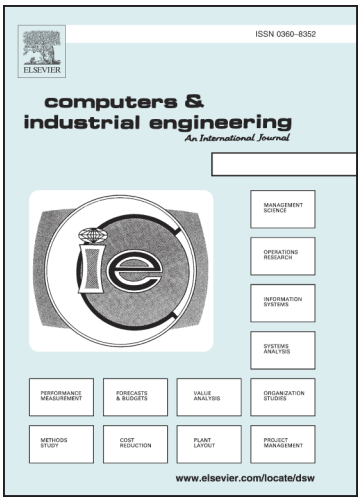
Journal Pre-proofs



Efficient Genetic Algorithm for Feature Selection for Early Time Series Classification

Gilseung Ahn, Sun Hur

PII: S0360-8352(20)30079-6
DOI: https://doi.org/10.1016/j.cie.2020.106345
Reference: CAIE 106345

To appear in: *Computers & Industrial Engineering*

Received Date: 28 March 2019
Revised Date: 9 January 2020
Accepted Date: 6 February 2020

Please cite this article as: Ahn, G., Hur, S., Efficient Genetic Algorithm for Feature Selection for Early Time Series Classification, *Computers & Industrial Engineering* (2020), doi: https://doi.org/10.1016/j.cie.2020.106345

Efficient Genetic Algorithm for Feature Selection for Early Time Series Classification

Gilseung Ahn[ahn.kilseung@gmail.com], Sun Hur[*,hursun@hanyang.ac.kr]

Department of Industrial and Management Engineering, Hanyang University, Ansan, 15588, Korea

[*]Corresponding author.

Highlights

A multi-objective feature selection for early classification is considered.

Starting time of classification is important for early classification.

Our model minimizes the starting time and execution time of classification.

We designed a genetic algorithm with better performance and faster convergence.

Abstract

This paper addresses a multi-objective feature selection problem for early time series classification. Previous research has focused on how many features to consider for a classifier, but has not considered the starting time of classification, which is also important for early classification. Motivated by this, we developed a mathematical model for which the objectives are to maximize classification performance and minimize the starting time and execution time of classification. We designed an efficient genetic algorithm to generate solutions with high probability. In experiment, we compared the proposed algorithm and general genetic algorithm under various experimental settings. From the experiment, we verified that the proposed algorithm can find a better feature set in terms of classification performance, starting time and execution time of classification than feature set found by general genetic algorithm.

Keywords: Time Series Classification; Earliness; Feature Selection; Genetic Algorithm

1. Introduction

Time series classification is used to predict the class label of a time series instance by a well-trained classifier (Deng *et al.*, 2013). That is, if a time series instance i, $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \cdots, x_T^{(i)}\right)$, is given, its label, $y^{(i)}$, is predicted by the classifier $f(\cdot)$ as $\hat{y}^{(i)} = f\left(\boldsymbol{x}^{(i)}\right)$. Time series classification is used to accomplish tasks in many fields, including fault detection in the manufacturing field (Lee *et al.*, 2017), disease diagnosis in the medical field (Lacy *et al.*, 2018), and stock trend analysis in the financial field (Moews *et al.*, 2019).

Various classifiers such as neural networks (NNs) and support vector machines (SVMs) are employed and modified to classify time series. Ignatov (2018) employed a convolutional neural network (CNN) to recognize human activity from accelerometer data. Kim and Cho (2018) developed a C-LSTM (CNN- Long Short-Term Memory model) NN to detect anomalies in web traffic data. CNNs and LSTMs in the developed model extracted spatial features and temporal characteristics, respectively. Emoto *et al.* (2018) used NNs to detect low-intensity snoring episodes from a sleeping sound dataset. Cheng and Dong (2019) employed SVM technology to monitor the nanomachining process with respect to the machining performance. Kalantarian *et al.* (2016) used SVMs to segment streaming time-series audio signals probabilistically.

