The International Journal of Advanced Manufacturing Technology (2018) 99:1917–1926
https://doi.org/10.1007/s00170-018-2571-z

ORIGINAL ARTICLE

Predicting tool wear with multi-sensor data using deep belief networks

Yuxuan Chen[1] · Yi Jin[1] · Galantu Jiri[2]

Received: 21 February 2018 / Accepted: 13 August 2018 / Published online: 29 August 2018

Abstract

Tool wear is a crucial factor influencing the quality of workpieces in the machining industry. The efficient and accurate prediction of tool wear can enable the tool to be changed in a timely manner to avoid unnecessary costs. Various parameters, such as cutting force, vibration, and acoustic emission (AE), impact tool wear. Signals are collected by different sensors and then constitute the raw data. There are two main types of methods used to make predictions, namely model-based and data-driven methods. Data-driven methods are typically preferred when a mathematical model is not available. In such a situation, artificial intelligent methods, such as support vector regression (SVR) and artificial neural networks (ANNs), are applied. Recently, deep learning algorithms have been widely used because of their accuracy, computing speed, and excellent performance in solving nonlinear problems. In this study, a deep learning network called deep belief network (DBN) is applied to predict the flank wear of a cutting tool. To confirm the superiority of the DBN in predicting tool wear, the performance of the DBN is compared with the performances obtained using ANNs and SVR in terms of the mean-squared error (MSE) and the coefficient of determination (R^2), considering data from more than 900 experiments.

Keywords Tool wear prediction · Deep belief network · Support vector regression · Artificial neural network

1 Introduction

In the machining industry, tool condition is a crucial factor influencing workpiece quality. Various types of failure, such as excessive load, overheating, deflection, fracture, fatigue, corrosion, and wear, can lead to decreased productivity, increased production costs, and unexpected machine downtime. According to previous studies, the cost of maintenance performed to lessen the influence of tool failure can range from 15 to 40% of the cost of goods produced [1]. According to Malekian et al. [2], cutting tool failures typically represent approximately 20% of machine tool downtime, and tool wear has a direct impact on the surface finish quality. Various factors, such as force, cutting force, vibration, acoustic emission (AE), temperature, and surface roughness, may influence tool wear. The wear and condition of a cutting tool must be predicted to ensure adequate replacement and avoid damage.

The prediction of the health condition of a tool based on cutting conditions is referred to as prognostics and health management, and there are two methods used to perform prognostics [3–5], namely model-based and data-driven prognostics. Model-based prognostics are based on the mathematical description of a system. Different algorithms are used to construct the models, such as hidden Markov models [6], Wiener and gamma processes [7], and Kalman filters [8]. For these models, an in-depth understanding of the physical processes is required to make accurate predictions, but in many situations, particularly for complex manufacturing systems and processes, previous knowledge of the system behavior is not available. Therefore, data-driven prognostics have been proposed for such situations.

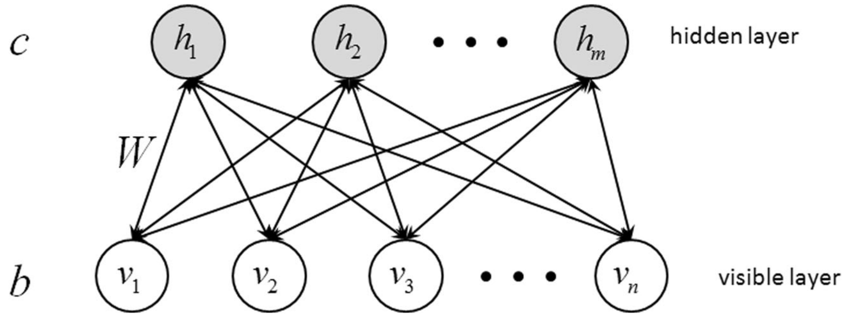
Data-driven prognostics are based on learning algorithms and large training data to build models for predicting. On the one hand, no in-depth knowledge is required for building models; on the other hand, real-time data from machines can be easily collected in condition monitoring systems. Some parameters are generally used in predicting tool wear: cutting force increases with increasing wear, and vibration varies with increasing tool wear due to the rubbing between a workpiece and chip against the tool. Besides, AE is a signal

✉ Yi Jin
jinyi08@ustc.edu.cn

[1] School of Engineering Science, University of Science and Technology of China, Hefei, People's Republic of China

