Expert Systems with Applications xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process

M. Correa [a,*], C. Bielza [b], J. Pamies-Teixeira [c]

[a] Instituto de Automática Industrial – Spanish National Research Council, Ctra. Campo Real km. 0.200, 28500 Arganda del Rey, Madrid, Spain
[b] Universidad Politécnica de Madrid, Departamento de Inteligencia Artificial, 28660 Boadilla del Monte, Madrid, Spain
[c] Universidade Nova de Lisboa, Faculdade de Ciencias e Tecnologia, Quinta da Torre 2829-516 Caparica, Portugal

ARTICLE INFO

Keywords:
Bayesian networks
Artificial neural networks
Surface roughness
High-speed milling
Supervised classification

ABSTRACT

Machine tool automation is an important aspect for manufacturing companies facing the growing demand of profitability and high quality products as a key for competitiveness. The purpose of supervising machining processes is to detect interferences that would have a negative effect on the process but mainly on the product quality and production time. In a manufacturing environment, the prediction of surface roughness is of significant importance to achieve this objective. This paper shows the efficacy of two different machine learning classification methods, Bayesian networks and artificial neural networks, for predicting surface roughness in high-speed machining. Experimental tests are conducted using the same data set collected in our own milling process for each classifier. Various measures of merit of the models and statistical tests demonstrate the superiority of Bayesian networks in this field. Bayesian networks are also easier to interpret that artificial neural networks.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Quality is defined as the extent to which a product conforms to the design specifications and how it complies with the requirements of component functionality. For some industries, such as automotive and aeronautical sectors, the quality of their parts is very important given the high requirements to which they are subject.

However, difficulties arise from the fact that a measure of quality can only be evaluated "out-of-process", resulting in losses because there is no alternative to removing defective parts from the production line. Therefore, it is necessary to incorporate machine learning methods that provide in-process solutions to predict quality from some measured variables.

Nowadays, many papers have been published about modelling the machining process and, more specifically, about the prediction of surface quality in machining processes. Researchers have approached the problem from different points of view and using different techniques. The most frequently used are artificial neural networks (ANNs) (Huang & Chen, 2003; Samson & Chen, 2003; Tsai, Chen, & Lou, 1999) and linear and multiple regression (Aboulatta & Mádl, 2001; Feng & Wang, 2003; Kirby, Zhang, & Chen, 2004). However, their models focus on very reduced environments and with limited experimentation.

In Correa, Bielza, Ramírez, and Alique (in press), we have recently proven the advantages of using Bayesian networks (BNs) as a successful solution for predicting surface quality in high-speed milling. As an important added value, the current research includes the influence of the geometry of the workpiece and the hardness of the material to be machined as key variables in the model construction aimed at a particular subdomain that contains a range of aluminium hardnesses used in automotive and aeronautical pieces. This is a landmark in this application domain, since it extends and generalizes the scope of the experimentation, which is no longer confined to a single test profile.

BN models were learnt from data. These data were collected in our laboratory using experimental design to guarantee statistical validity. Besides BNs, we constructed ANNs, known to be a strong competitor widely used in this field, to make a comparison and to demonstrate the superiority of BNs. As far as we know, there have been no comparisons of how well these two techniques solve this kind of problem.

Obviously, these two models have already been compared in other contexts like e.g. modelling manufacturing processes (Perzyk, Biernacki, & Kochański, 2005), discriminating plants, weeds and soil in color images (Marchant & Onyango, 2003), and modelling the response time of service-oriented systems (Zhang

* Corresponding author. Tel.: +34 91 8711900; fax: +34 91 8717050.
E-mail addresses: macorrea@iai.csic.es, macoval@gmail.com (M. Correa).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.09.024

& Bivens, 2007). The two have been used in a combined fashion (Antal, Fannes, Timmerman, Moreau, & de Moor, 2003).

Our paper aims to compare the two approaches (BNs and ANNs) in the context of a practical industrial problem, the prediction of *surface roughness* in high-speed milling. The proposed models are target the automotive and aeronautical industry using some typical geometric features and a number of aluminium alloys giving a wide range of hardness.

The remainder of this paper is structured as follows. Section 2 presents the difficulties for measuring quality in a high-speed milling process, how surface roughness is inspected and which techniques and data we have used for surface roughness monitoring, since this is a variable that is very difficult to measure in-process. Sections 3 and 4 summarise the main principles of BNs and ANNs, respectively, and introduce the models on which our comparison is based. Section 5 focuses on the quantitative comparison of the two models and, in particular, on the knowledge engineering aspects of both. Finally, Section 6 concludes with our most important findings.

2. Quality in a machining process

The quality of a component is often associated with its surface aspect and appearance. However, this can be misleading in the sense that the surface may contain features that are not reflected in its appearance. Beyond aspect, a machined surface may contain imperfections and deviations from the nominal expected surface that compromise the quality of the component. Hence, it is necessary to clearly define a measure of surface roughness.

2.1. Surface roughness

There are many parameters for characterizing surface roughness, but the most used measure in industry is roughness average, *Ra*, representing the arithmetic mean of the absolute ordinate values $f(x)$ within a sampling length (L), see the following equation:

$$Ra = \frac{1}{L}\int_0^L |f(x)|\mathrm{d}x \tag{1}$$

The unit of measurement of roughness is the micrometre (μm), and, according to ISO:1302 4288:1996, machining processes are able to produce ranges of *Ra* from 0.006 μm to 50 μm. This parameter is primarily used to monitor the production process, which may gradually change the surface due, for example, to the wear of the cutting tool. As *Ra* is an average, the defects on the surface do not have much influence on its results. Hence *Ra* is not used for defect detection. However, roughness is of significant interest in manufacturing because it correlates strongly with the friction interaction with another surface. The roughness of a surface defines how that surface feels, how it looks, how it behaves in a contact with another surface, and how it behaves for coating or sealing. For moving parts, roughness determines how the surface will wear, how well it will retain lubricant, and how well it will hold a load.

Surface roughness can be measured by contact or non-contact methods, but all the methods are based on recording the surface height profile. We selected contact methods because of their reliability, although they are a *post-process* control and do not allow immediate corrective actions. Robust models to predict the surface roughness from experimental data can be very helpful for solving this problem, i.e. to guarantee surface quality *during* the machining process. These models would serve as virtual sensors acting while the machining process is taking place to optimize the final surface roughness. Some of the most recent proposals adopted by researchers to predict surface roughness are presented below.

The most commonly used artificial intelligence (AI) techniques within this context are ANNs with different training algorithms. Backpropagation is the most tried and tested algorithm. It provides very good results in the milling process, as investigated by one of the authors (Correa, 2003).

In the last 10 years, Iowa State University has conducted detailed research on topics like prediction and control in machining targeting tool state and surface roughness. The group led by J. Chen has published several studies on this topic for turning and milling. These works included ANNs and neuro-fuzzy nets (Lou & Chen, 1997, 1999; Lou, Chen, & Li, 1999).

