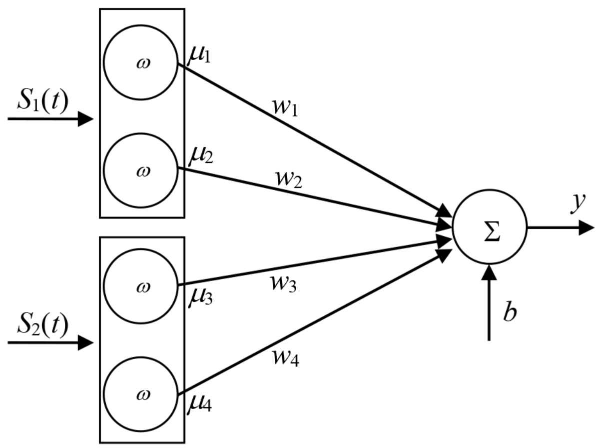

Figure 6. SFD network with one output and two discretization layers for two segments.

4. Experimental results

Because of their partially stationary characteristic, ECG signals were preferred for the classification experiments in the study. Thus, the results obtained from the ECG can also be generalized for stationary signals. The ECG database used in the experiments is a benchmark database with free access on the Internet, and it was utilized in many studies [29,42–44]. In the preparation of the datasets, 20 segments with lengths of 1 s were cut from random locations in each long-term signal and then aligned. According to random subsampling cross-validation, the big dataset with 660 segments was divided into three sets (training, validation, and test) where each one had balanced class ratios: 264 segments for the training, 66 for the validation, and 330 for the test. The classification results were obtained on the test set.

With regard to measurement in binary classification problems, there are many measures that have been introduced in the literature. Although correct classification accuracy is the most popular one, probability excess was chosen for the study because of its independency of the relative class frequency. It can be concisely represented by sensitivity + specificity – 1. Sensitivity and specificity show the correct classification accuracies for each one of the two classes.

In neural network models, the biggest fear is overfitting, and some techniques are implemented to prevent overfitting. The best known of them, which is also used in multilayered perceptron network applications, is the use of the validation set as the stop criterion for training. In this study, it was used to avoid determination of a specific solution from the training set. On the other hand, although the coefficients of discretization (k) and equal time slicing (c) might naturally to lead to overfitting, in an interesting manner, the proposed network model was determined to be self-protected against overfitting. When k and c values continued to increase after reaching the optimum value, the classification success of the network decreased or stayed stable.

In the training procedure, there are some parameters that need to be managed, such as learning rate and momentum coefficient. In addition, the activation function for the neurons should be selected according to the problem. Since all of these replaceable parameters can improve the performance of the system, the network should be tested by the different values of these parameters. According to the trial-and-error method, the various options of these three parameters (learning rate, momentum coefficient, and activation function) were tried, but the contributions of both the momentum coefficient and activation function did not have any positive effect on the success ratio. The effect of momentum coefficient (α) values on the success is shown in the figure below.

As seen in Figure 7, the momentum coefficient has no positive effect on the classification success. On the contrary, it has a negative effect in many points. Only when $\alpha = 0.21$ does the classification success reach the same ratio as $\alpha = 0$ in fewer iterations. The results are presented in the figure below.

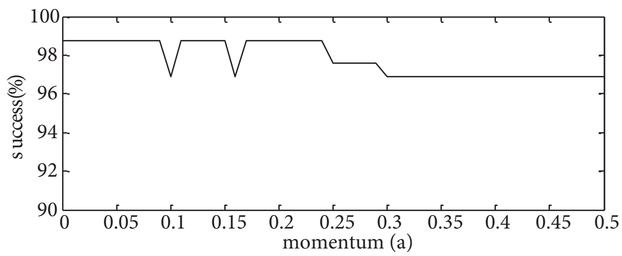

Figure 7. The relation between the classification success and momentum coefficient (for $k = 5$, $c = 32$, and $\eta = 0.006$).

As seen in Figure 8, while the iteration number exceeds 300 for $\alpha = 0$, the iteration number stays under 250 for $\alpha = 0.21$. There is basically no difference in the success, the entropy, or the inconsistency because of the inconsistency alarm. Therefore, the momentum coefficient and activation function are regarded as $\alpha = 0.21$ and the sigmoid activation function in the experiments.

