

# Feature selection tree for automated machinery fault diagnosis

Kar Hoou Hui<sup>1,\*</sup>, Ching Sheng Ooi<sup>1</sup>, Meng Hee Lim<sup>1</sup>, and Mohd Salman Leong<sup>1</sup>

<sup>1</sup>Institute of Noise and Vibration, Universiti Teknologi Malaysia Kuala Lumpur, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia

**Abstract.** Intelligent machinery fault diagnosis commonly utilises statistical features of sensor signals as the inputs for its machine learning algorithm. Due to the abundance of statistical features that can be extracted from raw signals and the accuracy of inserting all the available features into the machine learning algorithm for machinery fault classification, less accurate fault classification may inadvertently result due to overfitting issues. It is therefore only by selecting the most representative features that overfitting outcomes can be avoided and classification accuracy be improved. Currently, the genetic algorithm (GA) is regarded as the most commonly used and reliable feature selection tool for the improvement of accuracy for any machine learning algorithm. However, the greatest challenge for GA is that it may fall into a local optima and be computationally demanding. To overcome this limitation, a feature selection tree (FST) is here proposed. Numerous experimental dataset feature selections were executed using FST and GA; their performance is compared and discussed. Analysis showed that the proposed FST resulted in identical or superior optimal feature subsets when compared to the renowned GA method, but with a 20-time faster simulation period. The proposed FST is therefore more efficient in performing feature selection task than GA.

## 1 Introduction

Currently, machine learning is the most efficient tool in eliminating human intervention or assisting human supervision in machinery fault diagnosis. Pattern classification in machine learning is an analytical process used to categorize a dataset into a finite number of assigned divisions based on its characteristics [1]. However, the accuracy of the pattern classification model relies on the quality input features (attributes) and adopted learning algorithm (e.g., artificial neural network, support vector machine (SVM) and Bayesian network). The quantity of features included represents the number of search spaces or dimensions of hyperplanes [2]. The execution time of a learning algorithm and data sampling number required is directly proportional to the number of features necessary to achieve desirable classification accuracy. In other words, larger numbers of the input features require tedious computational and implementation costs, but does not guarantee better performance due to parameter overfitting [3].

Various feature selection techniques have been documented and proven to be capable of efficiently verifying the usefulness of a feature [4]. Forman argued that the emphasis that machine learning algorithms place on quality of features instead of quantity is two-fold: to yield a fulfilling breakthrough and to conserve effort and resources, both at the present time and in the future [5]. Khiabani et al. reported that feature selection significantly improved the performance of a machine learning algorithm, especially when dealing with a vast

number of features [6]. Thus, feature selection plays an important role in the quantitative and qualitative measurement of inputs.

The feature selection method can be classified into the filter method, wrapper method, and embedded method [6–8]. The feasibility of a particular feature subset when subjected to the target function is described by the statistical characteristics of the filter method. For instance, statistical measures utilizing correlation [9], symmetrical uncertainty [10] and optimal relevancy-redundancy relationships [11] have been implemented under the filter method division. A typical wrapper method applies a search mechanism in seeking the best feature combination based on the performance criterion of the classifier. The popular options here chosen are the heuristic, greedy (hill-climbing), stochastic and best-first options [12]. The embedded method integrates the feature selection task into the training process of the classifier [13] before progression to the validation phase.

Noticeably, the wrapper and embedded techniques share a similarity in terms of artificial intelligence involvement in the classifier accuracy comparison and training stages, respectively. In contrast, the filter method independently performs feature ranking of the scoring function in a straightforward manner, while the wrapper and embedded methods acquire feedback from the intended classifier as an indication of the representativeness of a feature subset. Clearly, the application of the performance assessment strategy in possible available subset solutions using an intended classifier is a concern in both the wrapper and embedded

\* Corresponding author: [huikarhoou@gmail.com](mailto:huikarhoou@gmail.com)

methods, especially when dealing with large data. The wrapper method remains an attractive feature selection option as it selects the best combination of features based on its performance with the desired classifier, although a trade-off between computational cost and prediction accuracy is non-trivial. Amongst the alternatives to the wrapper method, GA is regarded as a common and effective technique in this category.