Tsai et al. (1999) presented a surface roughness prediction system for the milling process, where they innovated and included spindle vibration and rotation – VAPR (vibration average per revolution) – in the roughness recognition system. Most of the sensors they used were developed for turning. To find the predicted *Ra* value, they developed two statistical models of multiple regression, and one ANN model based on off-line trained backpropagation. The three models were tested in an end-milling operation using 6061 aluminium, and four-flutes tools. The criterion used to judge the model is efficiency and ability to predict average roughness values was the percentage roughness deviation. The results showed that ANN model predictions are much closer to the real *Ra* values than using the multiple regression model.

Feng and Wang (2003) focused on the development of an empirical model for predicting surface roughness in turning, comparing a linear regression model with an ANN using the same input/output variables in both models. They used the geometric roughness model of Boothroyd and Knight (1989) to predict surface finish: $R_i = f^2/32r'$, where R_i is the arithmetic mean expected of surface roughness (μm), f is the feed rate (mm/rev) and r' is the tool nose radius (mm). The model took on a relatively large radius and a slow velocity. One conclusion of this study was that the ANN and regression models are quite similar with respect to the errors. Both models have a statistically satisfactory behaviour from the modelling point of view.

Benardos and Vosniakos (2003) reviewed the state of the art of the prediction of surface roughness in machining, emphasizing two main problems: (1) determination of the values of the process parameters that produce the desired quality product (technical specifications); (2) maximization of the manufacturing system performance using available resources. One conclusion was a recommendation to use a combination of AI research approaches. Another contribution of this paper is a fishbone diagram of the set of parameters that are believed to influence surface roughness.

Yang, Chen, Chow, and Lin (2006) proposed an adaptive surface roughness control system for end-milling operations. This system was based on the neuro-fuzzy training scheme proposed by Chen (2000). The fuzzy regions were defined for each parameter: cutting speed, feed rate, resulting force on the cutting plane (*Fxy*), normal force to the cutting plane (*Fz*), *Ra* deviation (*DRa*) and feed rate deviation (*Df*). The system has two subsystems, one for predicting in-process *Ra* and another to control the feed rate (*Df*) that is adapted based on the predicted *Ra*.

Kirby et al. (2004) published the development of a surface roughness prediction system using accelerometers in a turning operation with multiple regression techniques. In 2006, the same authors (Kirby, Chen, & Zhang, 2006), developed an adaptive control system that uses the same technique proposed by Chen (2000) and developed for milling by Yang et al. (2006). These models require more flexibility to be adapted for use in industry.

2.2. Experimental procedure and data collection

Because experiments can be very expensive in terms of costs and time, our data collection was based on two designs of experiments

Please cite this article in press as: Correa, M., et al. Comparison of Bayesian networks and artificial neural networks ... *Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.09.024

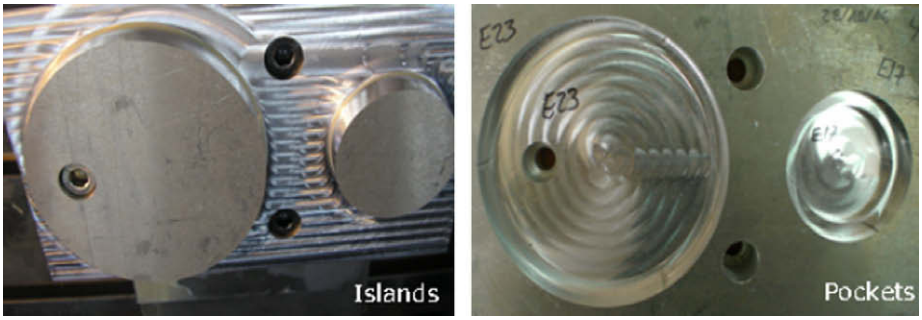

Fig. 1. Geometric design of the test profiles in experiment 1.

(DoE) in a Kondia HS1000 machining center equipped with a Siemens 840D open-architecture CNC, machining 170 × 100 × 25 aluminium samples with hardnesses ranging from 65 to 152 Brinell. As referred to earlier, these materials are commonly used in automotive and aeronautical applications.

A fractional design using eight factors at two levels each (2^8) was applied for an initial screening. This way only a small number of experiments were necessary. For this design, islands (convex) and pockets (concave) were used to represent the geometry, as shown in Fig. 1, with a 10 mm maximum depth of cut cutting with Sandvik tools. The eight factors or variables selected were: feed per tooth (*fz*), cutting speed (*Vc*), axial depth of cut (*ap*), tool diameter (*diam*), radial depth of cut (*ae*), material hardness (*HB*), geometry radius (*Radius*) and geometry curvature (*geom*). The *Ra* labels (class) were allocated in accordance with the average roughness value (μm) established according to ISO:1302 (2002). They are as follows: Mirror (0.10 μm), Polished (0.20 μm), Ground (0.40 μm), Smooth (0.80 μm), Fine (1.60 μm), Semi-fine (3.20 μm), Medium (6.30 μm), Semi-rough (12.50 μm), Rough (25 μm) and Clean (50 μm). The summary of the levels is presented in Table 1. There were 32 experiments, see Table 2.

In addition to the above factors, the following cutting parameters were included: spindle speed (*rpm*), feed rate (*f*) and the resulting forces applied to all the cutting planes (F_{XY}). The cutting forces were measured using a Kistler 9257B dynamometer.

Before applying a second design, a feature subset selection was performed to discard irrelevant variables. The goodness of a subset of variables was assessed using a filter approach, i.e. ranking the variables in terms of some scoring metric usually based on intrinsic characteristics of the data computed from simple statistics on the empirical distribution. Here we chose the information gain with respect to *Ra* (the class variable) as the scoring metric to evaluate the worth of a variable. The information gain $I(Ra, X)$ is the difference between the entropy of *Ra* and the conditional entropy of *Ra* given *X* for any variable *X*. The process was performed using Weka[1] software, a free suite of machine learning software written in Java, developed at the University of Waikato (Australia).

Five variables were selected: *fz*, *ae*, *diam*, *HB*, *geom*. They resulted in a combination of curvature and radius characteristics. These were then used in the second experiment.

For the second round of experiments, the Response Surface Methodology (RSM) (Myers & Montgomery, 2002) was applied, using the final five selected variables, this time with five levels. The summary of the levels is presented in Table 3. The only cutting parameters selected were *rpm* and F_{XY}. The RSM planning resulted in 26 experiments (see Table 4), and three replications to estimate the error. The geometries selected in this design are shown in Fig. 2.

In all cases the surface roughness *Ra* was measured with the Karl Zeiss Surfcom 130 stylus profilometer.

