



# Feature extraction and selection from acoustic emission signals with an application in grinding wheel condition monitoring

T. Warren Liao \*

Industrial Engineering Department, Louisiana State University, Baton Rouge, LA 70803, USA

## ARTICLE INFO

### Article history:

Received 24 May 2007

Received in revised form

10 May 2008

Accepted 30 September 2009

Available online 28 October 2009

### Keywords:

Acoustic emission

Wheel condition monitoring

Feature extraction

Autoregressive model

Discrete wavelet decomposition

Feature selection

Ant colony optimization

Sequential forward floating selection

## ABSTRACT

Feature extraction and feature selection are two important issues in sensor-based condition monitoring of any engineering systems. In this study, acoustic emission signals were first collected during grinding operations, next processed by autoregressive modeling or discrete wavelet decomposition for feature extraction, and then the best feature subsets are found by three different feature selection methods, including two proposed ant colony optimization (ACO)-based method and the famous sequential forward floating selection method. Posing monitoring as a classification problem, the evaluation is carried out by the wrapper approach with four different algorithms serving as the classifier. Empirical test results were shown to illustrate the effectiveness of feature extraction and feature selection methods.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

Grinding is a major finishing process used to fabricate engineering objects to the specified dimensions and surface roughness. Grinding operations are carried out with grinding wheels as tools and the tool condition changes from sharp to dull as the grinding operation progresses. A dull wheel is inefficient, draws higher power, and affects the quality of the ground workpiece. Close monitoring of tool condition is very important to grinding as well as other machining operations. In the current industrial practice, tool condition is often monitored by the operator overseeing the operation.

Efforts have been made in the past to develop a sensor-based grinding wheel condition monitoring system to improve the efficiency and integrity of grinding operations. In their keynote paper on grinding process monitoring, Tonshoff et al. (2002) reviewed the sensors used to monitor both the micro- and macro-topography of the active abrasive layer of the grinding wheel, which recognized as the dominant features of the grinding process. A brief review of past publications is given in chronological order below. Mokbel and Maksoud (2000) analyzed raw AE signals using fast Fourier transform and compared the AE spectral amplitudes of different diamond wheel bond types, grit

sizes, and their conditions generated by using different grinding wheel/truing speed ratios with the surface roughness ( $R_a$ ) of the ground mild steel specimens. Hwang et al. (2000) reported that the amplitude of the AE signal, collected in high-speed grinding of silicon nitride using an electroplated single-layered diamond wheel, monotonically increases with wheel wear. To monitor an alumina wheel surface condition in cylindrical grinding of carbon steel, Lezanski (2001) extracted statistical and spectral features from multiple signals (forces, vibration, and AE) and applied a feed-forward backpropagation neural network to select eight features, which are grinding depth of cut, coolant volume rate, standard deviation of vibration, means value of vibration power spectrum, range of vibration power spectrum, mean value of AE RMS, range of AE RMS, and range of RMS power spectrum. The eight selected features were used to train a neuro-fuzzy model for classification. A best performance of 83.3% classification accuracy was reported.

Furutani et al. (2002) introduced an in-process method for measuring the topography change of an alumina grinding wheel in cylindrical grinding. A pressure sensor is set beside a grinding wheel with a small gap. When grinding fluid is dragged into the gap, hydrodynamic pressure, which corresponds to the gap length and the wheel topography, can be measured. The pressure signals were analyzed by fast Fourier transform (FFT). Higher frequency components of the pressure spectra were found to increase with the wheel loading and dulling. Warkentin and Bauer (2003) investigated the relationships between wheel wear and grinding

\* Tel.: +1 225 5785365; fax: +1 225 5785109.

E-mail address: [ieliao@lsu.edu](mailto:ieliao@lsu.edu)

forces for different depths of cut when surface grinding mild steel with an aluminum oxide wheel. For depths of 10, 20, and 30  $\mu\text{m}$ , respectively, the average normal and tangential forces show a trend with a slight positive slope. For depths of 40 and 50  $\mu\text{m}$ , the data shows a negative slope and piecewise negative slope, respectively.

Hosokawa et al. (2004) developed a practical wheel surface condition monitoring system, in which the characteristics of the wheel surface are discriminated on the basis of the dynamic frequency spectrum signals of grinding sound using a neural network technique. The system was successfully tested in the plunge grinding of carbon steel with a vitrified-bonded alumina wheel and in the plunge grinding of hardened die steel with a resinoid-bonded CBN wheel. Kwak and Ha (2004) showed that the grinding force signal in surface plunge grinding of STD11 with an alumina wheel could be better processed by wavelet de-noising than by FFT filtering. To detect the wheel dressing time clearly, the approximation coefficients A4 of Daubechies wavelet transform were used because they were suitable for the detection of a sudden signal change at the time or frequency domain. Liao et al. (2006) extracted features from acoustic emission signals collected at 1 MHz in grinding alumina with a resin-bonded diamond wheel and then applied an adaptive genetic clustering algorithm to the extracted features in order to distinguish different states of grinding wheel condition. Liao et al. (2007) proposed another clustering method, which involves first extracting features from acoustic emission signals based on discrete wavelet decomposition by following a moving window approach, next generating a distance (dissimilarity) matrix using HMM, and then applying a clustering algorithm to obtain clustering results. Table 1 summarizes the previous studies on sensor-based grinding wheel condition monitoring.

Feature extraction and feature selection are two important issues in sensor-based tool health monitoring. Feature extraction is a process that transforms the original sensory signal into a

number of potentially discriminant features. The above review indicates that feature extraction is often carried out by different methods; but feature selection is rarely studied except Lezanski (2001). Feature selection refers to a process that selects an optimal subset of the extracted features based on an evaluation criterion. By removing the irrelevant and redundant not only reduce the dimension but also may increase the classification accuracy. For a dataset with  $N$  features, there exist  $2^N$  candidate subsets. Even for a moderate  $N$ , the search space is exponentially prohibitive for exhaustive search. The search strategies that have been employed in feature selection can be roughly grouped into three categories:

- (1) Complete search such as branch and bound that guarantees to find the optimal result.
- (2) Sequential search that adds or removes features one (or some) at a time. Three cases can be distinguished: sequential forward selection if starting with an empty set and adding one feature (or some numbers of features) at a time; sequential backward elimination if starting with the entire set of features and removing one feature (or some features) at a time; bidirectional search if the above two are combined.
- (3) Random search such as genetic algorithm that starts with a randomly selected subset and then proceeds to generate the next subset in a completely random manner or with sequential search.