## 2 Genetic Algorithm (GA) and Its Limitations

Since its invention in 1975, GAs have been documented to have enormous development and application in various fields, solving multi-dimensional optimization problems, ranging from grinding operations [14], nanofluid density estimation [15] and a video steganography model [16], to name but a few. Nonetheless, the following case studies explicitly show that GAs opting for large, bulky matrices [17] reduced the partial least square (PLS) model complexity by means of adequately tuned GA feature selection and feature elimination, resulting in better near-infrared (NIR) spectral resolution and interpretation, leading to end result enhancement. By comparing this to the grid algorithm, Huang and Wang obtained better SVM classification performance by simultaneously fine-tuning the feature subset option and the kernel's parameter using GA [18]. The measurement of the GA-SVM model is supported by Wang et al. regarding manipulating the generated electroencephalogram (EEG) signal as an input for a typical brain computer interface (BCI) system [19].

Asghari Oskoei and Hu integrated GA feature selection procedures with an artificial neural network (ANN) classifier [20]. The cascaded model was shown to be more advantageous in classifying myoelectric signals (MES) available in both time and frequency domains. With the aim of achieving input adaptability and model classifier simplicity in a biometrics hand system application, Luque et al. utilized a combination of GA and simple classifiers (kNN, LDA) [21]. A better classification performance was acquired by taking less than 6.25% of the total features available. Meanwhile, GA outperformed the classification process without feature selection, and also outperformed other feature selection algorithm options and a handwriting character recognition case study by utilizing a statistical extension index combination of the Fisher linear discriminant method and covariance matrices [22]. Oluleye et al. reported the involvement of GAs in exploring optimal input features for novel kNN-classification functions in image recognition. GAs outclassed WEKA software in terms of flexibility and feature size reduction, achieving better accuracy in both Flavia and Ionosphere datasets [23]. The review clearly suggested that GAs are amongst the most common, effective and versatile techniques for optimization purposes, regardless of the type or size of the data or classifier [24].

GA was built on the analogy of Darwin's fundamental genetic evolution theory, to identify an

optimal solution for a targeted function with assistance from the duplication of probable answers in every iteration. Hence, in the case of feature selection, it focuses on discovering the finest binary vector resembling a chromosome with regards to the search space. The fitness is based on optimization criteria, which aim towards preferable values, either the global maxima or minima, located within the search boundary. The heuristic search of the abovementioned learning algorithm starts with the random initialization of a group of candidate solution generations called populations to eliminate bias exploration and possibly cover most of the boundary area [25]. Unless posteriori knowledge is available regarding the targeted data, one is able to reduce it to the high probable area. The candidate solutions are batches of encrypted chromosomes consisting of a unique combination of a binary string (ones and zeros). For each iteration, the fitness calculation process is conducted by involving only values associated with the chromosome value '1' and ignoring those assigned the value '0.' Every individual chromosome is subjected in advance to an independent evaluation function and is compared to other chromosomes during the fitness function in the following stage. Technically, GA-feature selection performs an evolution process by applying a parallel computational system (population evaluation) and intelligence strategy (optimization function) [26]. The computational time and effort are direct proportional to the size of the population group. Thus, a trade-off relationship between accuracy and computational effort (implementation of the algorithm, parameter setting, calculation time and outcome interpretation) is inevitable for the GA wrapper method non-spatial dataset measurement.

## 3 Data Collection

The bearing conditions dataset used in this study was downloaded from the Case Western Reserve University Bearing Data Center website specifically to represent ball bearings in healthy and faulty conditions (rolling element, inner raceway and outer raceway faults). The test rig consisted of a 2-horse power (HP) motor, a torque transducer and a dynamometer. The arrangement of the test rig was used to simulate different conditions of the bearing (Figure 1). The motor operated at approximately 1750 rpm with a 1-HP load. Vibration data were collected at a sampling rate of 12 kHz by accelerometers that attached to the bearing housing.



**Fig. 1.** Experimental test rig.

A total of 400 sets of time series vibrations were extracted from the raw continuous vibration signal collected from a 7-mils fault diameter with a 1-HP load. Then, the 400 sets of vibration data were divided into two sets of data, of which one set was used to establish the relationship between the input and output of the machine learning model (training phase), and the other set was used to validate the trained machine learning model (testing phase). The distribution of the vibration dataset employed in this study is tabulated in Table 1.