In short, we collected 1262 data items to build the models, i.e. for network learning and testing, from the variables shared by both experiments. Like the feature subset selection, these models were developed using the Weka program.

3. Bayesian network approach

A BN is a probabilistic graphical model, a directed acyclic graph that represents a set of variables (nodes) and their probabilistic conditional independencies (encoded in its arcs). Nodes can represent any kind of variable: a measured parameter, like *Ra*, a latent variable or a hypothesis. There are efficient algorithms that perform inference and learning in BNs (Neapolitan, 2004; Castillo, Gutiérrez, & Hadi, 1997).

If there is an arc from node A to another node B, A is called a parent of B, and B is a child of A. The set of parent nodes of a node x_i is denoted by $parents(x_i)$. A directed acyclic graph is a BN relative to a set of variables if the joint probability distribution of the node variables can be written as the product of the local distributions of each node and its parents as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(x_i))$$

The aim of supervised classification is to classify instances *i* given by certain characteristics $\mathbf{x_i} = (x_{i1}, \ldots, x_{in})$ into *r* class labels, c_i, $i = 1, \ldots, r$. x_{il} denotes the value of variable x_l observed in instance *i*. The main principle of a Bayesian classifier is the application of Bayes' theorem. Bayes' theorem, Eq. (2), calculates the *posterior* probability $P(c_j|\mathbf{x_i})$ from the conditional probabilities $P(\mathbf{x_i}|c_k)$ and the prior probabilities $P(c_k)$ as

$$P(c_j|\mathbf{x_i}) = \frac{P(\mathbf{x_i}|c_j)P(c_j)}{\sum_k P(\mathbf{x_i}|c_k)P(c_k)} \quad (2)$$

The *posterior* probability $P(c_j|\mathbf{x_i})$ is the probability that a sample with characteristics $\mathbf{x_i}$ belongs to class c_j. The prior probability $P(c_j)$ is the probability that a sample belongs to class c_j given no information on its characteristic values. The probabilities of Eq. (2) can be estimated from the expert or from a training set required to build the classifier, where each instance *i* is given by $(\mathbf{x_i}, c_i)$.

Bayes' theorem is used to predict the class and classify each unseen instance: a new instance or example *j*, only characterized with the values $\mathbf{x_j}$ of the predictor variables, is given a class label according to the class that has the maximum *posterior* probability. A useful property of the Bayesian classifier is that it is optimum in the sense that the expected rate of misclassifications is reduced to a minimum (Ripley, 1996).

3.1. Tree Augmented Naïve Bayes

Among the different Bayesian classifiers, we will focus on two specific structures: Naïve Bayes and Tree Augmented Naïve Bayes

[1] http://www.cs.waikato.ac.nz/ml/weka/.

Table 1
Factors and factor levels used in the DoE in experiment 1

Factors	*fz*	*Vc*	*ap*	*diam*	*ae*	*HB*	*Radius*	*geom* (1/radius)
Units	mm/tooth	m/min	mm	mm	mm	Brinell	mm	Island (−), pocket (+)
	0.04	500	5	12	1	65	20	−0.050
	0.04	500	5	12	1	65	20	+0.050
	0.13	850	10	16	5	70	40	−0.025
	0.13	850	10	16	5	70	40	+0.025

Table 2
Planning of experiment 1

Exp.	*fz* (mm/tooth)	*Vc* (m/min)	*ap* (mm)	*diam* (mm)	*ae* (mm)	*HB* (Brinell)	*geom* (mm)	*rpm*
E1-1	0.04	500	5	12	1	65	−0.050	13263
E1-2	0.13	500	5	12	1	145	−0.025	13263
E1-3	0.04	850	5	12	1	145	+0.025	22547
E1-4	0.13	850	5	12	1	65	+0.050	22547
E1-5	0.04	500	10	12	1	145	+0.050	13263
E1-6	0.13	500	10	12	1	65	+0.025	13263
E1-7	0.04	850	10	12	1	65	−0.025	22547
E1-8	0.13	850	10	12	1	145	−0.050	22547
E1-9	0.04	500	5	16	1	65	+0.025	9947
E1-10	0.13	500	5	16	1	145	+0.050	9947
E1-11	0.04	850	5	16	1	145	−0.050	16910
E1-12	0.13	850	5	16	1	65	−0.025	16910
E1-13	0.04	500	10	16	1	145	−0.025	9947
E1-14	0.13	500	10	16	1	65	−0.050	9947
E1-15	0.04	850	10	16	1	65	+0.050	16910
E1-16	0.13	850	10	16	1	145	+0.025	16910
E1-17	0.04	500	5	12	5	65	+0.050	13263
E1-18	0.13	500	5	12	5	145	+0.025	13263
E1-19	0.04	850	5	12	5	145	−0.025	22547
E1-20	0.13	850	5	12	5	65	−0.050	22547
E1-21	0.04	500	10	12	5	145	−0.050	13263
E1-22	0.13	500	10	12	5	65	−0.025	13263
E1-23	0.04	850	10	12	5	65	+0.025	22547
E1-24	0.13	850	10	12	5	145	+0.050	22547
E1-25	0.04	500	5	16	5	65	−0.025	9947
E1-26	0.13	500	5	16	5	145	−0.050	9947
E1-27	0.04	850	5	16	5	145	+0.050	16910
E1-28	0.13	850	5	16	5	65	+0.025	16910
E1-29	0.04	500	10	16	5	145	+0.025	9947
E1-30	0.13	500	10	16	5	65	+0.050	9947
E1-31	0.04	850	10	16	5	65	−0.050	16910
E1-32	0.13	850	10	16	5	145	−0.025	16910

Table 3
Factors and factor levels used in the DoE in experiment 2

Factors	*fz*	*diam*	*ae*	*HB*	*geom* (1/radius)
Units	mm/tooth	mm	mm	Brinell	Convex (−), concave (+)
	0.025	8	1	67	−0.042
	0.050	10	2	92	−0.021
	0.750	12	3	94	0
	0.100	16	4	145	+0.021
	0.130	20	5	152	+0.042

(TAN). The first paradigm (Minsky, 1961) is the simplest model. It is defined by the conjunction between the conditional independence hypotheses of the predictor variables given the class, yielding the following factorization to be substituted in Eq. (2):

$$P(\mathbf{x_i}|c_j) = P(x_{i1}|c_j)\cdots P(x_{in}|c_j)$$

Although this assumption is violated on numerous occasions in real domains, the paradigm still performs well in many situations (Domingos & Pazzani, 1997; Hand & You, 2001).

The TAN classifier (Friedman, Geiger, & Goldszmit, 1997) extends the Naïve Bayes model with a tree-like structure across the predictor variables. This tree is obtained by adapting the algorithm proposed by Chow and Liu (1968) and calculating the conditional mutual information for each pair of variables given the class.