Earliness is critical for time series classification in time-sensitive applications such as detecting a medical patient's anomaly or predicting equipment unavailability (Xing *et al.*, 2012). Because many applications of time series classification are time-sensitive, early classification of time series is an important research topic. Its goal is to classify a time series instance as soon and accurate as possible (Rodríguez *et al.*, 2001). To achieve this goal, various pre-processing techniques and efficient classifiers have been proposed. For example, Hills *et al.* (2014) constructed time series classifiers based on shapelet transformation to calculate similarity between two time series instances in a short time. Martinez *et al.* (2018) proposed a deep reinforcement learning framework for early classification of time series, by considering trade-off between earliness and classification accuracy. Hatami and Chira (2013) developed an ensemble for early classification of gas signal. The main idea of the ensemble is to use the agreement of an ensemble to decide to accept the candidate label after observing

part of time series. If the agreement ratio is larger than cut-off, then the ensemble does not wait until the whole time series is collected but determines the label immediately with the part of time series. Xi *et al.* (2006) pointed out that dynamic time warping (DTW), a typical distance metric between two time series instances, is very expensive in terms of computational cost and thus is not suitable for early time series classification. In this regard, they proposed a computational reduction technique to speed up one-nearest-neighbor DTW.

Because the early classification of time series has multi-objectives including maximizing classification performance and minimizing classification time, it can be regarded as multi-objective optimization problem (Mori *et al.*, 2017). In general, there exist two or more Pareto optimal solutions for this problem, it is not clear which solution is superior. In this regard, many methods have been proposed to compare solutions with multi-objectives. Typical methods are weighted global criterion method, weighted sum method, lexicographic method, weighted min-max method, bounded objective function methods and so forth (Marler and Arora, 2004). Among them, the weighted global criterion method is one of the easiest and the most popular methods which convert the original multi-objectives into a single objective by linearly combining them, where weight for each objective function is determined by a user (Kasimbeyli, 2013).

The number of dimensions of a time series is a critical factor affecting earliness, and many researchers have tried to develop dimension reduction techniques, such as feature selection, for time series. Feature selection involves composing a subset of the feature set according to objectives such as the maximization of classification performance and/or minimization of classification time of a pre-defined classifier. Because the feature set for a time series is usually too large to compare all possible subsets, meta-heuristic algorithms such as genetic algorithms are employed to solve the problem.

Table 1 summarizes the existing studies on feature selection with respect to meta-heuristic algorithms and their objectives.

As seen in this table, previous studies considered that the number of selected features (i.e., subset size) affected the classifier's training and execution time of a pre-defined classifier. However, they did not consider the starting time of classification, which is also important for earliness, since the classification completion time is the sum of starting time and execution time. As far as our survey is concerned, this is the first research to present a multi-objective mathematical model for feature selection considering the performance, execution time, and

starting time of classification.

The major contents and contributions of this paper can be summarized as follows. First, this paper establishes a multi-objective mathematical model for feature selection considering the performance, execution time, and starting time of classification. The execution and starting time of the classification are measured by means of the selected feature vector. Second, it develops an efficient GA to solve the established model. GA is selected because the way to handle the probability is suitable for our problem. That is, the probability to select a specific feature during the process of the algorithm should be adaptively determined according to the feature index and the number of candidate features, which is relatively easier in GA than other heuristic methods. However, general GA may search inferior solution spaces including solutions to select many features resulting in long execution and starting time of classification.

The rest of this paper is organized as follows. Section 2 describes the problem and the mathematical model. Section 3 explains the proposed GA for solving the established model. Section 4 compares the proposed GA and general GAs in various environments, and verifies that the proposed GA outperformed the general GA. Finally, Section 5 concludes the paper and suggests future research.

2. Problem Description and Mathematical Model

The feature selection problem we considered was to find the subset S of feature set $\Omega = \{x_1, x_2, \cdots, x_T\}$, so that the classification performance of a pre-selected classifier with S is maximized and the execution time and starting time of classification are minimized. The classification performance was measured by F-measure, which is the harmonic mean of precision and recall, and is described by the following equations:

$$\mathrm{F-measure} = 2 \times (\mathrm{precison} \times \mathrm{recall})/(\mathrm{precison} + \mathrm{recall}), \quad (1)$$

$$\mathrm{precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\mathrm{recall} = \frac{TP}{TP + FN}, \quad (3)$$

where TP, TN, FP, and FN indicate true positive (TP), true negative (TN), false positive (FP) and false negative (FN), respectively.