**Table 1.** Vibration data distribution.

Bearing Condition	Training Data	Testing Data
Healthy	50	50
Rolling Element Fault	50	50
Inner Raceway Fault	50	50
Outer Raceway Fault	50	50

### 4 Fault feature extraction

In this section, time series vibration data from section 3 is subjected to statistical analyses with the purpose of acquiring statistical features. The obtained features, namely the skewness factor, kurtosis factor, crest factor, shape factor, impulse factor and margin factor, were converted from the corresponding equations in Table 2. Subsequently, the statistical features were used as features (inputs) for SVM model training and testing purposes. Each statistical feature shown has unique characteristics and carries informative data regarding system status.

**Table 2.** Statistical features.

No.	Statistical Feature	Equation
1	Skewness Factor	$\frac{\frac{1}{N} \sum_{n=1}^N (x(n) - \bar{x})^3}{\left(\sqrt{\frac{1}{N} \sum_{n=1}^N (x(n) - \bar{x})^2}\right)^3}$
2	Kurtosis Factor	$\frac{\frac{1}{N} \sum_{n=1}^N (x(n) - \bar{x})^4}{\left(\sqrt{\frac{1}{N} \sum_{n=1}^N (x(n) - \bar{x})^2}\right)^4}$
3	Crest Factor	$\frac{\max x(n) }{\sqrt{\frac{1}{N} \sum_{n=1}^N x(n)^2}}$
4	Shape Factor	$\frac{\sqrt{\frac{1}{N} \sum_{n=1}^N x(n)^2}}{\frac{1}{N} \sum_{n=1}^N  x(n) }$
5	Impulse Factor	$\frac{\max x(n) }{\frac{1}{N} \sum_{n=1}^N  x(n) }$
6	Margin Factor	$\frac{\max x(n) }{\left(\frac{1}{N} \sum_{n=1}^N \sqrt{ x(n) }\right)^2}$

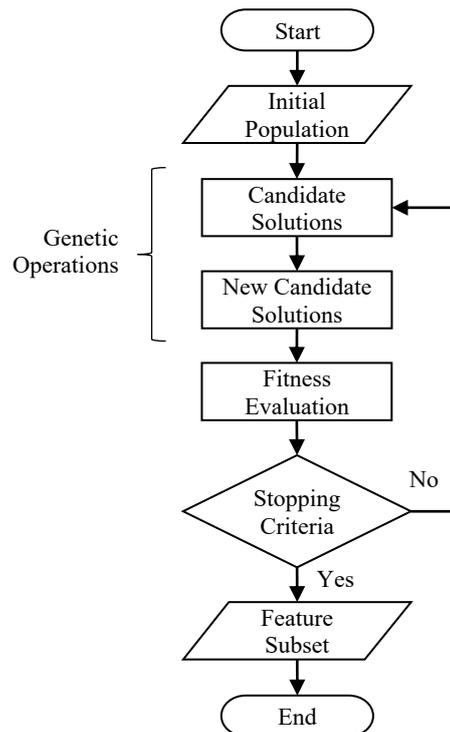
Since there was a total of 100 samples for each bearing condition, 50% of the samples were selected randomly as training data to synthesize the machine learning model, while the remaining 50% of the samples were used to validate the trained machine learning model.

### 5 Feature Selection Methods

This section discusses two different feature selection methods that were implemented. It covers the theory, characteristics, implementation flow and unique yet advantageous ways used in determining the optimal feature subset to feed into a model classifier, so that the targeted system condition prediction performance could be optimized.

#### 5.1 Genetic Algorithm

Figure 2 displays the process of feature selection initiated by GA, whereby the arrow shows the chromosome flow direction into intermediate stages. Briefly, a chromosome commenced as a random candidate solution in a population group and were subjected to an evaluation function to determine whether it would be selected for or eliminated from the next round based on goodness of fit. Genetic operators acted on the chosen chromosomes by performing selection and reproduction processes for seizing a population batch in the next cycle. The cycle continued until one of the stopping criteria were met. The end result was that the most probable feature subset was generated.