3.2. Surface roughness modelling: a Bayesian network approach

As mentioned above, seven variables measured inside and outside the milling process were taken to construct the network structure. The average surface roughness, *Ra*, was chosen as the *class* variable, and it was the only variable measured post-process.

Table 5 shows the variables used and the respective assigned intervals. In our case we took the range from Mirror to Smooth for *Ra* because it is only possible to obtain at most Smooth *Ra* values with the face milling operation and type of material (aluminium) used in the experiments. The *geom* labels correspond to: S = slot, Convex−− = convex geometry with greater angles, Concave++ = concave geometry with greater angles, Convex− = convex geometry with smaller angles and Concave+ = concave geometry with smaller angles. Fig. 3 shows an example of these labels.

Fig. 4 shows the TAN structure we have learnt. It illustrates the relationships and type of causal effect existing between its nodes. This provides more information on the relationship of each variable with the class and on the relationship among all predictor variables than ANNs (see the next section), which work like a black box. It is interesting to see how the physical relationship between *geom*, *fz* and F_{XY} is evident from the BN structure. Beyond the obvious dependence of all the variables on *Ra*, the *geom* and F_{XY} variables are also influenced by the material hardness (*HB*). This kind

Table 4
Planning of experiment 2

Exp.	*fz* (mm/tooth)	*diam* (mm)	*ae* (mm)	*HB* (Brinell)	*geom* (mm)	*rpm*
E2-1	0.100	16	2	145	+0.021	18000
E2-2	0.100	10	4	145	+0.021	18000
E2-3	0.050	16	4	145	−0.021	18000
E2-4	0.100	16	4	92	−0.021	18000
E2-5	0.100	16	2	92	−0.021	18000
E2-6	0.100	10	2	92	+0.021	18000
E2-7	0.050	10	2	145	−0.021	18000
E2-8	0.050	10	4	92	+0.021	18000
E2-9	0.050	16	2	145	+0.021	18000
E2-10	0.100	10	4	145	−0.021	18000
E2-11	0.050	16	4	92	+0.021	18000
E2-12	0.050	10	2	92	−0.021	18000
E2-13	0.025	12	3	94	0.000	18000
E2-14	0.130	12	3	94	0.000	18000
E2-15	0.075	8	3	94	0.000	22500
E2-16	0.075	20	3	94	0.000	15000
E2-17	0.075	12	1	94	0.000	18000
E2-18	0.075	12	5	94	0.000	18000
E2-19	0.075	12	3	67	0.000	18000
E2-20	0.075	12	3	152	0.000	18000
E2-21	0.075	12	3	94	−0.042	18000
E2-22	0.075	12	3	94	0.042	18000
E2-23	0.075	12	3	94	0.000	18000
E2-24	0.075	12	3	94	0.000	18000
E2-25	0.075	12	3	94	0.000	18000
E2-26	0.075	12	3	94	0.000	18000

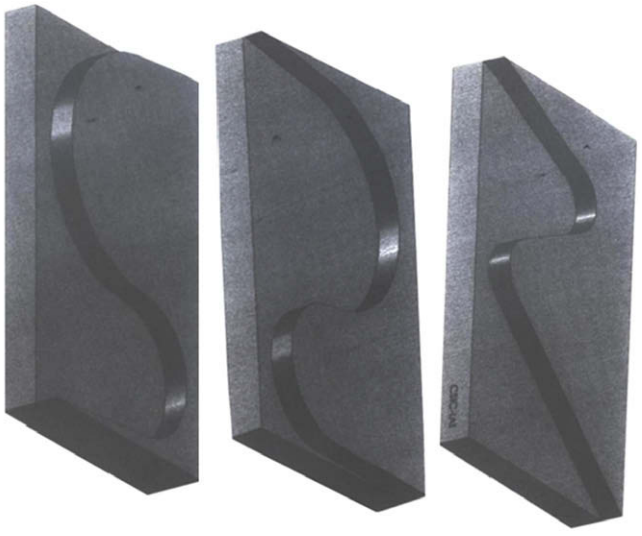

Fig. 2. Geometric design of the test profiles in experiment 2.

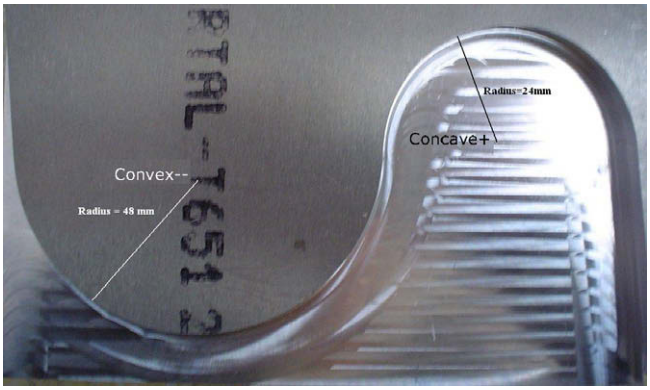

Fig. 3. Example of the use the geometry labels in a profile.

of "discovery" is one of the aims of this paper because it cannot be derived from any physical equation of the cutting process. Moreover, feed per tooth (*fz*) is directly related to *HB*. Finally, splindle speed (*rpm*) and radial depth of cut (*ae*) depends probabilistically

Table 5
Variables and intervals assigned

States	Variables and domains							
	F_{XY} (N) [lower, upper)	*fz* (mm/tooth) [lower, upper)	*diam* mm	*ae* mm	*HB* (Brinell) [lower, upper)	*geom* (mm) Label [lower, upper)	*rpm* [lower, upper)	*Ra* (μm) Label [lower, upper)
0	[22, 79)	[0.0250, 0.0460)	8	1	[65, 92)	Convex−− [−50, −30)	[15000, 17500)	Mirror [0.10, 0.25)
1	[79, 88)	[0.0460, 0.0670)	10	2	[92, 94)	Convex− [−30, −10)	[17500, 20000)	Polished [0.25, 0.35)
2	[88, 100)	[0.0670, 0.0880)	12	3	[94, 94)	*S* [−10, 10)	[20000, 22500)	Ground [0.35, 0.75)
3	[100, 148)	[0.0880, 0.1090)	16	4	[94, 94)	Concave+ [10, 30)		Smooth [0.75, 1.50)
4	[148, 178)	[0.1090, 0.1300]	20	5	[94, 109)	Concave++ [30, 50]		
5	[178, 208)				[109, 111)			
6	[208, 228)				[111, 145)			
7	[228, 275)				[145, 152]			
8	[275, 324)							
9	[324, 488]							

Please cite this article in press as: Correa, M., et al. Comparison of Bayesian networks and artificial neural networks ... *Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.09.024