The execution time of classification for a time series instance $\boldsymbol{x}^i$ is defined as difference between time the instance enters a classifier and time the classifier determines class of the instance. It is proportional to the number of selected features (i.e., length of $\boldsymbol{x}^i$). In addition, classification for a time series instance can start only after all values of selected features are collected. Therefore, the starting time of classification is the time that the value of the last feature is collected. Thus, the starting time of classification is proportional to the last index among selected features. For example, suppose there is a sensor measuring a certain value every second, and two feature subsets $S_1 = \{x_1, x_2, x_3\}$ and $S_2 = \{x_1, x_2, x_{100}\}$ are considered. Since their sizes are the same, it may take approximately the same time to execute the classifiers with S_1 and S_2. However, the classifier with S_2 can start to classify an instance after 100 seconds, while the other classifier S_1 can start after only 3 seconds.

The mathematical model for the problem under consideration is constructed as follows:

$$\text{Minimize} \quad z_1 = 1 - F(S), \tag{4}$$

$$\text{Minimize} \quad z_2 = \frac{|S|}{T}, \tag{5}$$

$$\text{Minimize} \quad z_3 = \frac{\max\{t \mid x_t \in S\}}{T}, \tag{6}$$

where $F(S)$ is the F-measure of a classifier with feature subset S, which is a decision variable. Equation (4) is for maximizing the classification performance, Equation (5) is for minimizing the number of selected features per unit time, and Equation (6) is for minimizing the last feature to be selected. We unify objective functions (4) − (6) by a weighted sum as follows:

$$\text{Minimize} \quad Z = w_1 z_1 + w_2 z_2 + w_3 z_3, \tag{7}$$

where w_k $(k = 1, 2, 3)$ is weight for z_k and $w_1 + w_2 + w_3 = 1$. Note that a user can determine w_k by considering relative importance of z_k. For example, a user who thinks z_2 and z_3 are equally important, and z_1 is twice as important as z_2, can determine $w_1 = 0.5$, $w_2 = 0.25$ and $w_3 = 0.25$.

3. The Proposed Genetic Algorithm

The proposed GA is unique in terms of genetic operators, and this section focuses on describing the operators. Readers can refer to Yang and Honavar (1998) for the process of using GAs for feature selection.

Solution p $(p = 1, 2, \cdots, P)$ in the h^{th} $(h = 1, 2, \cdots, H)$ generation used for a genetic algorithm is expressed as follows:

$$\boldsymbol{c}_p^h = \left(c_{p,1}^h, c_{p,2}^h, \cdots, c_{p,T}^h\right), \tag{8}$$

where the element $c_{p,t}^h$ is binary, and is defined as follows:

$$c_{p,t}^h = \begin{cases} 1, \text{ if } x_t \text{ is selected by } \boldsymbol{c}_p^h, \\ 0, \quad \text{otherwise.} \end{cases} \tag{9}$$

That is, the feature subset composed by $\boldsymbol{c}_p^h$ is $S = \left\{s_t | c_{p,t}^h = 1\right\}$.

The initial solution $\boldsymbol{c}_p^1$ is generated with the following probability:

$$\Pr\left(c_{p,t}^1 = 1\right) = \left(1 - \frac{t}{T+1}\right)^{(1-w_1)} \times 0.5 \quad \text{for all} \quad t. \tag{10}$$

This probability is always smaller than or equal to 0.5, which implies that the proposed GA selects features with low probability. It becomes smaller as the feature index t is bigger; therefore, those features whose values are measured later tend to be selected with smaller probability. This can help the second and third objective function values be minimized. Finally, the probability decreases as the weight, w_1, of the first objective function increases. This means that the proposed GA searches through many different potential solutions when the classification performance is more important than the start time and execution time. If we let $w_1 = 1$, i.e., if we consider the classification performance only, then the probability becomes 0.5, which reduces to the general GA.

A crossover operator is defined similarly to the initial solution-generating operator. Let $\boldsymbol{c}_p^h$ be the child of two parental chromosomes $\boldsymbol{c}_{p_1}^{h-1}$ and $\boldsymbol{c}_{p_2}^{h-1}$. Then, $c_{p,t}^h$ $(t = 1, 2, \cdots, T)$ is probabilistically determined by $c_{p_1,t}^{h-1}$ and $c_{p_2,t}^{h-1}$ as follows:

$$\Pr\left(c_{p,t}^h = 1 | c_{p_1,t}^{h-1}, c_{p_2,t}^{h-1}\right) = \left(1 - \frac{t}{T+1}\right)^{(1-w_1)} \times \frac{c_{p_1,t}^{h-1} + c_{p_2,t}^{h-1}}{2}. \tag{11}$$