Depending upon how the evaluation is done, a feature selection algorithm can be distinguished either as a filter approach, a wrapper approach, or a hybrid. The filter approach evaluates and selects feature subset based on the general characteristics of the data without involving any learning model. On the other hand, the wrapper approach employs a learning model and uses its performance as the evaluation criterion. Compared with the filter approach, the wrapper approach is

**Table 1**  
Summary of studies on sensor-based grinding wheel condition monitoring.

Reference	Tool state(s)	Grinding conditions	Signal(s) used	Signal analysis	Features	Prediction/detection algorithm	Test results
Furutani et al. (2002)	Wheel loading and dulling	Fixed	Hydrodynamic pressure	FFT	Spectral amplitude	Non used	Not available
Hosokawa et al. (2004)	Wheel wear	Varied dressing feed	Sound	Frequency spectrum	Sound pressure level values within the frequency range at 20 Hz intervals	Neural network	80–100%
Hwang et al. (2000)	Wheel wear	Fixed	AE	Power spectral density	RMS of AE signals, magnitude of peak frequency of AE power spectral density	None used	Not available
Kwak and Ha (2004)	Wheel loading	Fixed	Force	Daubechies wavelet transform	Approximation coefficients A4	None used	Not available
Lezanski (2001)	Wheel wear	Three depths of cut	Forces, vibration, and AE	Statistical and spectral	8 statistical and spectral features selected by a feed-forward BP neural network	Neuro-fuzzy model	83.3% classification accuracy
Mokbel and Maksoud (2000)	Generated by using different grinding/truing speed ratios	Fixed	AE	Fast Fourier transform	Spectral amplitude	None used	Not available
Liao et al. (2006)	Wheel wear	Varied	AE	Discrete wavelet decomposition	Wavelet energy	Adaptive genetic clustering	76.7%
Liao et al. (2007)	Wheel wear	Varied	AE	Discrete wavelet decomposition	Wavelet energy	Hidden Markov model + clustering	75–100% depending upon the grinding condition

known to be more accurate but cost higher computationally. The hybrid approach is designed to trade accuracy with computational speed by applying the wrapper approach to only the set of those features selected by the filtering approach first.

This paper investigates the above-mentioned two important issues in acoustic emission sensor-based condition monitoring of a grinding wheel used in grinding operations. For feature extraction, two methods are employed with one based on time-series modeling and one based on discrete wavelet decomposition. For feature selection, three methods are used including the sequential forward floating selecting method and two newly proposed methods based on ant colony optimization (ACO). More details about the feature extraction and feature selection methods are briefly described in Sections 2 and 3, respectively. Section 4 describes the data and gives the test results on the identification of grinding wheel condition to show the effectiveness of the proposed methods, followed by the discussion. Section 6 concludes the paper and identifies topics for future study.

## 2. Grinding experiments, AE signals, and feature extraction

Acoustic emission signals collected at 1 MHz during grinding two ceramic materials (alumina and silicon nitride) with a resin-bonded diamond wheel are used in this study. To avoid redundancy, experimental details as described in Liao et al. (2006, 2007) are omitted here.

Two methods are used to extract features from the AE signals: one using discrete wavelet decomposition and another using autoregressive modeling. For the former, the multilevel 1-D wavelet decomposition function, *wavedec*, available in Matlab is chosen by specifying db1 Daubechies wavelets (which is the same as the Haar wavelets) with  $N=12$ . The energy of each decomposed component is then computed. The energy of a signal is the sum of the squares of its coefficient values. For an  $N$ -level decomposition, the number of extracted features is  $N+1$ . Therefore, a total of 13 features are obtained for each feature vector.

Autoregressive modeling essentially involves with building a regression model of auto-correlated errors existed in the AE signals. The model has the following form:

$$y_t = c + \beta t + v_t \quad (1)$$

$$v_t = \phi_1 v_{t-1} + \phi_2 v_{t-2} + \dots + \phi_p v_{t-p} + \varepsilon_t$$

Eq. (1) can be rewritten as

$$y_t = c + \beta t + (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)^{-1} \varepsilon_t \quad (2)$$

where  $y_t, t=1, 2, \dots, n$ , is a AE signal segment and  $\varepsilon_t \sim WN(0, \sigma^2)$ .

One major issue in building a stochastic AR model is to determine the appropriate model order. To this end, we utilize three autocorrelation plots (autocorrelation, partial autocorrelation, and inverse autocorrelation) and two well-known criteria: Akaike information criterion (AIC) and Durbin–Watson statistic. After analyzing many signal segments for different wheel conditions, the results indicate that 20 is an acceptable AR order.

The Yule–Walker method is used to estimate the coefficients based on the chosen model order. For an AR model,  $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \dots + \phi_p y_{t-p} + \varepsilon_t$ , where  $\phi_1, \phi_2, \dots, \phi_p$  are the coefficients of the model, and  $\varepsilon_t$  is stochastic shock. It can be shown that the autocorrelation function  $\rho(\tau), \tau=1, \dots, p$ , is related to the autoregressive parameters  $\phi_i, i=1, \dots, p$ , through the Yule–Walker equation for the autoregressive process:

$$\begin{aligned} \rho(1) &= \phi_1 \rho(0) + \phi_2 \rho(1) + \phi_3 \rho(3) + \dots + \phi_p \rho(p) \\ \rho(2) &= \phi_1 \rho(1) + \phi_2 \rho(0) + \phi_3 \rho(2) + \dots + \phi_p \rho(p-1) \\ &\dots \\ \rho(p) &= \phi_1 \rho(p) + \phi_2 \rho(p-1) + \phi_3 \rho(p-2) + \dots + \phi_p \rho(0) \end{aligned} \quad (3)$$

The autocorrelation  $\rho(\tau)$  is estimated by

$$\hat{\rho}(\tau) = \frac{\sum_{t=\tau+1}^N [(y_t - \bar{y})(y_{t-\tau} - \bar{y})]}{\sum_{i=1}^N (y_i - \bar{y})^2},$$

where  $N$  is the size of the sample. Plugging  $\hat{\rho}(\tau)$  into Eq. (3), the equation is solved to obtain the estimations of AR model coefficients. The model coefficients are then used as the features for subsequent classification. Only the model coefficients are then used as the features for subsequent classification; the intercept and the slope are not used.

## 3. Feature selection

Three feature selection methods, including two ACO-based methods and sequential forward floating selection (SFFS), are used to remove the irrelevant and redundant features, potentially exist in the two different feature vectors extracted by the two methods described in the previous section. Details of the ACO-based methods and a brief summary of the SFFS method are given in Sections 3.1 and 3.2, respectively, below.

### 3.1. Ant colony optimization-based feature selection methods

The first use of ACO for feature selection seems to be reported in (Yan and Yuan, 2004). In that work, Principal Component Analysis (PCA) was used to extract eigenfaces from images at the preprocessing stage; and then ACO was applied to select the optimal subset features using cross-validation with support vector machine as the classifier. Shen et al. (2005) modified ACO to select variables in quantitative structure–activity relationship (QSAR) modeling and to predict inhibiting action of some diarylimidazole derivatives on cyclooxygenase enzyme. In another work, ant colony optimization was tailored to perform feature selection for the purpose of prostate tissue characterization with support vector machine as the classifier, based on Trans-rectal Ultrasound images (Mohamed et al., 2005). It was shown that ACO outperformed genetic algorithms in obtaining a better feature subset that leads to higher classification accuracy. ACO was modified by substituting the mutual information for random as some problem-specific local heuristics to select feature subsets for modeling of temperature data with support vector machine (Zhang and Hu, 2005). A two-step ant colony system for feature selection was developed to split the heuristic search into two stages, in which candidate feature subsets are generated in the first stage and then some of them are randomly selected as the initial state for the ants in the second stage (Bello et al. (2006)). A modified discrete binary ant colony optimization algorithm was shown to greatly improve the fault diagnosis performance of support vector machine for the Tennessee Eastman process (Wang and Yu (2006)). In a most recent study, the performance of backpropagation feed-forward networks in medical diagnosis was shown to greatly improve when it was used together with ant colony optimization for feature selection (Sivagaminathan and Ramakrishnan, 2007). Most of the ACO-based feature selection methods reviewed above implements the random search strategy with feature subsets generation guided by ant colony optimization. The only exception is that of Sivagaminathan and Ramakrishnan, in which a strategy similar to the sequential forward selection is followed. The above review indicates that so far none has used ant colony optimization for feature selection in the domain of condition monitoring.