**Fig. 2.** GA feature selection flow chart.

The implementation of GAs was realized using MATLAB software (version R2014b). A specialized toolbox was developed for GA fulfilment. The *gaoptimset* extension function was particularly useful because it allows the algorithm parameter setting to be enclosed under a single command. The investigated dataset basically extracted a data matrix describing roller bearing operating conditions. The data matrix was composed of six numerical features derived from the captured vibration signals: skewness factor, kurtosis factor, crest factor, shape factor, impulse factor and margin factor. The interest of the current study was to employ GAs to discover distinctive feature subsets by means of exploiting a standout characteristic to explain the roller bearing situation. The GA-based feature selection algorithm parameter setting is listed in Table 3, below.

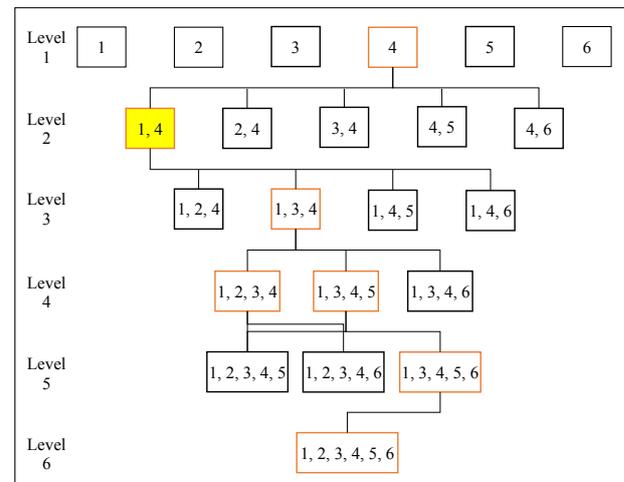
**Table 3.** Genetic algorithm parameter setting.

Option	Parameter Value
PopulationSize	25
Generations	30
PopulationType	Bitstring
SelectionFcn	@selectiontournament
TournamentSize	2
CrossoverFcn	@crossoverarithmetic
EliteCount	2
MutationFcn	@mutationuniform
StallGenLimit	20
TolFun	1 <sup>-8</sup>

### 5.2 The Proposed Feature Selection Tree

In this study, a novel feature selection method named the feature selection tree (FST) is proposed for carrying out the feature selection task. The FST employed the SVM as a wrapper in feature selection. The performance of each feature was based on the SVM training accuracy. The FST shortened execution time by avoiding repeated computations of the performance of identical feature combinations. The repeated assessment of the identical feature combinations usually occurred to the feature selection algorithms that randomly generated feature combinations. This significantly increased unnecessary computational time. Thus, the FST only evaluated unique combinations of features. In addition, the FST generated next level feature combinations based on the performance of the previous level. Fig. 5 illustrates the methodology of the FST algorithm. In the first level selection, the algorithm evaluates each individual feature. Then, the algorithm generates the second level feature combinations by combining unselected individual features with the features that performed above average (red-outlined rectangle in Figure 3). This process terminates when the feature combination has fully utilized all extracted features. Lastly, the algorithm selects feature combinations with the least number of features from the top 5% of the highest training accuracy (yellow-filled rectangle in Figure 3) as the most

representative features of the entire dataset. In addition to selecting the most representative features for the dataset, feature selection also reduces the feature dimensionality for machine learning algorithms. As a result, skewness factor and shape factor (i.e., features 1 and 4) were selected in this example.



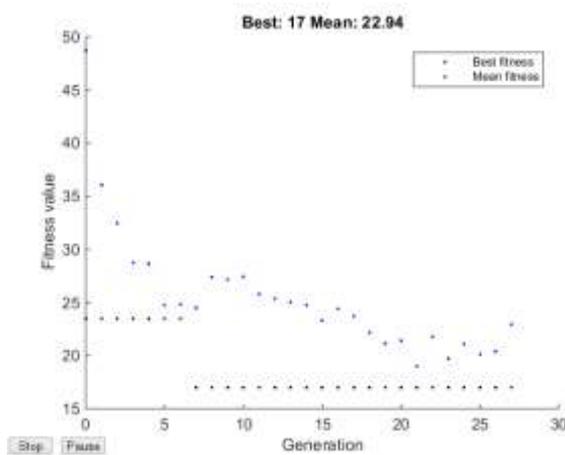
**Fig. 3.** The proposed feature selection algorithm (features 1, 2, 3, 4, 5 and 6 represent skewness factor, kurtosis factor, crest factor, shape factor, impulse factor and margin factor, respectively).