Therefore, the probability of equation (11) is 0 when both $c_{p_1,t}^{h-1}$ and $c_{p_2,t}^{h-1}$ are 0, and is $\left(1-\frac{t}{T+1}\right)^{(1-w_1)}$ when both are 1. But if $c_{p_1,t}^{h-1} \neq c_{p_2,t}^{h-1}$, then the probability becomes $\left(1-\frac{t}{T+1}\right)^{(1-w_1)} \times 0.5$. From this, we can see that the child's chromosome always becomes 0 with probability one if both genes of parents are 0. In contrast, the probability may not be one even though both genes of parents are all 1.

Finally, the proposed GA employs flip bit mutation operators (usually adopted by general GAs) to prevent most solutions from becoming zero vectors. The GA randomly selects with probability p, and exchanges elements 1 and 0, as depicted in **Figure 1**. As an example, randomly selected $c_{p,2}^h$, $c_{p,5}^h$, and $c_{p,8}^h$ (shaded cells in the figure) are changed to $1-c_{p,t}^h$ $(t=2,5,8)$ by the operator.

A pseudocode of the proposed algorithm is presented as follows.

Input: $S=\{x_1, x_2, \cdots, x_T\}$, P, H, w_1, w_2, w_3, n_s, n_m

Procedure

Step 1. Initialize the best score as zero and h as 1.

Step 2. Generate initial solutions $\boldsymbol{c}_p^1$ for $p=1, 2, \cdots, P$ using equation (10)

Step 3. Score every solution in the current generation using the objective function in (7)

Step 4. Find the solution with the lowest score in the current generation, and update best solution as the solution and best score as its score if its score is smaller than the best score

Step 5. Initialize future generation as an empty set

Step 6. Select $n_s \leq P$ solutions from the initial solutions based on the objective function value in (7) and append them to future generation

Step 7. Repeat to generate a child by crossover using equation (11) for randomly selected two solutions in the future generation $P-n_s$ times

Step 8. Randomly select $n_m \leq P$ solutions in the future generation and apply the flip bit mutation operator.

Step 9. Update current generation as the future generation.

Step 10. Increase h by 1. If h equals to H, then terminate this algorithm and return the best solution. Otherwise, go to **Step 3**.

4. Experiment and Results

4.1 Datasets and Process

The objective of the experiment is to verify that the proposed GA is efficient by comparing the classification performance of well-known time series classifiers, where one classifier is trained with the features selected by the proposed GA, and another is trained with the features by general GA under various experimental condition. Here, the general GA uses $\Pr\left(c_{p,i}^{1}=1\right)=0.5$ for all p and i to generate initial solutions, and $\Pr\left(c_{p,i}^{h}=1 \mid c_{p_1,i}^{h-1}, c_{p_2,i}^{h-1}\right)=\frac{c_{p_1,i}^{h-1}+c_{p_2,i}^{h-1}}{2}$, where $\boldsymbol{c}_{p_1}^{h-1}$ and $\boldsymbol{c}_{p_2}^{h-1}$ are the parents of $\boldsymbol{c}_{p}^{h}$, as explained in the Section 3.

GA parameters are set as following: (1) the maximum number of iterations is 50, (2) the number of solutions in each generation is 20, (3) the number of selected solutions in each generation is 10 (that is, 10 solutions with the smallest objective function values are selected), (4) the ratio of mutations is 0.2, (5) the parameter p, of the flip bit mutation operator, is 0.1. In general, searching for a solution includes training a classifier, which requires extended computer time, but the proposed GA searches for a much smaller number of solutions. This is one of the strong points of the proposed GA.

Five benchmark datasets for time series classification were obtained from UEA & UCR Time Series Classification Repository (http://www.timeseriesclassification.com). The information on these datasets is presented in **Table 2**.

The considered weight settings are $(w_1, w_2, w_3) = (1/3, 1/3, 1/3), (1/2, 1/4, 1/4), (1/4, 1/2, 1/4), (1/4, 1/4, 1/2), (1/7, 3/7, 3/7), (3/7, 1/7, 3/7), (3/7, 3/7, 1/7)$ and the classifier types are NN and SVM. Therefore, a total of 5 (datasets) × 7 (weight settings) × 2 (classifiers) = 70 cases are considered.