This section describes the two proposed ACO-based feature selection methods with one following the sequential search strategy and the other the random search strategy. The two methods are designed to include some basic traits commonly seen in nearly all ACO meta-heuristics as well as some unique to the specific task—feature selection. These traits are elaborated in Section 3.1.1. The algorithm of each method is given in Section 3.1.2. The wrapper approaches for performance evaluation are described in Section 3.3.

### 3.1.1. Traits

(1) *It employs a colony of artificial ants that work cooperatively to find good solutions.*

Each artificial ant is a simple agent that possesses the path-finding behavior of a real ant observed in real ant colonies. Although each artificial ant can build a feasible solution, high quality solutions are the result of the cooperation among the individuals of the whole colony. In the context of feature selection, each artificial ant is assigned the task to find feasible feature subsets and the whole colony work together to find the optimal feature subsets. The size of colony, or the number of artificial ants, is a parameter needed to be specified, which is similar to the population size, or the numbers of chromosomes, in a genetic algorithm.

(2) *Each artificial ant builds its solution by applying a stochastic local search policy.*

The stochastic local search policy is guided primarily both by ants' private information, which contains the memory of the ants' past actions, and by publicly available pheromone trails and a priori problem-specific local information. In the context of feature selection, ants' private information contains the features that have been selected in building a feature subset.

The stochastic local search policy used in the sequential search strategy follows that used by Sivagaminathan and Ramakrishnan (called the state transition rule in their paper), which is similar to the pseudo-random-proportional rule used in the traveling salesman problem (Dorigo et al. (1999)). Specifically an ant chooses a feature  $u^*$  from the unselected feature set  $U$  at iteration  $t$  according to the following probability:

$$p_{u^*}(t) = \begin{cases} 1 & \text{if } q < q_0 \text{ \& } u^* = \operatorname{argmax}\{\tau_u(t) \times \eta_u(t)^\beta\} \\ 0 & \text{if } q < q_0 \text{ \& } u^* \neq \operatorname{argmax}\{\tau_u(t) \times \eta_u(t)^\beta\} \\ \tau_u(t) \times \eta_u(t)^\beta / \sum_{u \in U} [\tau_u(t) \times \eta_u(t)^\beta] & \text{if } q \geq q_0 \end{cases} \quad (3)$$

In the above equation,  $q$  is a random variable uniformly distributed over  $[0, 1]$ ;  $q_0$  is a tunable parameter called exploitation probability factor, which determines the relative importance of exploitation (in the case that  $q < q_0$ ) versus exploration (in the case that  $q \geq q_0$ );  $\tau_u$  is the pheromone level of feature  $u$ ;  $\eta_u$  represents the inverse of the cost parameter; and  $\beta$  is a parameter that determines the relative importance of pheromone versus heuristic. The cost parameter can be used to account for the feature cost; setting them all equal if there is no distinction in the feature costs. Setting  $\beta$  to zero means that all features are given equal priority irrespective of their costs.

The stochastic local search policy used in the random search strategy is modified from Eq. (3). Specifically an ant builds a feature subset by choosing each feature  $u$  from the entire feature set at iteration  $t$  according to the following probability:

$$p_u(t) = \begin{cases} 1 & \text{if } q_1 < q_0 \text{ \& } \{\tau_u(t) \times \eta_u(t)^\beta\} / \sum_{u \in U} \{\tau_u(t) \times \eta_u(t)^\beta\} > q_2 \text{ or if } q_1 \geq q_0 \text{ \& } q_3 > 0.5 \\ 0 & \text{if } q_1 < q_0 \text{ \& } \{\tau_u(t) \times \eta_u(t)^\beta\} / \sum_{u \in U} \{\tau_u(t) \times \eta_u(t)^\beta\} \leq q_2 \text{ or if } q_1 \geq q_0 \text{ \& } q_3 \leq 0.5 \end{cases} \quad (3')$$

In the above equation,  $q_1$ ,  $q_2$ , and  $q_3$  are all random variables uniformly distributed over  $[0, 1]$ . All other notations are identical to those defined in Eq. (3).

(3) *To cooperate, artificial ants communicate by stigmergy similar to real ants via depositing pheromone on the ground while walking.*

The shorter the path between the food source and the nest, the sooner the pheromone is deposited by the real ants; which in turn leads to more ants taking the shorter path. In the meanwhile the pheromone deposited on the longer path receives few or no new deposits and gets evaporated over time. Artificial ants deposit an amount of pheromone that is a function of the quality of the solution found. Unlike real ants, the timing for updating pheromone trails is often problem dependent. Our methods update pheromone trails only after a solution is generated.

Our pheromone updating policy modifies from that used by Sivagaminathan and Ramakrishnan. For this policy the elitist strategy is adopted; i.e., only the best ant is allowed to update pheromone trails. For any feature,  $u$ , in the best feature subset found in iteration  $t$  denoted by  $X_0(t)$ , its pheromone level,  $\tau_u$ , is incremented by applying the global updating rule:

$$\tau_u(t+1) = (1 - e)\tau_u(t) + e/(\min E(t) + \varepsilon), \quad \forall u \in X_0(t). \quad (4)$$

In the above equation,  $e$  denotes the pheromone evaporation parameter;  $\min E(t)$  is the classification error associated with the best feature subset found by all ants in iteration  $t$ ; and  $\varepsilon$  is a small number added to prevent dividing by zero when  $\min E(t)$  equals to zero ( $\varepsilon = 0.1/\text{number of data records in this study}$ ). For any feature not in the best feature subset, its pheromone level,  $\tau_u$ , is updated by applying the following local updating rule:

$$\tau_u(t+1) = \begin{cases} (1 - e)\tau_u(t) & \text{if } \tau_u(t) = \tau_u(0), \quad \forall u \notin X_0(t) \\ (1 - e)\tau_u(t) + e\tau_u(0) & \text{if } \tau_u(t) \neq \tau_u(0), \quad \exists u \notin X_0(t) \end{cases} \quad (4')$$

### 3.1.2. Algorithms

The two ACO-based feature selection methods are named ACO-S and ACO-R with S and R denote sequential forward search and random search, respectively.

3.1.2.1. ACO-S. The ACO-S algorithm has the following steps:

#### ACO-S Algorithm

- (1) Load the data and normalize it by feature to remove the magnitude effect
- (2) Set parameters including maximum number of runs, max  $R$ , maximum size of feature subsets, max  $S$ , maximum number of ants, max  $A$ , maximum number of iterations, max  $T$ , exploitation probability factor,  $q_0$ , pheromone evaporation parameter,  $e$ , feature cost ( $=1$  for all features in this study), and the relative importance parameter between pheromone and heuristic,  $\beta$  ( $=0$  in this study).
- (3) For  $r=1$ : max  $R$ ,

(3.1) Initialize the pheromone level associated with each feature ( $=1$  in this study).

(3.2) For  $s=1$ : max  $S$ ,

(3.2.1) For  $t=1$ : max  $T$ ,

(3.2.1.1) Initialize the feature subset found by each ant to be null and its value, in terms of classification error, to be a large number.