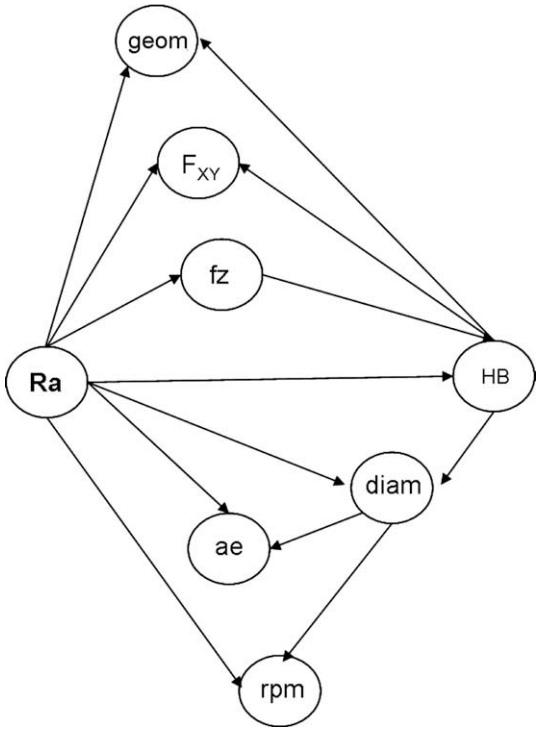

Fig. 4. Bayesian network using TAN algorithm.

on the tool diameter (*diam*), since these variables are physically associated with diameter.

TAN network learning also involves estimating all the conditional probability distributions of each variable given its parents, i.e. its quantitative part. See, for example, Table 6, which shows the distribution of *HB* given *Ra* and *fz*. From any BN we can reason in any direction, querying the network about any marginal probability or any *posterior* probability given some evidence (observation).

4. Artificial neural network approach

An ANN is a mathematical or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase (Hecht-Nielsen, 1990).

An ANN usually organizes its units into several layers. The first layer or input layer, the intermediate layers or hidden layers, which are not always present because they are sometimes not needed, and the last or output layer. The information to be analyzed is presented (or fed) to the neurons of the first layer and then propagated to the neurons of the second layer for further processing. These results are propagated to the next layer and so on, through to the last layer, converting the information into the network output. The goal of an ANN is to discover some association between input and output patterns.

4.1. Backpropagation algorithm

The backpropagation algorithm is a supervised learning method, an implementation of the Delta rule. It is more useful for feed-forward networks (networks that have no feedback, or simply, that have no loop connections). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires all transfer functions used by the artificial neurons (or "nodes") to be differentiable (Hagan, Demuth, & Beale, 1996).

Backpropagation is used to calculate the error gradient of the network with respect to its modifiable weights. This gradient is almost always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Backpropagation may have practical problems of getting trapped in local minima and knowing when the procedure has converged.

It is important to note that backpropagation networks are necessarily multilayer perceptrons (MLP) usually with one input, one hidden, and one output layer. They can be represented as a function

$$c_k = f_k\left(\alpha_k + \sum_{j\to k} w_{jk} f_j\left(\alpha_j + \sum_{i\to j} w_{ij} x_i\right)\right)$$

where w_{ij} (w_{jk} resp.) are the weights that connect input i to the hidden layer k (connect hidden layer j to the output layers k, resp.), α are the biases and f are the activation functions.

For the hidden layer to provide a useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Commonly used non-linear activation functions include the logistic function, the softmax function, and the Gaussian function.

The backpropagation algorithm for calculating a gradient has been rediscovered a number of times and is a special case of a more general technique called *automatic differentiation* (AD) (Griewank, 2000). Fundamental to AD is the decomposition of differentials provided by the chain rule. For the simple composition $f(x) = g(h(x))$, the chain rule gives $\frac{df}{dx} = \frac{dg}{dh}\frac{dh}{dx}$. There are usually two distinct AD modes, forward accumulation (or forward mode) and reverse accumulation (or reverse mode, which is used for the backpropagation algorithm). Forward accumulation specifies that one traverses the chain rule from right to left, that is, $\frac{dh}{dx}$ is computed first and next $\frac{dg}{dh}$, and reverse accumulation traverses from left to right.

Table 6
Conditional probability table of $P(HB|Ra,fz)$

fz	4	4	4	4	3	3	3	3	2	2	2	2	1	1	1	1	0	0	0	0
HB/*Ra*	S	G	P	M	S	G	P	M	S	G	P	M	S	G	P	M	S	G	P	M
7	0.12	0.12	0.12	0.12	0.84	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
6	0.12	0.12	0.12	0.12	0.02	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
5	0.12	0.12	0.12	0.12	0.02	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
4	0.12	0.12	0.12	0.12	0.02	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
3	0.12	0.12	0.12	0.12	0.02	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.15
2	0.13	0.13	0.13	0.13	0.02	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
1	0.13	0.13	0.13	0.13	0.03	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
0	0.14	0.14	0.14	0.14	0.03	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14

Ra labels: S = smooth, G = ground, P = polished, M = mirror. *HB* and *fz* labels are shown in Table 5.

4.2. Surface roughness modelling: An artificial neural network approach

To build an ANN, the same data set was used with the same variables as in the above BN.

We built an MLP network with a 7–11–4 topology, i.e. seven input nodes, a hidden layer of eleven neurons and a final output layer with four neurons providing the predicted *Ra* class. As noted above, we had already used this network successfully in the milling process (Correa, 2003). The parameters were set as: a sigmoid function activation in all layers, momentum 90% and learning rate 1%. The input variables used in the first layer are the same input variables as in the BN and the output layer corresponds with the four states of the *Ra* class, see Fig. 5.

5. Comparison study: Bayesian networks vs. artificial neural networks

In this section, we compare the two classifiers, BNs and ANNs, assuming that the aim of a classification model is to correctly classify new cases.

5.1. Experimental setting and performance measures

Models should not be validated on the same data used to create the classifier. Accordingly, the *K*-fold cross-validation method was chosen (Stone, 1974). The original sample is partitioned into *K* disjoint subsamples. Of the *K* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $K-1$ subsamples are used as training data. The cross-validation process is then repeated *K* times (the folds), with each of the *K* subsamples used exactly once as the validation data. Then, the *K* results from the folds are averaged (or otherwise combined) to produce a single estimate of the classifier accuracy. In both models we chose $K = 10$ and the same seed in the selection of randomness for determining the folds. Thus, we have the same folds for both classifiers and therefore the classification results can be considered as paired samples.

There are several statistical tests for comparing a performance measure of two classifiers over a single data set (Demšar, 2006). Also, there are several performance measures to compare the goodness of a classifier. Here, we use the classification accuracy or percentage of correctly classified observations. Finally, we analyse the interpretability of both models.