[2] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

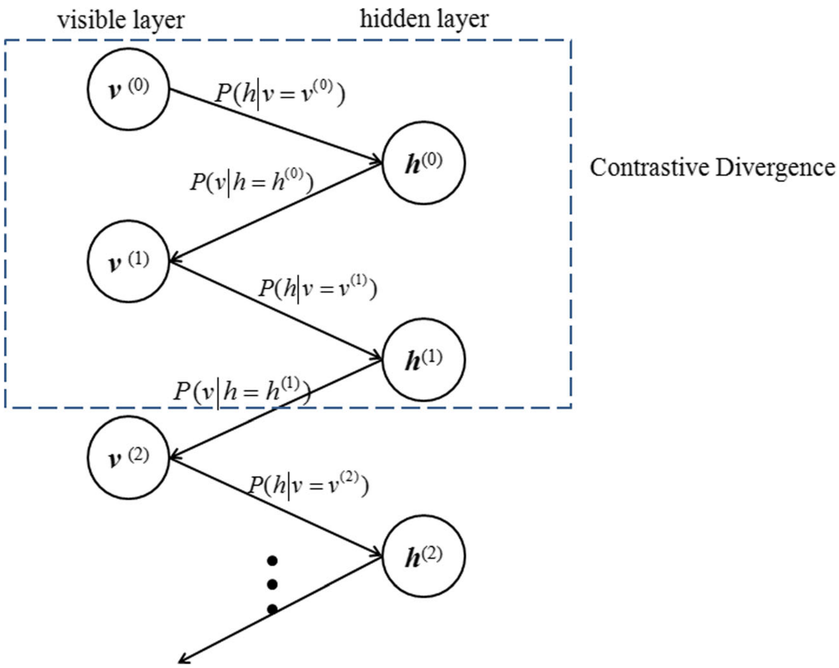
Fig. 1 The RBM architecture



dependent on the state of cutting process because it is the transient elastic energy released in materials deforming when being cut. These signals are collected by different sensors and stored to obtain detailed information regarding the state of the cutter [9]. In other words, massive data are offered to diagnostic systems, but the data are typically collected more rapidly than diagnosticians can analyze it [10], thus representing an issue that data-driven prognostics aim to solve. Many data-driven prognostics have been proposed based on artificial intelligent algorithms, such as support vector regression (SVR), fuzzy logic classifiers, and artificial neural networks (ANNs). Benkedjouh et al. [11] proposed a method for assessing tool condition and predicting lifespan using two nonlinear feature reduction techniques combined with SVR. Zhang and Zhang [12] presented a tool wear model based on a least-squares support vector machine (LS-SVR) for a ball-end milling cutter. That study demonstrated that the LS-SVR-based tool wear model can predict tool wear within a certain range of cutting conditions in milling operations. Furthermore, Li et al. [13] conducted analyses to determine effective features that reveal tool conditions using an improved SVR called v-SVR, where v is a parameter controlling the number of support vectors. ANNs are widely used by researchers because of their high fault tolerance and adaptability, noise suppression capability, and ability to handle large volumes of data [14]. Samanta and Nataraj [15] employed an ANN to diagnose bearing faults and characterize the bearing health conditions. Lee [16] developed a system based on neural networks by adopting a quick propagation algorithm in tool condition monitoring. Recently, D'Addona et al. [17] used an ANN along with DNA-based computing (DBC) to monitor tool wear and improve the accuracy of tool wear degree identification. Moreover, Coppel et al. [18] combined ANN with GA and ACO algorithm in constructing an adaptive control optimization system and estimated cutting tool wear with ANN, which lead to a significant decrease in production cost.

However, neural networks have limitations; specifically, these networks commonly adopt shallow architectures, which

Fig. 2 The Gibbs sampling procedure and the CD algorithm



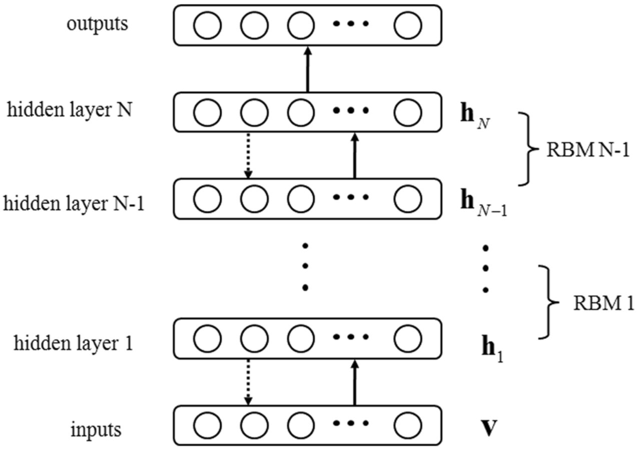


Fig. 3 The DBN architecture

means that the networks only include one hidden layer. This architecture limits the capacity of neural networks to learn the complex nonlinear relationships between tool wear and machining parameters. Thus, a deep architecture network must be established to solve this problem. A deep belief network (DBN) is used in this study.

In 2006, the concept of deep learning and the structure of DBNs were proposed [19]. DBNs employ a hierarchical structure with multiple stacked restricted Boltzmann machines (RBMs) and employ a greedy layer-by-layer learning algorithm followed by a fine-tuning procedure. DBNs have been successfully used in computer vision and automatic speech recognition in the last decade [20, 21]. They have also been used in feature extraction for cutting state monitoring [22], fault diagnosis of aircraft engines and electric power transformer [23], and diagnosis followed by fault characteristic mining for rotating machinery [24]. The aforementioned applications demonstrate that DBNs are a promising tool for handling massive amounts of data, which is one of the challenges in facing the modern manufacturing industry. Although DBNs have been gradually adopted in the manufacturing industry, few studies have focused on their application in tool wear prediction, particularly using data collected from multiple sensors. Thus, we investigate the ability of DBNs to predict tool wear using an experimental dataset composed of cutting force, vibration in three directions, and AE. Furthermore, to demonstrate the superiority of DBNs in predicting tool wear, the performance of DBNs is compared with those of feed-forward back-propagation (FFBP) ANNs and LS-SVR in terms of training time, predicting time, and accuracy.