(3.2.1.2) For  $a=1$ : max  $A$ ,



(3.2.1.2.1) While the feature subset size,  $s$ , is not yet reached  
Select a feature from the unselected list according to Eq. (3).  
End while

(3.2.1.2.2) Evaluate the classification error of the feature subset found by each ant with a selected classifier.  
End for

(3.2.1.3) Update the pheromone level of each feature appeared in the best feature subset found by all ants, according to Eq. (4). In the meantime, let the pheromone level of each feature not appeared in the best feature subset be changed according to Eq. (4').

(3.2.1.4) Record the best feature subset in each iteration, and other interesting information such as the pheromone update profiles associated with each feature.  
End for

End for

(3.3) Record the best feature subset in each run.  
End for

(4) Write output to the file.

(3.2.3) Update the pheromone level of each feature appeared in the best feature subset found by all ants, according to Eq. (4). In the meantime, let the pheromone level of each feature not appeared in the best feature subset be changed according to Eq. (4').

(3.2.4) Record the best feature subset in each iteration, and other interesting information such as the pheromone update profiles associated with each feature.  
End for

End for

(3.3) Record the best feature subset in each run. End for

(4) Write output to the file.

### 3.2. Sequential forward floating selection

The sequential forward floating selection (SFFS) algorithm was proposed by Pudil et al. (1994), as an improvement of the sequential forward selection (SFS) method. The SFFS is basically a bottom up search procedure which includes new features by means of applying the SFS procedure starting from the current feature set, followed by a series of successive conditional exclusion of the worst feature in the newly updated set provided that further improvement can be made to the previous sets. The details of the SFFS algorithm are omitted here to save space.

### 3.3. Wrapper approaches for feature subset evaluation

The wrapper approach is employed to evaluate the goodness of a feature subset found by a feature selection method. To compare their performances, five learning models are alternatively used. The performance is measured in terms of the average test error of stratified 10-fold cross-validations. The five learning models are the nearest mean (NM),  $k$ -nearest neighbor (KNN), fuzzy  $k$ -nearest neighbor (FKNN) (Keller et al. (1985)), center-based nearest neighbor (CBNN) (Gao and Wang (2007)), and  $k$ -means-based nearest prototype (KMNP). Most of these classifiers are well-known, except the last two. The CBNN classifier finds the centroid from a given class and defines a center-based line (CL) that connects a training example with its class centroid. In the classification phase, an unseen test datum  $\mathbf{x}$  is assigned a class label of the training example used to define the CL that is closest to  $\mathbf{x}$ . For KMNP, the number of prototypes for each class is set equal to the size of the smallest class;  $k$ -means clustering is then used on each class to find the prototypes, and finally KNN is applied for classification. More details can be found in the original references.

## 4. Test results

All three feature selection methods and five classification algorithms were coded in Matlab and executed using a Dell Latitude C640 laptop computer with Mobile Intel 2 GHz Pentium 4-M CPU. A total of 320 records each of both wavelet energy

### 3.1.2.2. ACO-R. The ACO-R algorithm has the following steps:

#### ACO-R Algorithm

(1) Load the data and normalize it by feature to remove the magnitude effect

(2) Set parameters including maximum number of runs, max  $R$ , maximum size of feature subsets, max  $S$ , maximum number of ants, max  $A$ , maximum number of iterations, max  $T$ , exploitation probability factor,  $q_0$ , pheromone evaporation parameter,  $e$ , feature cost (=1 for all features in this study), and the relative importance parameter between pheromone and heuristic,  $\beta$  (=0 in this study).

(3) For  $r=1$ : max  $R$ ,

(3.1) Initialize the pheromone level associated with each feature (=1 in this study).

(3.2) For  $t=1$ : max  $T$ ,

(3.2.1) Initialize the feature subset found by each ant to be null and its value, in terms of classification error, to be a large number.

(3.2.2) For  $a=1$ : max  $A$ ,

(3.2.2.1) While the feature subset is null  
Build a feature subset from the entire feature set according to Eq. (3').  
End while

(3.2.2.2) Evaluate the classification error of the feature subset found by each ant with a selected classifier.  
End for

End for

**Table 2**

Summary of classification error rates for the wavelet energy levels data.

	NM	1NN	F1NN	CBNN	KMNP
None (%)	34.69	13.13	13.13	11.25	13.13
SFFS	23.44	10	10	7.81	10
ACO-S (10,10)	23.44 $\pm$ 0.0	10 $\pm$ 0	10.25 $\pm$ 0.34 (best=10)	8.13 $\pm$ 0.22 (best=7.81)	10.25 $\pm$ 0.34 (best=10)
ACO-R (50,50)	24.69 $\pm$ 0	10.13 $\pm$ 0.17 (best=10)	10.19 $\pm$ 0.28 (best=10)	8.13 $\pm$ 0.22 (best=7.81)	10.25 $\pm$ 0.26 (best=10)

features and AR coefficient features were obtained from the AE signals collected under different grinding conditions and two tool conditions (sharp vs. dull). The test results were obtained with 10 ants and 10 iterations for the ACO-S method and 50 ants and 50 iterations for the ACO-R for both datasets. In the table, they are denoted as ACO-S (10, 10) and ACO-R (50, 50), respectively. Due to its stochastic nature, five runs were made for each ACO-based method by setting both  $q_0$  and  $e$  to 0.5 arbitrarily. The effects of other parameter values are discussed in Section 5. Both the classification accuracy and the CPU time were recorded and serve as the two primary measures of performance. The classifier errors are given as “average  $\pm$  standard deviation” in the table, with the best performance noted.

#### 4.1. Wavelet energy features

Tables 2–4 summarize the classification error rates, CPU times, and locally best feature subsets found for the 13-dimensional wavelet energy features, respectively. In Table 4, the fraction after a best feature subset indicates the number of runs that it was found over a total of five runs. Based on these results, the following observations can be made:

- (1) Feature selection improves the classification accuracy, regardless the feature selection method and classifier used.
- (2) In terms of classification accuracy, the CBNN classifier is the best among all five tested regardless of whether feature selection is applied or not, or which feature selection method it is when one is used.
- (3) The lowest classification error of 7.81% is achieved by all three feature selection methods. In terms of CPU time, the SFFS method requires less time than the ACO-based methods.
- (4) The associated best feature subset found by all three feature selection methods contains seven features, i.e., 2, 3, 7, 8, 10, 11, and 13. The reduction ratio is 46%.
- (5) Comparing the two ACO-based feature selection methods, ACO-S seems to be better than ACO-R because the best results are obtained in more runs in shorter CPU time.
- (6) The three nearest neighbor classifiers attain the same classification error and they outperform the nearest mean classifier.