5.1.1. Classification accuracy

The confusion matrices show the classifier accuracy, see Tables 7 and 8. This is the honest estimate of the true error rate, i.e. an indicator of how good the classifier is or the probability of it classifying new cases correctly. The BN appears to output better results than the ANN classifier, with an accuracy of 96.3% and 94.8%, respectively. Looking closer at each *Ra* class, most classes of the BN classifier outweigh those of the ANN. Note that the BN correctly classifies 100% of the cases with a Smooth *Ra* whereas the ANN only manages 89%. Ground *Ra* behaves similarly: only one case is misclassified by the BN, against nine cases misclassified by the ANN. The same situation is exhibited for Polished *Ra* with 17 cases misclassified by the BN (3.4%) compared to the 30 cases misclassified by the ANN (6%). Only for Mirror *Ra*, were the ANN results slightly better with a 94% accuracy against 92.5% for the BN.

Table 9 summarises the measures of merit of each classifier. The formulas for calculating the indicators are as follows:

$$\text{TP Rate} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\text{FP Rate} = \frac{\text{FP}}{\text{TN}+\text{FP}}$$

$$\text{Recall} = \text{TP Rate}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \times 100$$

$$\text{F-measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall}+\text{Precision}}$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{FP}+\text{TN}}$$

$$\textit{Kappa} = \frac{P(A)-P(E)}{1-P(E)}$$

where TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative

The *Kappa* statistic includes measures of class accuracy within an overall measurement of classifier accuracy. It is a better

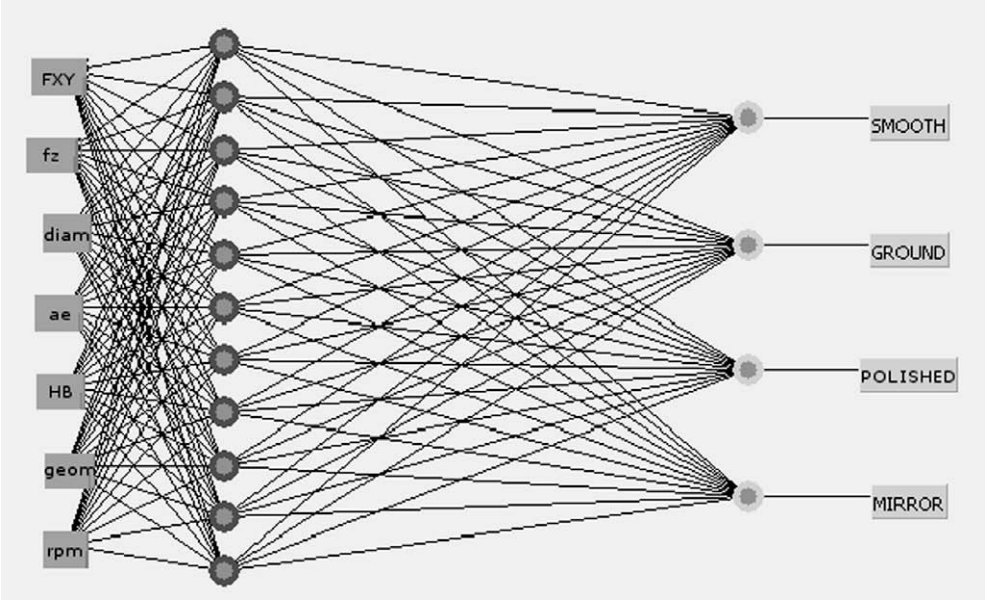

Fig. 5. Artificial neural network using backpropagation algorithm.

Table 7
Confusion matrix using BN

REAL →	Smooth	Ground	Polished	Mirror
ASSIGNED ↓				
Smooth	18 (100%)	0	0	0
Ground	0	359 (99.7%)	1 (0.3%)	0
Polished	0	2(0.4%)	493 (96.6%)	15 (3%)
Mirror	0	0	28 (7.5%)	346 (92.5%)

Table 8
Confusion matrix using ANN

REAL →	Smooth	Ground	Polished	Mirror
ASSIGNED ↓				
Smooth	16 (89%)	0	2 (11%)	0
Ground	1 (0.5%)	351 (97.5%)	8 (2%)	0
Polished	2 (0.4%)	3 (0.6%)	480 (94.8%)	25 (5%)
Mirror	0	2 (0.5%)	22 (5.5%)	350 (94%)

Table 9
Summary of the measures of merit for each classifier

Measure	BN	ANN
Correctly classified instances	1216 (96.35%)	1197 (94.84%)
Incorrectly classified instances	46 (3.64%)	65 (5.15%)
Kappa statistic	0.94	0.92
Mean absolute error (MAE)	0.03	0.04
Root mean squared error (RMSE)	0.13	0.14
Relative absolute error (RAE)	10.41%	13.05%
Root relative squared error (RRSE)	32.66%	33.70%

measure of classifier accuracy than overall accuracy because it considers inter-class agreement.

Kappa is computed as $\frac{P(A)-P(E)}{1-P(E)}$, where $P(A)$ is the observed agreement among the predicted and the observed class, and $P(E)$ is the expected agreement, that is, $P(E)$ represents the probability that this agreement is by chance. The values of *Kappa* are constrained to the interval $[-1,1]$. A *Kappa* value of 1 means perfect agreement, a *Kappa* value of 0 means that agreement is equal to chance, and $a-1$ *Kappa* value means "perfect" disagreement.

Note that the BN classifier outputs better indicators than the ANN classifier, reflected not only in the *Kappa* statistic, but in the lower MAE and RMSE rates.

A more detailed accuracy per *Ra* class is shown in Fig. 6. The solid color bars correspond to the BN model, while the dotted bars correspond to the ANN model. Note that although the "Accuracy" appears to be quite similar with both classifiers, the BN reports better performance measures than the ANNs.

However, the role of randomness must be evaluated and measured by conducting hypothesis tests to determine the statistical significance of our performance estimates. We use two statistical tests to determine whether one learning algorithm outperforms the other one on the surface roughness prediction task.

First, we took our *K*-fold cross-validated paired results. A single tailed paired *t*-test yielded a *p*-value of ≈ 0.001. The null hypothesis was that the mean accuracy of the BN classifier is equal to the mean accuracy of the ANN classifier. The *p*-value indicates that the null hypothesis should be rejected, concluding that BN accuracy is statistically superior to ANN accuracy. The values for obtaining these results are shown in Table 10, where $p_i^A(p_i^B)$ is an estimation of the probability of correctly classifying new cases when the BN (ANN, respectively) has been tested using fold i (the rest of data were used as training data).

Additionally, we also calculated the McNemar test based on a χ^2 test of goodness-of-fit that compares the distribution of counts expected under the null hypothesis to the observed counts. Under the null hypothesis, the two algorithms should have the same error rate, which means that $n_{01} = n_{10}$, see Table 11.

If the null hypothesis is true, the following statistic is distributed as a χ^2 distribution with 1 degree of freedom. It incorporates a Yates' correction for continuity to account for the fact that the statistic is discrete while the χ^2 distribution is continuous:

$$\frac{(|n_{01}-n_{10}|-1)^2}{n_{01}+n_{10}} = \frac{(|6-25|-1)^2}{6+25} = 10.45$$

This result provides a *p*-value $\ll 0.05$. So the null hypothesis can be rejected in favor of the hypothesis that the two algorithms perform differently, and BN is superior.