## 6 Results and Discussion

The following section presents, compares and discusses the performance of the aforementioned two types of feature selection method. The performance quality under consideration includes implementation practicability, execution time and accuracy.

### 6.1 Genetic Algorithm

As the MATLAB software GA toolbox is patterned to target global minimal points as the optimization goal, the SVM classifier confusion function was adopted as the optimization function. Confusion can occur when given a set of features, for instance, an SVM classifier unable to predict a roller bearing condition as defined. The GA is responsible for optimize the fitness function by searching for the minimal point of unwanted confusion rate. The lower the error, the higher the classification success rate.



**Fig. 4.** GA-roller bearing dataset simulated fitness evaluation.

Figure 4 illustrates GA-simulated fitness evaluation results for 30 generations using the roller bearing dataset. Two different coloured pointers were plotted over the generations. The mean fitness value portrayed in the blue colour pointer was reduced from 48% at initial simulation to a low point of 19% confusion as the simulated period increased. It converged at approximately 20% towards the end of the evaluation. The GA started with an unbiased, large search space and learned to concentrate into a high probability area and local optimal points. Hence, the average population fitness value decreased over time. It was observed that the mean fitness value experienced a minor peak after the eighth generation. This was due to the appearance of a new elite child, which caused a navigation of offspring for the new probable area during the transient period.

The best fitness value, represented by a black colour pointer, remained steady at 23.5% confusion rate until it dropped to 17% during the seventh generation. It then remained fixed for the rest of the simulation period. The simulation stopped at the 27th generation instead of the 30th since the best fitness value was static for 20 iterations. The GA stopping criteria were supported by the convergence of the pointer lines and a minor gap observed between both pointer lines, commencing at the 20th iteration. Based on the figure inspection, as the generations increased, the GA recruited fitter and fitter population groups but was not assisted in earning improved best confusion rates. Hence, with reference to the stopping criteria, the algorithm was satisfied with the feature subset obtained. A minor perturbation in the mean fitness value may have resulted from the recombination process while attempting to improve population diversity. The best chromosome included features 1 (skewness factor) and 4 (shape factor), providing a lowest confusion rate of 17%. The performance of the GA-SVM classifier was encouraging as the classification accuracy efficacy improved, with only 33% of the total features being preferred compared to a 24% confusion rate while utilizing all six features.

Typically, GA feature selection deals with a large dataset. The robustness of a GA against exponential increment in hypercube dimensions and data noise is

well documented. In this case study, although GAs managed to identify an optimal feature subset, concerns were raised regarding algorithm implementation. The effectiveness of the algorithm largely relied on parameter tuning, particularly population size design. Overdesign of a population is capable of obtaining a fast outcome despite that severe computation cost, low unique rate and chromosome redundancy prompting relatively high average fitness variation. In contrast, an under-designed population ensures a high unique rate of chromosomes but was an unacceptable option, as it led to longer simulation periods for value convergence and, in the worst case, to a fall in the local optimal point. From the evaluation findings, the trade-off effect could be minimized with a parameter adjustment: a population size of 25 yielded amongst the highest number of unique initial chromosomes and a fast and accurate feature subset.

## 6.2 Feature Selection Tree

Table 4 shows the training accuracy of the key combinations of features at each level. The yellow shaded feature combinations are feature combinations above the average training accuracy at each level, and the blue shaded training accuracy designates the top 5% training accuracy in the entire table. As a result, features 1 and 4 (skewness and shape factor) were selected to represent the entire bearing conditions dataset. The training accuracy in Table 4 indicated that entering all extracted features into the machine learning algorithm does not guarantee the highest classification accuracy, as the training accuracy for the selected features (i.e., features 1 and 4) was 81%, and the training accuracy for all extracted features was 74%. In contrast, the testing accuracy of the bearing faults dataset was 83% for the selected features and 76% for all extracted features. Therefore, a representative feature combination for the entire dataset was selected using the proposed FST algorithm.