**Table 3**  
Summary of CPU times (in seconds) for the wavelet energy levels data.

	NM	1NN	F1NN	CBNN	KMNP
SFFS	14	491	381	3200	671
ACO-S (10, 10)	26	1453	380	5710	1011
ACO-R (50, 50)	161	5993	3931	24,704	6032

**Table 4**  
Summary of best feature subsets found for the wavelet energy levels data.

	NM	1NN	F1NN	CBNN	KMNP
SFFS	{10}	{1–4, 6, 7, 9–13}	{1–4, 6, 7, 9–13}	{2, 3, 7, 8, 10, 11, 13}	{1–4, 6, 7, 9–13}
ACO-S (10, 10)	{10} 5/5	{2, 3, 10–13} 2/5  {1–4, 7, 9–13} 1/5 {1–4, 7, 8, 10–13} 1/5 {2–4, 6, 7, 9, 10, 11, 13} 1/5	{2, 3, 10–13} 1/5  {1–4, 7, 9–13} 1/5 {1–4, 7, 8, 10–13} 1/5	{2, 3, 7, 8, 10, 11, 13} 1/5	{2, 3, 10–13} 2/5  {1–4, 7, 8, 10–13} 1/5
ACO-R (50,50)	–	{2, 3, 10–13} 2/5 {2–4, 6, 7, 9–11, 13} 1/5	{1–4, 7, 9, 10–13} 1/5 {1–4, 7, 8, 10–13} 1/5 {2–4, 6, 7, 9–11, 13} 1/5	{2, 3, 7, 8, 10, 11, 13} 1/5	{1–4, 7, 8, 10–13} 1/5 {1–4, 7, 9, 10–13} 1/5

Figs. 1 and 2 are plotted to show how the pheromone levels of features change with the search process for the best run of each ACO-based feature selection method when CBNN is used as the classifier with both  $q_0$  and  $e$  set at 0.5. Theoretically, the maximal number of updates is determined by both the number of iterations and the number of features for the ACO-S method ( $130 = 10 \times 13$  in Fig. 1), and by the number of iterations specified for the ACO-R method (50 in Fig. 2). Obviously these profiles are determined by the updating rule given in Eqs. (4) and (4'). Note that Fig. 1 has far fewer numbers of updates, which indicate that in many cases no feature subsets were found.

From these profiles, it can be known which features are selected during different stages of the search process. For example, in Fig. 1 the first feature selected to increase its pheromone level in the very beginning is  $\times 3$ , then  $\times 11$ , and so on. It is interesting to point out that for ACO-S all features in the best feature subset have the same highest pheromone level, i.e., 12.16, at the end. For ACO-R, however, the same cannot be observed.

#### 4.2. AR coefficient features

Tables 5–7 summarize the classification error rates, CPU times, and locally best feature subsets found for the 20-dimensional AR coefficient features, respectively. Based on these results, the following observations can be made:

- (1) Feature selection improves the classification accuracy, regardless the feature selection method and classifier used.
- (2) The lowest classification error is 6.875% for this dataset, which was achieved by 5 different combinations of feature selection method and classifier. They include SFFS with 1NN, CBNN, and KMNP, and ACO-S with 1NN and KMNP.
- (3) The best feature subsets associated with the lowest classification error found by a different combination of feature selection method and classifier might differ. The smallest feature subset is  $\{ \times 1, \times 3, \times 4, \times 8, \times 9, \times 10 \}$ , which is found by ACO-S with 1NN and ACO-S with KMNP. The largest reduction ratio is thus 70%, reducing from 20 features to 6.
- (4) If CPU time is taken into consideration, then the best combination seems to be ACO-S with 1NN because it requires the least computational time to obtain the same best result. As noted in 3), this particular combination also found the smallest best feature subset.
- (5) The ACO-R feature selection method did not produce any best results for this dataset and it already took longer computational time than the other two feature selection methods. An experiment with increased number of ants, max  $A$ , and number of iterations, max  $T$ , with the nearest mean classifier still did not produce the best result, though the average classification error is reduced (see note 1 in Tables 4 and 5).

Figs. 3 and 4 are plotted to show how the pheromone levels of features change with the search process for the best run of ACO-S feature selection method when 1NN and KMNP are used as the classifier, respectively, with both  $q_0$  and  $e$  set at 0.5. Theoretically, the maximal number of updates is determined by both the number of iterations and the number of features for the ACO-S method ( $200=10 \times 20$ ). Obviously these profiles are determined by the updating rule given in Eqs. (4) and (4'). Note that both Figs. 3 and 4 have fewer numbers of updates, which indicate that in many cases no feature subsets were found, thus no update was done.

From these profiles, it can be known which features are selected during different stages of the search process. For example,

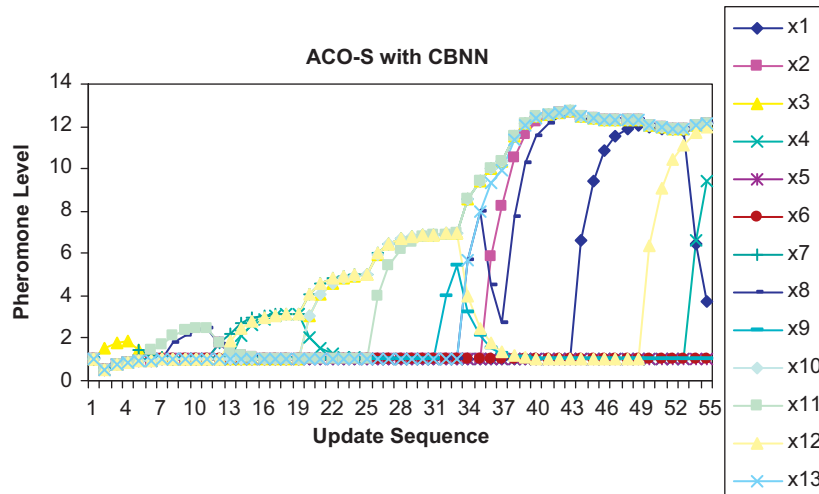
in Fig. 3 the first feature selected to increase its pheromone level in the very beginning is  $x_9$ , then  $x_{12}$ , and so on. It is interesting to point out that for both ACO-S with 1NN and ACO-S with KMNP all

**Table 6**

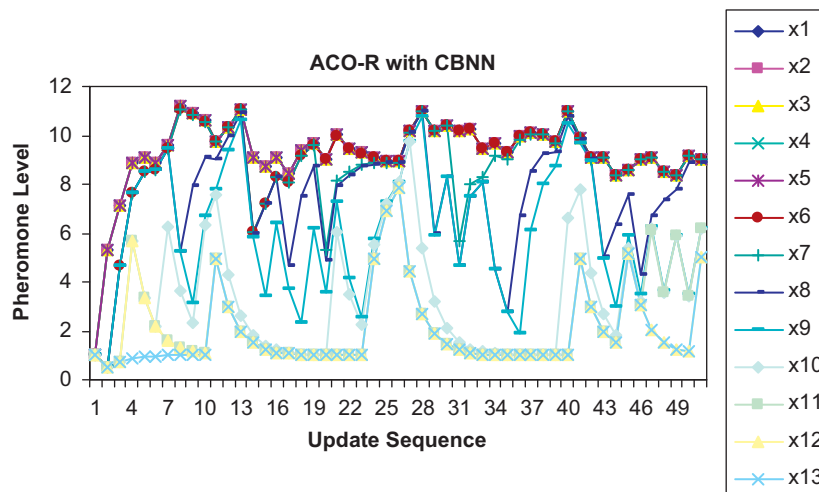
Summary of CPU times (in seconds) for the AR model coefficients data.

	NM	1NN	F1NN	CBNN	KMNP
SFFS	29.5	1517.7	1146.9	4968.3	1908.7
ACO-S (10, 10)	40.8	542.3	708.0	2298.8	843.7
ACO-R (50, 50)	147.0 <sup>a</sup>	6242.1	4180.9	23,189.3	5982.4

<sup>a</sup> When max  $A$  and max  $T$  were increased to 100, the CPU time becomes 770.7.



**Fig. 1.** Pheromone level change profile of an ACO-S run with CBNN as the classifier for the dataset of wavelet energy features.