5.1.2. Interpretability

ANNs have the disadvantage of not having symbolic reasoning and semantic representation. An ANN generally takes the shape of a "black box" model in the sense that the non-linear relationships of cause and effect are not easily interpretable, making it difficult to explain the results.

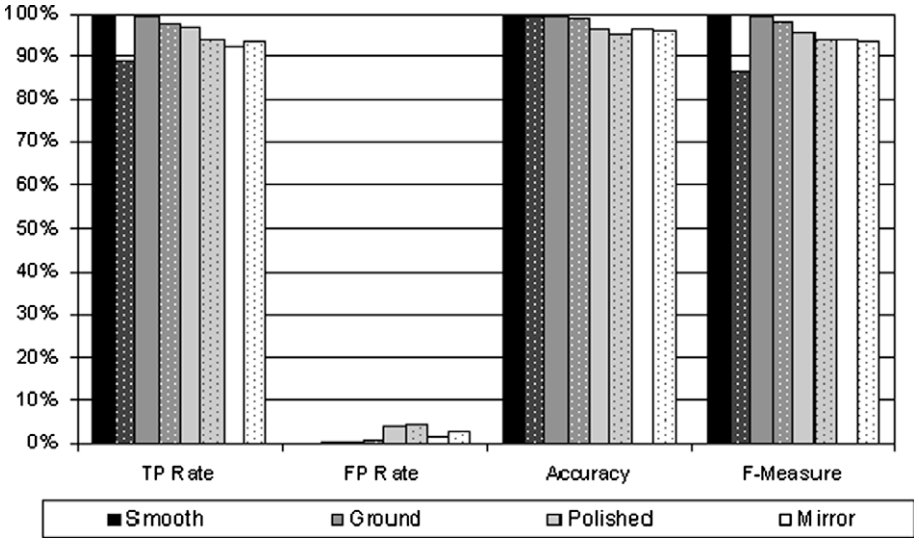

Fig. 6. Detailed accuracy by class.

Table 10
Summary of the paired *t*-test

Fold i	BN p_i^A	ANN p_i^B	BN - ANN $p_i^A - p_i^B$
1	0.94338	0.93224	0.01114
2	0.96905	0.94576	0.02328
3	0.94842	0.95018	−0.00175
4	0.96994	0.94024	0.02971
5	0.97331	0.93801	0.03531
6	0.96095	0.93166	0.02929
7	0.95064	0.91633	0.03430
8	0.95805	0.91207	0.04598
9	0.95805	0.94543	0.01261
10	0.94153	0.93926	0.00227
Mean			0.02221
Standard deviation			0.01555
t			4.51683

Table 11
Contingency table for McNemar test

Number of examples misclassified by both *BN* and *ANN* $n_{00} = \mathbf{38}$	Number of examples misclassified by *BN* but not by *ANN* $n_{01} = \mathbf{6}$
Number of examples misclassified by *ANN* but not by *BN* $n_{10} = \mathbf{25}$	Number of examples misclassified by by neither *BN* nor *ANN* $n_{11} = \mathbf{1193}$

On the other hand, the main advantage of BNs is that reasoning is based on a real-world model. The system has a thorough understanding of the processes involved, rather than just a mere association of data and assumptions. This is combined with a strong probabilistic theory enabling BNs to give an objective interpretation.

We can ask a BN questions involving observations or evidence in order to find the *posterior* probability of any variable or set of variables given some evidence. This makes different types of reasoning possible.

One type of question is *predictive reasoning* or causal inference. For example, we may ask "What is the probability of each *Ra* class given certain manufacturing requirements?" This is a prediction of effects. If we need to produce a 40 mm radius piece of aluminium with *HB* 67 Brinell and concave geometry (*geom* is Concave++), we would ask the BN to compute $P(Ra|HB = 67, geom = \text{Concave}++)$. Propagating this evidence, the network computes the following *Ra* probabilities: Mirror with a probability of 0.25, Polished with a probability of 0.57, Ground with a probability of 0.01, and Smooth with a probability of 0.17. With these requirements the highest probabilities give good qualities. This reasoning is correct since it corresponds with values obtained in experimental tests.

We can also ask *diagnostic reasoning* questions, like "What are the probabilities of unobserved variables if *Ra* is restricted?" Suppose then that we need to manufacture a piece with *Ra* = Mirror, and we want to know which are the model recommendations regarding *geom*, *HB* and *fz* to achieve that *Ra* class. We should compute $P(geom, HB, fz|Ra = \text{Mirror})$. Moreover, another advantage of BNs is the possibility of finding the most likely explanation or *abductive inference*. In our case, we look for the configurations of those three variables with the highest probability. The network recommendation for obtaining this *Ra* state is to manufacture pieces with soft hardness, 65–92 Brinell, geometry in convex++ and *fz* value in [0.08, 0.10) with a probability of 0.22. On the other hand, if we need to manufacture a piece with *Ra* = Ground, we ask the BN about $P(geom, HB, fz|Ra = \text{Ground})$. The network recommendation is medium hardness, 94-109 Brinell, geometry in slots and *fz* value in [0.04, 0.06) with a probability of 0.56.

A *total* abduction finds the configuration of all the unobserved variables that maximize the evidence probability, e.g. $\arg\max_{x_1,\ldots,x_n} P(x_1, \ldots, x_n|Ra = \text{Mirror})$. In this case, the most likely configuration is *diam* at 12 mm, *ae* at 2 mm, *HB* at medium hardness, 109–111 Brinell, F_{XY} ranging from 324 to 488 N, *geom* in slots, *fz* in [0.06, 0.08) mm/tooth and *rpm* at low revolutions, from 15000 to 17500 rpm. This means that for obtaining a piece with *Ra*=Mirror, the previous configuration is the most likely configuration of the remaining seven variables.

6. Conclusions and discussion

An ANN is a model often used to predict surface quality in machining processes. In this paper, we propose using BNs instead, showing a number of advantages over ANNs and extending the application domain to include features, not easily found in the experimental studies, that influence surface roughness, like the geometry of the workpiece and the hardness of material to be machined.

After validating both models with the same data and technique (*K*-fold cross-validation), BNs achieve the best results from the point of view of classifier goodness applied to the problem of quality prediction in high-speed milling processes. The results have been confirmed by several hypothesis tests.

As for the time it takes to build the model, BNs also outperform ANNs, requiring 0.08 CPU seconds and 12.69 CPU seconds, respectively, on a 3 GHz, 1.5 GB Dell Dimension PC.