4.2 Results

Table 3 shows the results for the general GA and the proposed GA in terms of the objective

function value. We stopped the iteration when the number of iterations becomes 100. In the table, columns "General GA" and "Proposed GA" indicate the objective function values from the feature set selected by the two GAs, and "Difference" is the value of "General GA" minus that of "Proposed GA". Positive values of "Difference" imply that the proposed GA showed better results than general GA.

As seen in **Table 3**, among 70 cases, only 9 cases showed that the general GA revealed better results than the proposed GA, and thus we conclude the proposed GA outperforms the general GA under most conditions. To be more specific, the proposed GA showed better results (i.e., smaller objective function values) except with dataset #1, whose length is 24 (which is very short for a time-series). This is natural because the proposed GA searches a relatively small number of feature subsets, and therefore may be inappropriate to the small feature sets. As mentioned previously, however, time series datasets are usually long enough to apply the proposed GA.

Figure 2 shows the graphs of the objective function values of features selected by two GA's, versus the number of iterations with dataset #2, for which the weights of objectives z_1, z_2, and z_3 are (1/3, 1/3, 1/3) and the NN is applied. The general GA attains its minimum objective value when the number of iterations reached 50, while the corresponding number of iterations for our proposed GA was 33. It is clear that the proposed GA not only showed better performance, but also converged faster than the general GA.

5. Conclusion

Early time series classification is one of the most important issues in many industrial fields. This paper addressed it by focusing on multi-objective feature selection, for which the objectives are based on performance, execution time, and starting time of classification. The feature selection problem was mathematically modeled and solved by means of an efficient GA we proposed in this research. We verified that the proposed GA outperformed the general GA under various settings in terms of efficiency and effectiveness.

In practice, the feature selection result derived by the proposed GA can help the manager in the manufacturing industry, especially controlled by many equipment sensors, decide which sensors are either critical or unnecessary for classification task. As each sensor

corresponds to the feature in the GA, it gives an effective and efficient way to select minimum number of sensors that achieves the best performance with the earliest start time of classification, which obviously leads to the remarkable reduction of management cost and production time.

This research focused on univariate time-series and thus the mathematical model and algorithm cannot be applied to obtain features for multivariate time-series. The model and algorithm for multivariate time-series will be investigated by the authors in the future.

Acknowledgement

This work has supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(2019R1A2C1088255).

References

Cheng, F., & Dong, J. (2019). Monitoring tip-based nanomachining process by time series analysis using support vector machine. *Journal of Manufacturing Processes*, *38*, 158-166.

Deng, H., Runger, G., Tuv, E., & Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, *239*, 142-153.

Emoto, T., Abeyratne, U. R., Kawano, K., Okada, T., Jinnouchi, O., & Kawata, I. (2018). Detection of sleep breathing sound based on artificial neural network analysis. *Biomedical Signal Processing and Control*, *41*, 81-89.

Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization, *Information Sciences*, *422*, 462-479.

Hatami, N., & Chira, C. (2013). Classifiers with a reject option for early time-series classification. IEEE Symposium on Computational Intelligence and Ensemble Learning, April.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, *28*(4), 851-881.

Huang, B., Buckley, B., & Kechadi, T. M. (2010). Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications, *Expert Systems with Applications*, *37*(5), 3638-3646.

Ignatov, A. (2018). Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, *62*, 915-922.

Kalantarian, H., Mortazavi, B., Pourhomayoun, M., Alshurafa, N., & Sarrafzadeh, M. (2016). Probabilistic segmentation of time-series audio signals using Support Vector Machines. *Microprocessors and Microsystems*, *46*, 96-104.

Kasimbeyli, R. (2013). A conic scalarization method in multi-objective optimization. *Journal of Global Optimization*, *56*(2), 279-297.

Kim, T. Y., & Cho, S. B. (2018). Web traffic anomaly detection using C-LSTM neural networks. *Expert Systems with Applications*, *106*, 66-76.

Lacy, S. E., Smith, S. L., & Lones, M. A. (2018). Using echo state networks for classification: A case study in Parkinson's disease diagnosis. *Artificial intelligence in medicine*, *86*, 53-59.

Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, *30*(2), 135-142.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6), 369-395.