**Fig. 2.** Pheromone level change profile of an ACO-R run with CBNN as the classifier for the dataset of wavelet energy features.

**Table 5**

Summary of classification error rates for the AR model coefficients data.

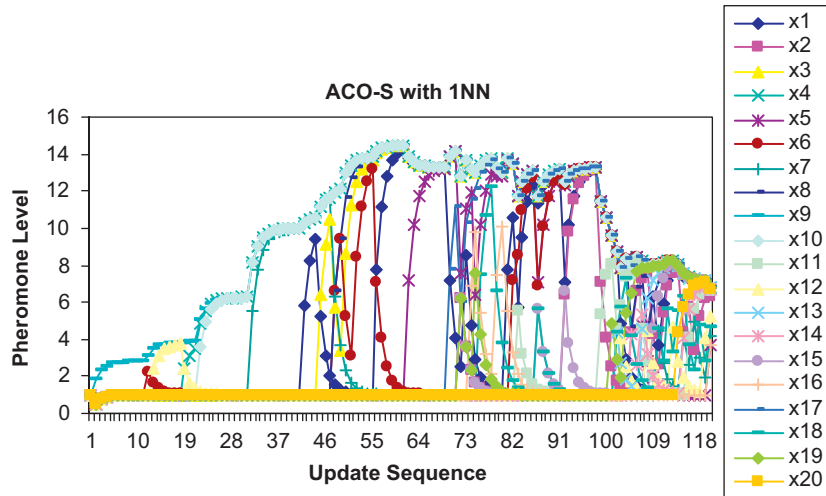
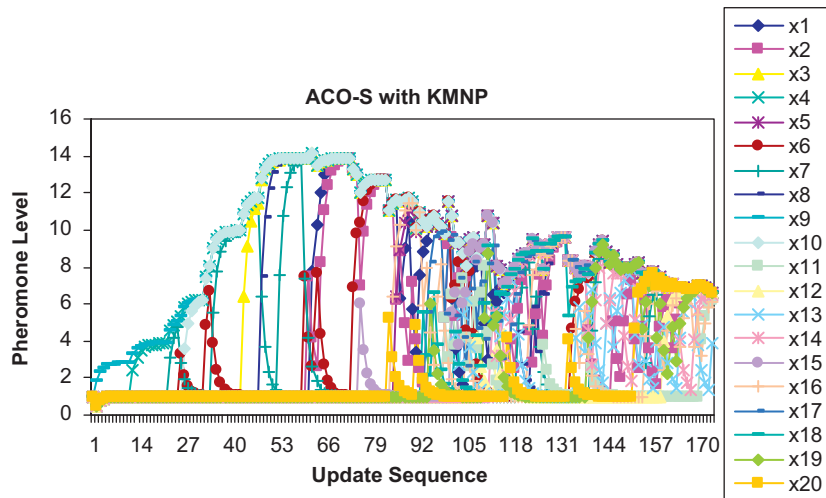
	NM	1NN	F1NN	CBNN	KMNP
No selection (%)	39.375	15.625	19.375	14.0625	15.625
SFFS	21.25	6.875	10.625	6.875	6.875
ACO-S	21.5 $\pm$ 0.51	9 $\pm$ 1.69	11.25 $\pm$ 0.44	9.38 $\pm$ 2.01	7.56 $\pm$ 0.81
(10, 10)	(best=20.625)	(best=6.875)	(best=10.625)	(best=7.19)	(best=6.875)
ACO-R	25.38 $\pm$ 1.6 <sup>a</sup>	8.38 $\pm$ 0.87	12.06 $\pm$ 0.93	8.44 $\pm$ 0.31	8.5 $\pm$ 0.78
(50, 50)	(best=22.81)	(best=7.19)	(best=10.625)	(best=8.125)	(best=7.19)

<sup>a</sup> When max  $A$  and max  $T$  were increased to 100, the error rates become 23.88  $\pm$  0.47 with the best=23.13.

**Table 7**

Summary of best feature subsets found for the AR model coefficients data.

	NM	1NN	F1NN	CBNN	KMNP
SFFS	{8, 9, 10, 12, 18}	{1, 3, 4, 5, 7, 9, 10, 17}	{1, 3, 4, 8, 9, 10}	{1, 3, 4, 7–10, 14, 15, 17–19}	{1, 3, 4, 5, 7, 9, 10, 17}
ACO-S (10, 10)	{9, 12, 13, 15, 19} <sup>a</sup> 1/5	{1, 3, 4, 8, 9, 10} 1/5	{1, 3, 4, 8, 9, 10} 1/5	–	{1, 3, 4, 8, 9, 10} 1/5
ACO-R (50, 50)	–	–	{2, 3, 4, 7, 8, 9, 10} 1/5	–	{1, 3, 4, 5, 7, 9, 10, 17} 1/5

<sup>a</sup> This feature subset has slightly lower error than that found by SFFS, as indicated in Table 5.**Fig. 3.** Pheromone level change profile of an ACO-S run with 1NN as the classifier for the dataset of AR coefficients.**Fig. 4.** Pheromone level change profile of an ACO-S run with KMNP as the classifier for the dataset of AR coefficients.**Table 8**

Summary of classification error rates for the iris dataset.

	NM	1NN	F1NN	CBNN	KMNP
No selection (%)	7.33	4.67	6	8	4.67
Exhaustive (%)	4	3.33	6	7.33	3.33
SFFS (%)	4	3.33	6	7.33	3.33
ACO-S (2, 2) (%)	4 ± 0	3.87 ± 0.73 (best=3.33)	6 ± 0 (best=6)	7.33 ± 0 (best=7.33)	3.87 ± 0.73% (best=3.33)
ACO-R (5, 5) (%)	4 ± 0 (best=4)	3.87 ± 0.73 (best=3.33)	6 ± 0 (best=6)	7.33 ± 0 (best=7.33)	3.33 ± 0%



features in the best feature subset have relatively high pheromone level at the end. However, there are features with higher pheromone level not part of the best feature subset.

## 5. Discussion

To reaffirm the validity of the test results presented in Section 4, the iris and wine datasets taken from the UCI Repository were also tested with the same test procedure and methods. The reason for choosing the iris dataset is its small size so that the exhaustive method can also be applied to identify the globally optimal feature subset.

Tables 8–10 summarize the stratified 10-fold cross-validation classification error rate, the CPU time, and the best feature subsets found for each combination of feature selection method and classifier, respectively. The above results were obtained with two ants and two iterations for the ACO-S method, regardless of the classifier used, and with five ants and five iterations for the ACO-R. Similar to previous tests, five runs were made for each ACO-based method by setting both  $q_0$  and  $e$  to 0.5 arbitrarily.

The results indicate that:

- (1) Feature selection improves the classification accuracy, regardless the combination of feature selection method and classifier used.
- (2) The globally optimal feature subset is {4}, {3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, and {3, 4} with the lowest classification error of 4%, 3.33%, 6%, 7.33%, and 3.33% for the NM, 1NN, F1NN, CBNN, and KMNP classifier, respectively.
- (3) All three feature selection methods can achieve the best performance, regardless the classifier used. The globally optimal feature subset was found by all feature selection and classifier combinations except one, i.e., ACO-R with NM, which found the best feature subset larger than the globally optimal feature subset.