**Table 4.** Training accuracy for the key combination of features (feature 1, 2, 3, 4, 5 and 6 represent skewness factor, kurtosis factor, crest factor, shape factor, impulse factor and margin factor, respectively).

Level 1		Level 2		Level 3	
Feature	Accuracy	Feature	Accuracy	Feature	Accuracy
1	0.285	12	0.715	123	0.720
2	0.405	14	0.810	124	0.730
3	0.025	16	0.705	125	0.690
4	0.500	23	0.410	126	0.695
5	0.230	24	0.500	134	0.735
6	0.340	25	0.405	136	0.710
		26	0.395	145	0.725
		34	0.500	146	0.735
		36	0.490	156	0.695
		45	0.500		
		46	0.500		
		56	0.375		
Level 4		Level 5		Level 6	
Feature	Accuracy	Feature	Accuracy	Feature	Accuracy
1234	0.735	12345	0.735	123456	0.740
1235	0.715	12346	0.735		
1236	0.735	12356	0.735		
1245	0.725	12456	0.725		
1246	0.720	13456	0.735		
1345	0.735				
1346	0.735				
1456	0.730				

### 6.3 Comparison

Table 5 compares the testing accuracy for features selected by the GA and FST algorithm. The analysis showed classification accuracy with the features selected by the two types of feature selection method is equivalent. Therefore, it was proven that the proposed FST is capable of selecting an equivalent feature subset in comparison to the GA. In the following section, the execution time of the GA and the FST algorithm will be compared.

**Table 5.** Comparison of testing accuracy for features selected by GA and FST.

	Without FS	GA	FST
Selected Features	N/A	1 & 4	1 & 4
Testing Accuracy	76%	83%	83%

Execution time comparison analyses for both feature selection methods are tabulated in Table 6. Generally, the FST execution time was relatively faster than that of

the GA. The comparison showed execution time of the FST is 13 times faster than GA. Thus, the proposed FST can be embedded into the machine learning training process to execute effective automated feature selection.

**Table 6.** Comparison of execution time for GA and FST algorithms.

	GA	FST
Execution Time (second)	36.2	2.8

### 7 Conclusion

In this study, a novel feature selection method designated the feature selection tree (FST) was proposed. The performance of the proposed FST was compared with one of the most commonly used feature selection methods, namely that of the genetic algorithm (GA). Preliminary results showed encouraging feature selection performance, with a notable increase in model classification accuracy using the reduced feature matrix dataset. Evidently, desirable feature size selection led to minimization of overfitting complications, resulting in optimal, unambiguous model interpretation. Further analysis showed that the proposed FST was able to select an equivalent or better feature subset than GA to represent the entire dataset. Although the proposed FST selected identical features as the GA for bearing data, the execution time for the proposed FST was reduced by up to 92% compared to the GA. In summary, the state-of-the-art FST is able to select an equivalent or better feature subset in a shorter execution time when compared to the GA. This is essential when dealing with a large number of inputs. Therefore, the favourable FST can be embedded into machine learning algorithms in order to improve the performance of these algorithms.

### Acknowledgments

The authors would like to extend their deepest gratitude to the Institute of Noise and Vibration UTM for funding the study under the Higher Institution Centre of Excellence (HiCoE) Grant Scheme (R.K130000.7809.4J226, R.K130000.7843.4J227 and R.K130000.7843.4J228).