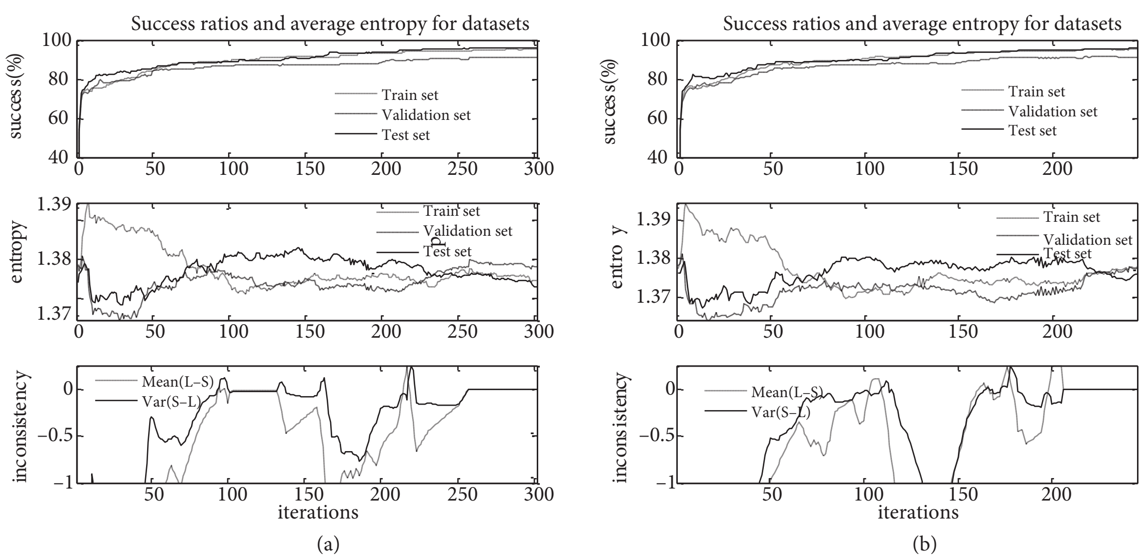

Figure 8. The performance results of the experiments for $\alpha = 0$ (a) and $\alpha = 0.21$ (b).

All classification experiments were performed ten times with different datasets (training, validation, and test sets). According to the best results obtained, the variation in the classification success ratio for different values of the coefficients of discretization (k) and equal time slicing (c) is shown in the figure below.

According to the best results in Figure 9, the optimum values of the coefficients of discretization and equal time slicing are $k = 5$ and $c = 32$, respectively. The classification experiments performed another ten times on different datasets using the parameters k and c provided the best success, and the result is given in the Table for comparison with the results of some studies that used the same database and presented with a validation method.