**Table 9**  
Summary of CPU times (in seconds) for the iris dataset.

	NM	1NN	F1NN	CBNN	KMNP
Exhaustive	0.75	5.33	3.52	19.87	5.32
SFFS	0.85	6.17	4.26	38.06	6.16
ACO-S (2, 2)	0.86	5.79	3.96	20.74	5.84
ACO-R (5, 5)	1.21	8.91	6.51	34.35	8.96

**Table 10**  
Summary of best feature subsets found for the iris dataset.

	NM	1NN	F1NN	CBNN	KMNP
Exhaustive	{4}	{3, 4}	{1, 2, 3, 4}	{1, 2, 3, 4}	{3, 4}
SFFS	{4}	{3, 4}	{1, 2, 3, 4}	{1, 2, 3, 4}	{3, 4}
ACO-S (2, 2)	{3, 4} 4/5 {4} 1/5	{3, 4} 3/5	{1, 2, 3, 4}	{1, 2, 3, 4}	{3, 4} 3/5
ACO-R (5, 5)	{3, 4} 4/5 {2, 3, 4} 1/5	{3, 4} 3/5	{1, 2, 3, 4}	{1, 2, 3, 4}	{3, 4}

**Table 11**  
Summary of classification error rates for the wine data.

	NM	1NN	F1NN	CBNN
None (%)	2.78	4.17	4.17	2.78
SFFS	1.39	0.69	1.39	0.69
ACO-S (10,10)	1.94 ± 0.31 (best=1.39)	1.25 ± 0.58 (best=0.69)	1.11 ± 0.38 (best=0.69)	1.53 ± 0.58 (best=0.69)
ACO-R (30,30)	1.67 ± 0.21 (best=1.39)	1.11 ± 0.38 (best=0.69)	1.11 ± 0.38 (best=0.69)	1.94 ± 0.31 (best=1.39)

- (4) For a dataset with small number of features like the iris dataset, it makes no sense to use any feature selection method at all. For such a dataset, feature selection methods took even longer CPU time than the exhaustive method.

Table 11 summarizes the classification accuracies for the wine dataset. Again, the results indicate that feature selection improves the classification accuracy, regardless the combination of feature selection method and classifier used. The above test results have clearly shown that both the ACO-based feature selection methods and the SFFS method are capable of finding the best feature subsets.

Next, the effects of two more parameters, specifically the exploitation probability factor,  $q_0$ , and the pheromone evaporation parameter,  $e$ , are investigated. With all other parameters fixed, these two parameters are varied at three levels each at 0.1, 0.5, and 0.9. Hence, a total of nine combinations are experimented including the one that has been used in the previous runs, i.e., the 0.5–0.5 combination. The 1NN classifier is chosen over the other classifiers for this investigation for the following reasons: (1) the CPU time required is not too long; and (2) it is the best classifier for the weld flaw identification dataset.

Table 12 summarizes the effects of the ACO-S parameters on the classification error, CPU time, and best feature subsets found for the dataset of wavelet energy features with max  $A$  and max  $T$  set at 10. The results indicate that all nine combinations of  $q_0$  and  $e$  can produce the lowest classification error of 10%. Using a large value of  $q_0$  and  $e$  takes less CPU time; however, the number of runs producing the best result also reduces. On the other hand, using a small value of  $q_0$  and  $e$  takes longer CPU time; however, the number of runs producing the best result also increases. Table 13 summarizes the results for the dataset of AR coefficients with max  $A$  and max  $T$  set at 10 as well. Three out of nine combinations did not produce the lowest classification error of 6.875% and they are the (0.1, 0.9), (0.5, 0.9), and (0.9, 0.1) combinations. Generally, to obtain lower classification error lower value of  $q_0$  is better if the  $e$  value is not too high, at the cost of slightly longer CPU time.

Lastly, the effectiveness of using db3, instead of db1, for the wavelet-based feature extraction method was also tested. The results are summarized in Table 14. Comparing with Table 2, it can be seen that:

- (1) Without feature selection, the classification accuracies of db3 are all worse than those of db1, regardless the classifier used.
- (2) With feature selection, the best classification accuracy of db3 is 2.19% lower than that of db1 for the CBNN classifier and

**Table 12**

Effects of ACO-S parameters on the wavelet energy levels data with 1NN.

Method	$q_0$	$e$	Classification error (%)	CPU time (s)	Best Feature Subsets
ACO-S (10, 10)	<b>0.1</b>	<b>0.1</b>	<b><math>10.0 \pm 0</math></b>	3044	{2, 3, 10–13} 5/5
	<b>0.1</b>	<b>0.5</b>	<b><math>10.0 \pm 0</math></b>	3030	{2, 3, 10–13} 5/5
	<b>0.1</b>	<b>0.9</b>	<b><math>10.0 \pm 0</math></b>	2462	{2, 3, 10–13} 5/5
	<b>0.5</b>	<b>0.1</b>	<b><math>10.0 \pm 0</math></b>	2874	{2, 3, 10–13} 4/5 {1–4, 7, 9–13} 1/5
	<b>0.5</b>	<b>0.5</b>	<b><math>10.0 \pm 0</math></b>	1453	{2, 3, 10–13} 2/5 {1–4, 7, 9–13} 1/5 {1–4, 7, 8, 10–13} 1/5 {2–4, 6, 7, 9, 10, 11, 13} 1/5
	0.5	0.9	$10.23 \pm 2.28$ (best=10)	409	{2, 3, 10–13} 1/5 {1–4, 7, 8, 10–13} 3/5
	0.9	0.1	$10.25 \pm 0.34$ (best=10)	3384	{1–4, 7, 9–13} 2/5 {1–4, 7, 8, 10–13} 1/5
	0.9	0.5	$10.31 \pm 0.31$ (best=10)	589	{1–4, 7, 8, 10–13} 1/5 {1–4, 7, 9–13} 1/5
	0.9	0.9	$10.44 \pm 0.42$ (best=10)	390	{2, 3, 10–13} 1/5 {1–4, 7, 8, 10–13} 1/5

**Table 13**

Effects of ACO-S parameters on the AR model coefficients data with 1NN.

Method	$q_0$	$e$	Classification error (%)	CPU time (s)	Best feature subsets
ACO-S (10, 10)	0.1	0.1	$7 \pm 0.17$ (best=6.875)	5427	{1, 3, 4, 8, 9, 10} 2/5 {2, 3, 4, 7, 8, 9, 10} 1/5
	<b>0.1</b>	<b>0.5</b>	<b><math>6.875 \pm 0</math></b>	3812	{3, 4, 5, 8, 9, 10, 16, 17, 18, 19} 1/5 {1, 3, 4, 8, 9, 10} 1/5 {2, 3, 4, 7, 8, 9, 10} 3/5
	0.1	0.9	$8.75 \pm 0.13$ (best=7.19)	927	–
	<b>0.5</b>	<b>0.1</b>	<b><math>6.875 \pm 0</math></b>	5142	{1, 3, 4, 8, 9, 10} 4/5 {2, 3, 4, 7, 8, 9, 10} 1/5
	0.5	0.5	$9 \pm 1.69$ (best=6.875)	542	{1, 3, 4, 8, 9, 10} 1/5
	0.5	0.9	$8.63 \pm 1.36$ (best=7.19)	557	–
	0.9	0.1	$8.19 \pm 0.15$ (best=7.19)	5197	–
	0.9	0.5	$7.625 \pm 1.16$ (best=6.875)	1589	{2, 3, 4, 7, 8, 9, 10} 1/5
	0.9	0.9	$8.56 \pm 0.17$ (best=6.875)	491	{1, 3, 4, 8, 9, 10} 1/5 {3, 4, 5, 8, 9, 10, 17} 1/5

**Table 14**

Summary of classification error rates for the db3-wavelet energy levels data.