### References

1. J. Yang and V. Honavar, Feature Subset Selection Using a Genetic Algorithm, (1997).
2. D. Whitley, A genetic algorithm tutorial, *Stat. Comput.* **4**, 65–85 (1994).
3. L. Zhuo, J. Zheng, F. Wang, X. Li, B. Ai, and J. Qian, A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **37**, 397–402 (2008).
4. I. Guyon, A. Elisseeff, and A. M. De, An Introduction to Variable and Feature Selection,

- J. Mach. Learn. Res.* **3**, 1157–1182 (2003).
5. G. Forman, An Extensive Empirical Study of Feature Selection Metrics for Text Classification, *J. Mach. Learn. Res.* **3**, 1289–1305 (2003).
  6. F. B. Khiabani, A. Ramezankhani, F. Azizi, F. Hadaegh, E. W. Steyerberg, and D. Khalili, A tutorial on variable selection for clinical prediction models: Feature selection methods in data-mining could improve the results, *J. Clin. Epidemiol.* **71**, 76–85 (2015).
  7. M. Tutkan, M. C. Ganiz, and S. Akyokuş, Helmholtz principle based supervised and unsupervised feature selection methods for text mining, *Inf. Process. Manag.* **52**, 885–910 (2016).
  8. V. Aksakalli and M. Malekipirbazari, Feature Selection via Binary Simultaneous Perturbation Stochastic Approximation, *Pattern Recognit. Lett.* **75**, 41–47 (2015).
  9. S. Jiang and L. Wang, Efficient feature selection based on correlation measure between continuous and discrete features, *Inf. Process. Lett.* **116**, 203–215 (2016).
  10. A. Moayedikia, K.-L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, Feature selection for high dimensional imbalanced class data using harmony search, *Eng. Appl. Artif. Intell.* **57**, 38–49 (2017).
  11. A. Senawi, H. L. Wei, and S. A. Billings, A new maximum relevance-minimum multicollinearity (MRmMC) method for feature selection and ranking, *Pattern Recognit.* **67**, 47–61 (2017).
  12. R. Kohavi and H. John, Artificial Intelligence Wrappers for feature subset selection, *Artif. Intell.* **97**, 273–324 (1997).
  13. X. Zhang, G. Wu, Z. Dong, and C. Crawford, Embedded feature-selection support vector machine for driving pattern recognition, *J. Franklin Inst.* **352**, 669–685 (2015).
  14. R. Saravanan, P. Asokan, and M. Sachidanandam, A multi-objective genetic algorithm (GA) approach for optimization of surface grinding operations, *Int. J. Mach. Tools Manuf.* **42**, 1327–1334 (2002).
  15. H. Karimi and F. Yousefi, Application of artificial neural network-genetic algorithm (ANN-GA) to correlation of density in nanofluids, *Fluid Phase Equilib.* **336**, 79–83 (2012).
  16. K. Dasgupta, J. K. Mondal, and P. Dutta, Optimized Video Steganography Using Genetic Algorithm (GA), *Procedia Technol.* **10**, 131–137 (2013).
  17. R. Leardi and A. L. Gonzalez, Genetic algorithms applied to feature selection in PLS regression-how and when to use them, *Chemom. Intell. Lab. Syst.* **41**, 195–207 (1998).
  18. C. L. Huang and C. J. Wang, A GA-based feature selection and parameters optimization for support vector machines, *Expert Syst. Appl.* **31**, 231–240 (2006).
  19. L. Wang, G. Xu, J. Wang, S. Yang, M. Guo, and W. Yan, Motor Imagery BCI Research Based on Sample Entropy and SVM, in *2012 6th Int. Conf. Electromagn. F. Probl. Appl. ICEF'2012*, (2012).
  20. M. Asghari Oskoei and H. Hu, Myoelectric control systems-A survey, *Biomed. Signal Process. Control* **2**, 275–294 (2007).
  21. R. M. Luque, D. Elizondo, E. Lopez-Rubio, and E. J. Palomo, GA-based feature selection approach in biometric hand systems, *Proc. Int. Jt. Conf. Neural Networks*, 246–253 (2011).
  22. C. De Stefano, F. Fontanella, C. Marrocco, and A. Scotto Di Freca, A GA-based feature selection approach with an application to handwritten character recognition, *Pattern Recognit. Lett.* **35**, 130–141 (2014).
  23. B. Oluleye, A. Leisa, J. Leng, and D. Dean, A Genetic Algorithm-Based Feature Selection, *Int. J. Electron. Commun. Comput. Eng.* **5**, 899–905 (2014).
  24. P. Moradi and M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, *Appl. Soft Comput.* **43**, 117–130 (2016).
  25. K. A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems (1975).
  26. M. Mitchell, *An Introduction to Genetic Algorithms* (1999).