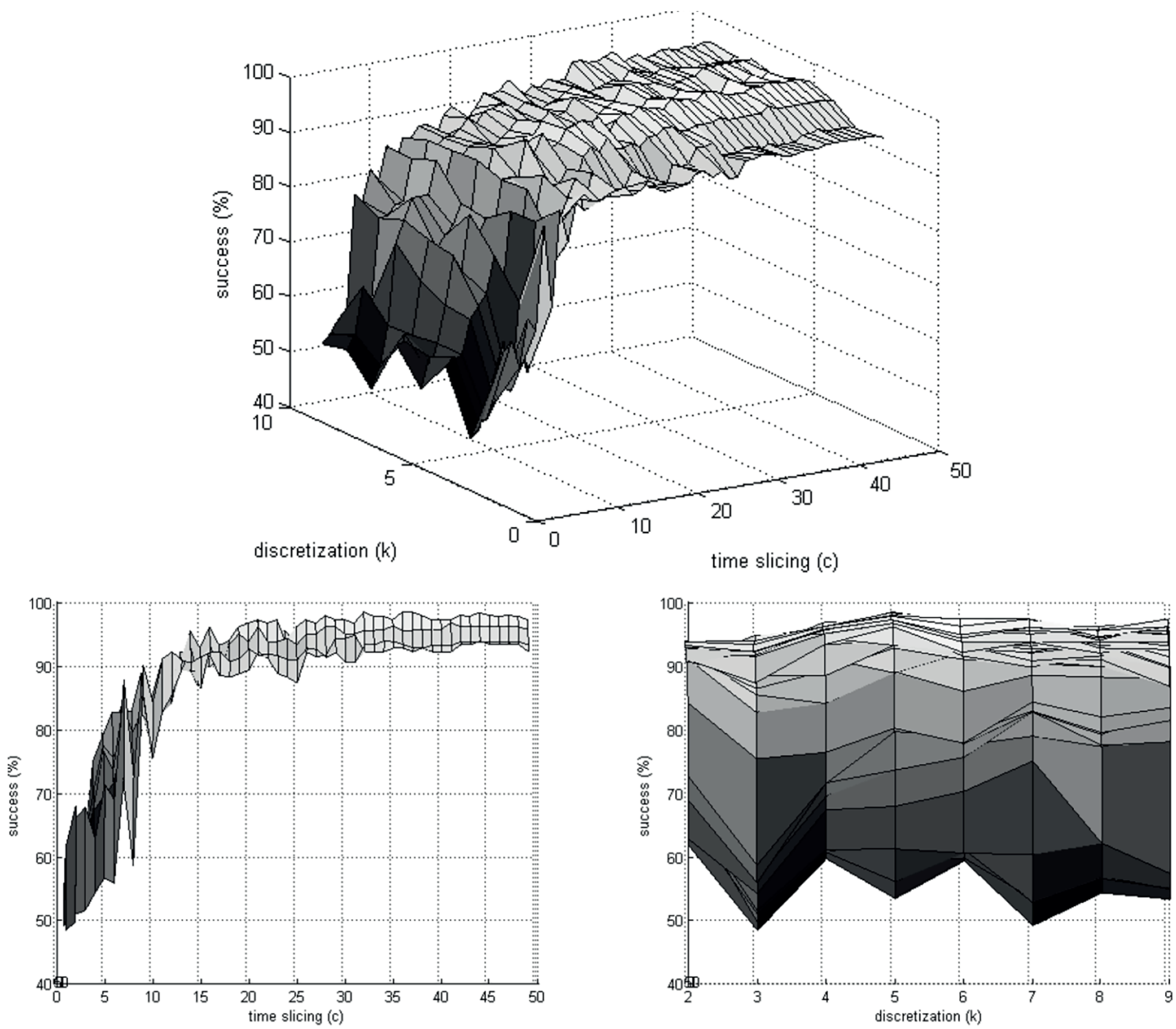

Figure 9. The effects of k and c coefficients on the success for $\alpha = 0.21$ and $\eta = 0.006$.

As seen in the Table, the success of the proposed study is in a good position among those of the studies on the same database. When we focus on the proposed method and Orhan's EFiA-EWiT method [29], this result may change the features used. Because their cut points of discrete intervals are different from each other, discretized signals produced by the proposed method and EFiA-EWiT have different statistical features. In addition to different statistical features and thus classification success, the proposed method is better than EFiA-

EWiT with regard to computational complexity. Orhan's EFiA-EWiT approach first obtains some features by using a discretization method that sorts samples of every ECG signal in the dataset. Then the approach classifies the features with a multilayered perceptron network. Thus, it spends most of its time in both sorting samples and training the network. On the contrary, the SFD approach simultaneously deals with training its perceptron layer and adjusting its discretization layer. As a theoretical result, SFD has less computational complexity than EFiA-EWiT. However, since the studies did not present any time information, we could not compare the methods with respect to computational time.

Table. The comparison of the best successes of the proposed model and other studies using the same databases.

Model	Sensitivity (%)	Specificity (%)	Probability excess (%)
Proposed study	99.44	99.33	98.77
Orhan [29]	99.36	99.30	98.66
Isler and Kuntalp [42]	82.76	100.00	82.76
Asyali [43]	81.82	98.08	79.90
Narin et al. [44]	82.75	96.29	79.04

5. Conclusion

The study presents the first application of the SFD approach for time series without having expertise on data. Although the approach seems like a simple combination of a known feature extraction method and a conventional classifier, it is a new technique that behaves differently because each method depends on each other in the training procedure. Since the proof of a new method is a difficult process that requires many test procedures, this new method is introduced by carrying out classification experiments using a time series. The chosen database includes only ECG signals. Because an ECG signal has a partially stationary characteristic, its classification process is more difficult than it would be for completely stationary signals. On the other hand, because the proposed method can be used without having expertise on data, comprehensive data like ECG signals are enough to introduce one such method. Usually ECG signals are interpreted according to the characteristics of their waves (P, Q, R, S, T), but detection of these waves and their features need ECG signal expertise. When a method independent of data is desired, more general features should be used, as in this study.