	NM	1NN	F1NN	CBNN	KMNP
None (%)	35	24.06	24.06	21.56	24.06
SFFS	23.13	9.38	9.38	10.00	9.38
ACO-S (10, 10)	$23.19 \pm 0.14$ (best=23.13)	$9.56 \pm 0.17$ (best=9.38)	$9.38 \pm 0$	$10.25 \pm 0.34$ (best=10)	$9.5 \pm 0.17$ (best=9.38)
ACO-R (50, 50)	$23.5 \pm 0.26$ (best=23.13)	$10.31 \pm 1.17$ (best=9.38)	$10.12 \pm 1.05$ (best=9.38)	$10.69 \pm 0.41$ (best=10)	$10.56 \pm 1.22$ (best=9.38)

slightly higher (< 1%) for other classifiers. Despite of these differences, feature selection results in higher accuracies than no selection at all.

## 6. Conclusions

This paper has presented the results obtained in our investigation of two important issues related to sensor-based tool health condition monitoring, i.e., feature extraction and feature selection. Two feature extraction methods, three feature selection methods, and five classifiers were employed in the study. Specifically, grinding wheel condition was monitored with acoustic emission signals. AE signals were extracted both by discrete wavelet decomposition and autoregressive modeling. The three feature selection methods include the sequential forward floating selec-

tion, and two ant colony optimization-based feature selection methods based on different search strategies. The testing method is stratified  $k$ -fold cross-validation. The performances were measured in terms of classification error rate and CPU time.

Based on the test results, the following conclusions can be drawn:

- (1) For both datasets of wavelet energy features and AR coefficients, feature selection improves the classification accuracy compared with using no feature selection at all.
- (2) For the dataset of wavelet energy features, center-based nearest neighbor produces the lowest classification error of 7.81% among all five classifiers tested, with features selected by either one of the three methods.
- (3) For the dataset of AR coefficients, five combinations of feature

selection and classification methods produce the lowest classification error of 6.875%. For this dataset, ACO-R seems to be outperformed by both the ACO-S method and the SFFS method.

- (4) For the two ACO-based methods, a proper selection of the parameters is necessary in order to produce the best result. Generally, to obtain lower classification error lower value of  $q_0$  is better if the  $e$  value is not too high. This, however, is achieved often at the cost of slightly longer CPU time.

The best combination of feature selection method and classifier seems to be data-dependent. It would be worthwhile to test with more datasets to figure out whether one particular combination works better than others in most datasets. It is also interesting to investigate why the ACO-R method works well for the dataset of wavelet energy features but not the dataset of AR coefficients. The major difference in the two datasets is in the number of features. But is that the major contributing factor? Lastly, a possible future research topic is to develop more and better ACO-based feature selection methods.

## References

- Bello, R., Puris, A., Nowe, A., Martínez, Y., García, M.M., 2006. Two step ant colony system to solve the feature selection problem. In: Martínez-Trinidad, J.F. et al. (Eds.), CIARP 2006, Lecture Notes in Computer Science, vol. 4225, pp. 588–596.
- Dorigo, M., Caro, G.D., Gambardella, L.M., 1999. Ant algorithms for discrete optimization. *Artificial Life* 5, 137–172.
- Furutani, K., Ohguro, N., Hieu, N.T., Nakamura, T., 2002. In-process measurement of topography change of grinding wheel by using hydrodynamic pressure. *International Journal of Machine Tools & Manufacture* 42, 1447–1453.
- Gao, Q.B., Wang, Z.Z., 2007. Center-based nearest neighbor classifier. *Pattern Recognition* 40, 346–349.
- Hosokawa, A., Mashimo, K., Yamada, K., Ueda, T., 2004. Evaluation of grinding wheel surface by means of grinding sound discrimination. *JSME International Journal, Series C* 47 (1), 52–58.
- Hwang, T.W., Whitenton, E.P., Hsu, N.N., Blessing, G.V., Evans, C.J., 2000. Acoustic emission monitoring of high speed grinding of silicon nitride. *Ultrasonics* 38, 614–619.
- Keller, J.M., Gray, M.R., Givens Jr., J.A., 1985. A fuzzy  $k$ -nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics* 15 (4), 580–585.
- Kwak, J.-S., Ha, M.-K., 2004. Detection of dressing time using the grinding force signal based on the discrete wavelet decomposition. *International Journal of Advanced Manufacturing Technology* 23, 87–92.
- Lezanski, P., 2001. An intelligent system for grinding wheel condition monitoring. *Journal of Materials Processing Technology* 109, 258–263.
- Liao, T.W., Hua, G., Qu, J., Blau, P.J., 2006. Grinding wheel condition monitoring with hidden Markov model-based clustering methods. *Machining Science and Technology* 10, 511–538.
- Liao, T.W., Ting, C.F., Qu, J., Blau, P.J., 2007. A wavelet based methodology for grinding wheel condition monitoring. *International Journal of Machine Tools & Manufacture* 47, 580–592.
- Mohamed, S.S., Youssef, A.M., El-Saadany, E.F., Salama, M.M.A., 2005. Artificial life feature selection techniques for prostate cancer diagnosis using TRUS images. In: Kamel, M., Campilho, A. (Eds.), *ICIAI 2005, Lecture Notes in Computer Science*, vol. 3656, pp. 903–913.
- Mokbel, A.A., Maksoud, T.M.A., 2000. Monitoring of the condition of diamond grinding wheels using acoustic emission technique. *Journal of Materials Processing Technology* 101, 292–297.
- Pudil, P., Novovičová, J., Kittler, J., 1994. Floating search methods in feature selection. *Pattern Recognition Letters* 15, 1119–1125.
- Shen, Q., Jiang, J.-H., Tao, J.-C., Shen, G.-L., Yu, R.-Q., 2005. Modified ant colony optimization algorithm for variable selection in QSAR modeling: QSAR studies of cyclooxygenase inhibitors. *Journal of Chemical Information and Modeling* 45, 1024–1029.
- Sivagaminathan, R.K., Ramakrishnan, S., 2007. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert Systems with Applications* 33, 49–60.
- Tonshoff, H.K., Friemuth, T., Becker, J.C., 2002. Process monitoring in grinding. *Annals of the CIRP* 51 (2), 551–571.
- Wang, L., Yu, J., 2006. A modified discrete binary ant colony optimization and its application in chemical process fault diagnosis. In: Wang, T.-D. et al. (Eds.), *SEAL 2006, Lecture Notes in Computer Science*, vol. 4247, pp. 889–896.
- Warkentin, A., Bauer, R., 2003. Analysis of wheel wear using force data in surface grinding. *Transactions of the ASME/DE la SCGM* 27 (3), 193–204.
- Yan, Z., Yuan, C., 2004. Ant colony optimization for feature selection in face recognition. In: Zhang, D., Jain, A.K. (Eds.), *ICBA 2004, Lecture Notes in Computer Science*, vol. 3072, pp. 221–226.
- Zhang, C., Hu, H., 2005. Ant colony optimization combining with mutual information for feature selection in support vector machines. In: Zhang, S., Jarvis, R. (Eds.), *AI 2005, Lecture Notes in Artificial Intelligence*, vol. 3809, pp. 918–921.