Martinez, C., Perrin, G., Ramasso, E., & Rombaut, M. (2018, September). A deep reinforcement learning approach for early classification of time series. *26th European Signal Processing Conference*. *Rome, Italy.*

Mlakar, U., Fister, I., Brest, J., & Potočnik, B. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems, *Expert Systems with Applications*, *89*, 129-137.

Moews, B., Herrmann, J. M., & Ibikunle, G. (2019). Lagged correlation-based deep learning for directional trend change prediction in financial time series. *Expert Systems with Applications*, *120*, 197-206.

Mori, U., Mendiburu, A., Dasgupta, S., & Lozano, J. A. (2017). Early classification of time series by simultaneously optimizing the accuracy and earliness. *IEEE transactions on neural networks and learning systems*, *29*(10), 4569-4578.

Rodríguez, J. J., Alonso, C. J., & Boström, H. (2001). Boosting interval based literals. *Intelligent Data Analysis*, *5*(3), 245-262.

Vignolo, L. D., Milone, D. H., & Scharcanski, J. (2013). Feature selection for face recognition based on multi-objective evolutionary wrappers, *Expert Systems with Applications*, *40*(13), 5077-5084.

Xi, X., Keogh, E., Shelton, C., Wei, L., & Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. Proceedings of the 23rd international conference on Machine learning, June, ACM.

Xing, Z., Pei, J., & Philip, S. Y. (2012). Early classification on time series. *Knowledge and information systems*, *31*(1), 105-127.

Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach, *IEEE transactions on cybernetics*, *43*(6), 1656-1671.

Yang, J., & Honavar, V. (1998). Subset Selection Using a Genetic Algorithm. In H. Liu, & H. Motoda (Eds.), *Feature Extraction, Construction and Selection* (pp. 117 – 136). Boston: Longman, (Chapter 8).

Figure 1. Flip bit mutation operator

Figure 2. Decrease of objective function values of feature sets selected using the general GA and the proposed GA

Table 1. Summarization of feature selection research

Research	Algorithm	Objective functions
Xue *et al.* (2013)	Particle swarm optimization	(1) Minimize error rate (2) Minimize rate of features used
Huang *et al.* (2010)	Genetic algorithm	(1) Maximize overall accuracy (2) Maximize precision (3) Maximize recall
Hancer *et al.* (2018)	Artificial bee colony optimization	(1) Maximize accuracy (2) Minimize number of selected features
Mlakar *et al.* (2017)	Differential evolution	(1) Maximize accuracy (2) Minimize number of selected features
Vignolo *et al.* (2013)	Genetic algorithm	(1) Maximize accuracy (2) Minimize number of selected features

Table 2. Datasets used for the experiment

Dataset	Name	Length	Train size	Test size	Number of classes	Type
#1	Melbourne Pedestrian	24	1200	2450	10	Traffic
#2	Computers	720	250	250	2	Device
#3	FordA	500	3601	1320	2	Sensor
#4	ECG5000	140	500	4500	5	ECG
#5	Wafer	152	1000	6164	2	Sensor

Table 3. Comparison of results for the general GA and the proposed GA

Classifier	Dataset	w_1	w_2	w_3	General GA	Proposed GA	Difference
NN	#1	1/3	1/3	1/3	0.3081	**0.2992**	0.0089
		1/2	1/4	1/4	**0.2934**	0.4347	-0.1413
		1/4	1/2	1/4	0.2293	**0.2273**	0.0020
		1/4	1/4	1/2	**0.3900**	0.4533	-0.0633
		1/7	3/7	3/7	**0.1371**	0.1399	-0.0028
		3/7	1/7	3/7	**0.2785**	0.3866	-0.1081
		3/7	3/7	1/7	**0.3137**	0.3754	-0.0617
	#2	1/3	1/3	1/3	0.2580	**0.1298**	0.1282
		1/2	1/4	1/4	0.2803	**0.1769**	0.1034
		1/4	1/2	1/4	0.2862	**0.0916**	0.1946
		1/4	1/4	1/2	0.2798	**0.1755**	0.1043
		1/7	3/7	3/7	0.2246	**0.0536**	0.1710
		3/7	1/7	3/7	0.2100	**0.1474**	0.0626