To perform the experiments, a dataset was prepared from an ECG database with two classes composed of both healthy subject and patient patterns. All classification experiments based on the SFD method were performed on this dataset. The SFD method has an algorithm that discretizes time signals by using class information. As in all kinds of neural network models, some of the parameters used in the SFD method proposed in this study for the first time required analysis for the most appropriate values. In addition to the coefficient discretization and equal time slicing, the experiments were repeated several times by changing the momentum coefficient and the learning rate. In the analyses, some details were detected about the parameters. For instance, the momentum coefficient had no effect on the rate of success, and additionally the learning rate was more useful if it was reduced in the moments of discrepancy in the achievement. The success assessment in the experiments was performed by probability excess criterion, and the results were compared with the results of other studies that had used the same database. Even if it did not have the highest success in the comparison, it would be still regarded as substantial because it does not need any other method and it is also independent from expertise on data. However, in order to guarantee success, measurements and tests on time series from different areas are needed. Determination of optimum values of coefficients by trial and error is another handicap of the method. However, when looking at the successful results obtained using the proposed method, it is expected to be useful in various academic fields.

References

[1] Abiyev RH. Fuzzy wavelet neural network based on fuzzy clustering and gradient techniques for time series prediction. Neural Comput Appl 2011; 20: 249-259.

[2] Costa I, Schönhuth A, Hafemeister C, Schliep A. Constrained mixture estimation for analysis and robust classification of clinical time series. Bioinformatics 2009; 25: i6-i14.

[3] Dietrich CD, Palm G, Riede K, Schwenker F. Classification of bioacoustic time series based on the combination of global and local decision. Pattern Recogn 2004; 37: 2293-2305.

[4] Jalba AC, Wilkinson MHF, Roerdink JBTM, Bayer MM, Juggins S. Automatic diatom identification using contour analysis by morphological curvature scale spaces. Mach Vision Appl 2005; 16: 217-228.

[5] Legrand B, Chang C, Ong S, Neo SY, Palanisamy N. Chromosome classification using dynamic time warping. Pattern Recogn Lett 2008; 29: 215-222.

[6] Sternickel K. Automatic pattern recognition in ECG time series. Comput Meth Prog Bio 2002; 68: 109-115.

[7] Tseng VS, Kao CP. A novel similarity-based fuzzy clustering algorithm by integrating PCM and mountain method. IEEE T Fuzzy Syst 2007; 15: 1188-1196.

[8] Zeng Z, Yan H. Supervised classification of share price trends. Inform Sciences 2008; 178: 3943-3956.

[9] Garcia S, Luengo J, Saez JA, Lopez V, Herrera F. A survey of discretization techniques: taxonomy and empirical analysis in supervised learning. IEEE T Knowl Data En 2013; 25: 734-750.

[10] Cadenas JM, Garrido MC, Martínez R, Bonissone PP. OFP_CLASS: A hybrid method to generate optimized fuzzy partitions for classification. Soft Comput 2012; 16: 667-682.

[11] Deng H, Runger G, Tuv E, Vladimir M. A time series forest for classification and feature extraction. Inform Sciences 2013; 239: 142-153.

[12] Shehzad K. EDISC: A class-tailored discretization technique for rule-based classification. IEEE T Knowl Data En 2012; 24: 1435-1447.

[13] Lai CP, Chung PC, Tseng VS. A novel two-level clustering method for time series data analysis. Expert Syst Appl 2010; 37: 6319-6326.

[14] Prekopcsák Z, Lemire D. Time series classification by class-specific Mahalanobis distance measures. Adv Data Anal Classif 2012; 6: 185-200.