The ANN optimization procedure does not guarantee the convergence to a global minimum. There are no principled methods of choosing the network parameters (number of hidden layers, number of nodes in the hidden layer(s), form of activation functions). On the other hand, BNs have an easy and fast construction procedure without tuning parameters. Note in favor of ANNs, that the memory requirements, represented by parameters in an analytical form, are smaller than for BNs, represented by tabular conditional probability tables. However, this is not such a relevant question nowadays where computer memory is cheap and extensive.

Relative speeds of operation follow the same pattern. Thus, the BN can be easily implemented as a simple table look-up, and it is intrinsically fast. However, the ANN requires a number of multiplications and additions at evaluation time, rendering it comparatively slow if high intensity predictions are required.

Both classifier models are simple to use, but BNs can be more easily understood by humans. ANN models work like a black box. In contrast, friendly and intuitive BNs help users to update models and increase the confidence in the correctness of the model to be finally adopted. Finding factors that determine surface roughness can help to optimize high-speed milling, which is extensible to other industrial applications. Moreover, BNs support inference in any direction providing responses to any kind of query, not only about surface roughness but also, given some evidence, about different predictor variables. Because they capture these different types of reasoning to infer knowledge, BNs are useful models with significant representation power. From the comparison performed here, BNs are preferred over ANNs.

Acknowledgements

We would like to thank Miguel Ramírez and Antonio Vallejo specially for their help in conducting the experiments.

This work was partially supported by the MOCAVE Project under Grant DPI2006-12736-C02-01 and by the Spanish Ministry of Education and Science, Project TIN2007-62626.

References

Aboulatta, O. B., & Mádl, J. (2001). Surface roughness prediction based on cutting parameters and tool vibrations in turning operations. *Journal of Materials Processing Technology, 118*, 269–277.

Antal, P., Fannes, G., Timmerman, D., Moreau, Y., & de Moor, B. (2003). Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection. *Artificial Intelligence in Medicine, 29*, 39–60.

Benardos, P. G., & Vosniakos, G. C. (2003). Predicting surface roughness in machining: A review. *International Journal of Machine Tools and Manufacture, 43*, 833–844.

Boothroyd, G., & Knight, W. A. (1989). *Fundamentals of machining and machine tools*. NY: Marcel Dekker.

Castillo, E., Gutiérrez, E., & Hadi, A. S. (1997). *Expert systems and probabilistic network models*. NY: Springer.

Chen, J. C. (2000). An effective fuzzy-nets training scheme for monitoring tool breakage. *Journal of Intelligent Manufacturing, 11*, 85–101.

Chow, C. K., & Liu, C. N. (1968). Approximating discrete probability distributions. *IEEE Transactions on Information Theory, 14*, 462–467.

Correa, M. (2003). *A survey of artificial intelligence techniques applied to the modeling of milling process*. Master thesis, Universidad Politécnica de Madrid, Madrid (in Spanish).

Correa, M., Bielza, C., Ramírez, M. de J., & Alique, J. R. (in press). A Bayesian network model for surface roughness prediction in the machining process. *International Journal of Systems Science*.

Demšar, J. (2006). Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero–one loss. *Machine Learning, 29*, 103–130.

Feng, C. X., & Wang, X. F. (2003). Surface roughness predictive modelling: Neural networks versus regression. *IIE Transactions, 35*, 11–27.

Friedman, N., Geiger, D., & Goldszmit, M. (1997). Bayesian network classifiers. *Machine Learning, 29*, 131–161.

Griewank, A. (2000). *Evaluating derivatives: Principles and techniques of algorithmic differentiation*. SIAM.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural network design*. Boston, MA: PWS Publishing Company.

Hand, D., & You, K. (2001). Idiot's Bayes – Not so stupid after all? *International Statistical Review, 69*, 385–398.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Addison-Wesley.

Huang, B., & Chen, J. C. (2003). An in-process neural network-based surface roughness prediction (INN-SRP) system using a dynamometer in end milling operations. *International Journal of Advanced Manufacturing Technology, 21*, 339–347.

ISO 1302:2002, Geometrical Product Specifications (GPS) – Indication of surface texture in technical product documentation.

Kirby, D., Chen, J. C., & Zhang, Z. (2006). Development of a fuzzy-nets-based in-process surface roughness adaptive control system in turning operations. *Expert Systems with Applications, 30*(4), 592–604.

Kirby, D., Zhang, Z., & Chen, J. C. (2004). Development of an accelerometer-based surface roughness prediction system in turning operations using multiple regression techniques. *Journal of Industrial Technology, 20*(4), 1–8.

Lou, J., & Chen, J. C. (1997). In-process surface recognition of a CNC milling machine using the fuzzy nets method. *Computers and Industrial Engineering, 33*(1–2), 401–404.

Lou, J., & Chen, J. C. (1999). In-process surface roughness recognition (ISRR) system in end-milling operations. *International Journal of Advanced Manufacturing Technology, 15*(3), 200–209.

Lou, J., Chen, J. C., & Li, C. M. (1999). Surface roughness prediction technique for CNC end-milling. *Journal of Industrial Technology, 15*(1), 1–6.

Marchant, J. A., & Onyango, C. M. (2003). Comparison of a Bayesian classifier with a multilayer feed-forward neural network using the example of plant/weed/soil discrimination. *Computers and Electronics in Agriculture, 39*, 3–22.

Minsky, M. (1961). Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers, 49*, 8–30.

Myers, R. H., & Montgomery, D. C. (2002). *Response surface methodology: Process and product optimization using designed experiments*. Wiley & Sons.

Neapolitan, R. E. (2004). *Learning Bayesian networks*. Prentice Hall.

Perzyk, M., Biernacki, R., & Kochański, A. (2005). Modeling of manufacturing processes by learning systems: The naïve Bayesian classifier versus artificial neural networks. *Journal of Materials Processing Technology*, 1430–1435.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge, UK: Cambridge University Press.

Samson, S. L., & Chen, J. C. (2003). On-line surface roughness recognition system using artificial networks system in turning operations. *International Journal of Advanced Manufacturing Technology, 22*, 498–509.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B, 36*, 111–147.

Tsai, Y. H., Chen, J. C., & Lou, S. J. (1999). An in-process surface recognition system based on neural networks in end-milling cutting operations. *International Journal of Machine Tools and Manufacture, 39*(4), 583–605.

Yang, L., Chen, J. C., Chow, H., & Lin, C. (2006). Fuzzy-nets-based in-process surface roughness adaptive control system in end-milling operations. *International Journal of Advanced Manufacturing Technology, 28*(3–4), 236–248.

Zhang, R., & Bivens, A. (2007). Comparing the use of Bayesian networks and neural networks in response time modeling for service-oriented systems. In *Proceedings of the 2007 workshop on service-oriented computing performance: Aspects, issues, and approaches*. Monterey California, USA.

Please cite this article in press as: Correa, M., et al. Comparison of Bayesian networks and artificial neural networks ... *Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.09.024