		3/7	3/7	1/7	0.3390	**0.1588**	0.1802
	#3	1/3	1/3	1/3	0.2909	**0.1711**	0.1198
		1/2	1/4	1/4	**0.2571**	0.2579	-0.0008
		1/4	1/2	1/4	0.3119	**0.1311**	0.1808
		1/4	1/4	1/2	0.3051	**0.2575**	0.0476
		1/7	3/7	3/7	0.2310	**0.0727**	0.1583
		3/7	1/7	3/7	0.2297	**0.2147**	0.0150
		3/7	3/7	1/7	0.2941	**0.2230**	0.0711
	#4	1/3	1/3	1/3	0.2893	**0.2288**	0.0605
		1/2	1/4	1/4	0.4011	**0.3412**	0.0599
		1/4	1/2	1/4	0.2929	**0.1744**	0.1185
		1/4	1/4	1/2	0.3856	**0.3407**	0.0449
		1/7	3/7	3/7	0.1975	**0.1053**	0.0922
		3/7	1/7	3/7	0.3296	**0.2938**	0.0358
		3/7	3/7	1/7	0.3729	**0.2888**	0.0841
	#5	1/3	1/3	1/3	0.0727	**0.0312**	0.0415
		1/2	1/4	1/4	0.0591	**0.0298**	0.0293
		1/4	1/2	1/4	0.0968	**0.0407**	0.0561
		1/4	1/4	1/2	0.0702	**0.0334**	0.0368
		1/7	3/7	3/7	0.0943	**0.0364**	0.0579
		3/7	1/7	3/7	0.0430	**0.0213**	0.0217
		3/7	3/7	1/7	0.1057	**0.0380**	0.0677
SVM	#1	1/3	1/3	1/3	0.2535	**0.2520**	0.0015
		1/2	1/4	1/4	0.3700	**0.3695**	0.0005
		1/4	1/2	1/4	0.1906	**0.1898**	0.0008
		1/4	1/4	1/2	0.3812	**0.3787**	0.0025
		1/7	3/7	3/7	0.1208	**0.1200**	0.0008
		3/7	1/7	3/7	0.3249	0.3253	-0.0004
		3/7	3/7	1/7	0.3152	**0.3142**	0.0010
	#2	1/3	1/3	1/3	0.2485	**0.1320**	0.1165
		1/2	1/4	1/4	0.2874	**0.1990**	0.0884
		1/4	1/2	1/4	0.2749	**0.0990**	0.1759
		1/4	1/4	1/2	0.2839	**0.1987**	0.0852
		1/7	3/7	3/7	0.2043	**0.0587**	0.1456
		3/7	1/7	3/7	0.2182	**0.1695**	0.0487
		3/7	3/7	1/7	0.3277	**0.1695**	0.1582
	#3	1/3	1/3	1/3	0.2780	**0.1888**	0.0892
		1/2	1/4	1/4	0.2505	0.2818	-0.0313
		1/4	1/2	1/4	0.3069	**0.1416**	0.1653
		1/4	1/4	1/2	0.3022	**0.2823**	0.0199
		1/7	3/7	3/7	0.2149	**0.0805**	0.1344
		3/7	1/7	3/7	0.2274	0.2417	-0.0143

		3/7	3/7	1/7	0.3043	**0.2417**	0.0626
	#4	1/3	1/3	1/3	0.2936	**0.2300**	0.0636
		1/2	1/4	1/4	0.3822	**0.3370**	0.0452
		1/4	1/2	1/4	0.2726	**0.1746**	0.0980
		1/4	1/4	1/2	0.3901	**0.3411**	0.0490
		1/7	3/7	3/7	0.1849	**0.1049**	0.0800
		3/7	1/7	3/7	0.3070	**0.2915**	0.0155
		3/7	3/7	1/7	0.3440	**0.2895**	0.0545
	#5	1/3	1/3	1/3	0.1495	**0.1222**	0.0273
		1/2	1/4	1/4	0.1450	**0.1441**	0.0009
		1/4	1/2	1/4	0.1363	**0.0878**	0.0485
		1/4	1/4	1/2	0.1492	**0.1474**	0.0018
		1/7	3/7	3/7	0.1129	**0.0521**	0.0608
		3/7	1/7	3/7	0.1163	**0.1155**	0.0008
		3/7	3/7	1/7	0.1721	**0.1486**	0.0235