[15] Jeong YS, Jeong MK, Omitaomu OA. Weighted dynamic time warping for time series classification. Pattern Recogn 2011; 44: 2231-2240.

[16] Wong AKC, Chiu DKY. Synthesizing statistical knowledge from incomplete mixed-mode data. IEEE T Pattern Anal 1987; 9: 796-805.

[17] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence; 1993.

[18] Liu H, Setiono R. Feature selection via discretization. IEEE T Knowl Data En 1997; 9: 642-645.

[19] Wu X. A Bayesian discretizer for real-valued attributes. Comput J 1996; 39: 688-691.

[20] Boulle M. MODL: A Bayes optimal discretization method for continuous attributes. Mach Learn 2006; 65: 131-165.

[21] Zhang G, Hu L, Jin W. Discretization of continuous attributes in rough set theory and its application. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems; 2004.

[22] Yang Y, Webb GI, Wu X. Discretization methods. In: Maimon O, Rokach L, editors. Data Mining and Knowledge Discovery Handbook. New York, NY, USA: Springer, 2010. pp. 101-116.

[23] Liu H, Hussain F, Tan CL, Dash M. Discretization: an enabling technique. Data Min Knowl Disc 2002; 6: 393-423.

[24] Yang Y, Webb GI. Discretization for naive-Bayes learning: managing discretization bias and variance. Mach Learn 2009; 74: 39-74.

[25] Bay SD. Multivariate discretization for set mining. Knowl Inf Syst 2001; 3: 491-512.

[26] Acosta-Mesa HG, Nicandro CR, Daniel-Alejandro GL. Entropy based linear approximation algorithm for time series discretization. Res Comp Science 2007; 32: 214-224.

[27] Orhan U, Hekim M, Ozer M. Epileptic seizure detection using probability distribution based on equal frequency discretization. J Med Syst 2012; 36: 2219-2224.

[28] Orhan U, Hekim M, Ozer M. EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. Expert Syst Appl 2011; 38: 13475-13481.

[29] Orhan U. Real-time CHF detection from ECG signals using a novel discretization method. Comput Biol Med 2013; 43: 1556-1562.

[30] Quilan JR. Induction of decision trees. Mach Learn 1986; 1: 81-106.

[31] Quinlan JR. C4.5: Programs for Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1993.

[32] Catlett J. On changing continuous attributes into ordered discrete attributes. In: Proceedings of the 5th European Working Session on Learning; 1991.

[33] Holte RC. Very simple classification rules perform well on most on most commonly used datasets. Mach Learn 1993; 11: 63-90.

[34] Ho KM, Scott PD. Zeta: A global method for discretization of continuous variables. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining; 1997.

[35] Kurgan L, Cios KJ. CAIM discretization algorithm. IEEE T Knowl Data En 2004; 16: 145-153.

[36] Tsai CJ, Lee CI, Yang WP. A discretization algorithm based on class-attribute contingency coefficient. Inform Sciences 2008; 178: 714-731.

[37] Chan CC, Bartur C, Srinivasan A. Determination of quantization intervals in rule based model for dynamic systems. In: Proceedings of the IEEE Conference on Systems, Man, and Cybernetics; 1991.

[38] Kerber R. ChiMerge: Discretization of numeric attributes. In: Proceedings of the 10th Conference on Artificial Intelligence; 1992.

[39] Liu H, Setiono R. Feature selection via discretization. IEEE T Knowl Data En 1997; 9: 642-645.

[40] Boulle M. Khiops: a statistical discretization method of continuous attributes. Mach Learn 2004; 55: 53-69.

[41] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. Physiobank, Physiotoolkit, and Physionet: components of a new research resource for complex physiologic signals. Circulation 2000; 101: e215-e220.

[42] Isler Y, Kuntalp M. Heart rate normalization in the analysis of heart rate variability in congestive heart failure. P I Mech Eng H 2010; 224: 453-463.

[43] Asyali MH. Discrimination power of long-term heart rate variability measures. In: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2003.

[44] Narin A, Isler Y, Ozer M. Investigating the performance improvement of HRV indices in CHF using feature selection methods based on backward elimination and statistical significance. Comput Biol Med 2014; 45: 72-79.