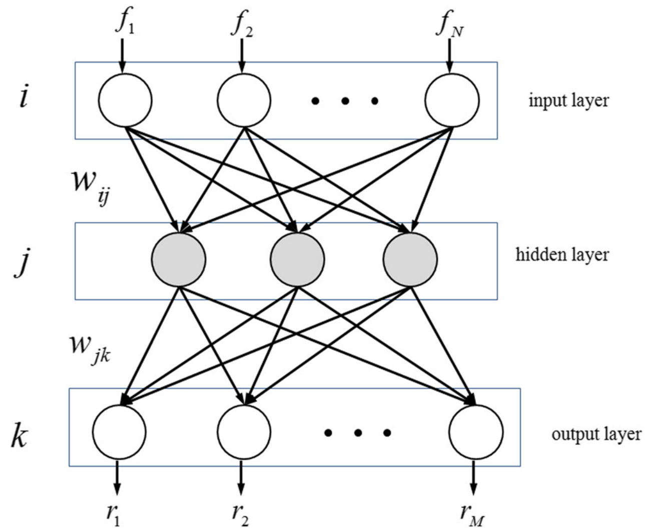


Fig. 4 The ANN architecture

The primary contributions of this paper are as follows:

- Tool wear in milling operations is predicted using a DBN, along with signals collected by multiple sensors. The experimental results demonstrate that the predictions are accurate and stable, with a mean-squared error (MSE) as low as 0.00692 (normalized data). The coefficient of determination (R^2) is as high as 0.9888, representing a significant improvement over existing techniques. Furthermore, this study represents the first application of DBNs for predicting tool wear using multi-sensor data.
- The performances of LS-SVR, ANNs, and DBNs in predicting tool wear are compared. Standard metrics, including the MSE, accuracy of regression (R^2), training time, and prediction time, are used to evaluate each technique. LS-SVR achieved the lowest mean-squared error but required nearly triple the runtime compared to that of other methods. The ANN and DBN produced similar levels of accuracy, but when the number of hidden neurons varied, the DBN was 10% more accurate than the ANN, which means that the DBN was more stable than the ANN.

The remainder of this paper is organized as follows: the DBN methodology and two other algorithms are presented in Section 2. The experimental setup is presented in Section 3, and conclusions and comparisons are drawn in Section 4. The deficiencies of this algorithm and plans for future work are discussed in Section 5.

2 Methodology

2.1 Deep belief network

The DBN is a two-way deep network and is a stacking of numerous RBMs. The training process consists of pre-training, fine-tuning, and prediction.

Table 1 Operating conditions

Parameter	Value
Spindle speed	10,400 RPM
Feed rate	1555 mm/min
y depth (radial) of the cut	0.125 mm
z depth (axial) of the cut	0.2 mm
Sampling rate	50 kHz/channel
Material	Stainless steel

Table 2 Detailed description of the sensor data

Signal collected	Data description
1	F_x: force (N) in the x-direction
2	F_y: force (N) in the y-direction
3	F_z: force (N) in the z-direction
4	V_x: vibration (g) in the x-direction
5	V_y: vibration (g) in the y-direction
6	V_z: vibration (g) in the z-direction
7	AE: acoustic emission (V)

RBMs are an effective feature extraction method for initializing the feed-forward neural network and can significantly improve the generalization capabilities of the network. Each RBM has two layers, called the visible and hidden layers, which composed of binary neurons. The neurons in the hidden layer are denoted as $\mathbf{h} = (h_1, h_2, \ldots, h_n) \in \{0, 1\}$, whereas those in the visible layer are denoted as $\mathbf{v}(v_1, v_2, \ldots, v_m) \in (0, 1)$. As shown in Fig. 1, each neuron in the visible layer is connected to neurons in the hidden layer. This type of fully connected adjacent layer and nonconnected interlayer structure ensures that the activation state of each neuron is independent of the other neurons as follows:

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{m} P\left(h_j\middle|\mathbf{v}\right) \tag{1}$$

Due to this independence, the operation of input vectors can be used as a matrix operation while updating the values of the hidden neurons. This process can increase the speed of training, making it suitable for online predictions.

In a binary RBM, the weights between the neurons of the visible layer and the hidden layer are undirected and are denoted as $\mathbf{w} = \{w_{ij}\}$ $(i = 1,\ 2, \ldots,\ n;\ j = 1,\ 2, \ldots,\ m)$, where w_{ij} denotes the weight between the hidden neuron (h_j) and the visible neuron (v_i). Therefore, the visible and hidden neurons have their biases, denoted by vectors $\mathbf{b}$ and $\mathbf{c}$, as shown in Fig. 1.

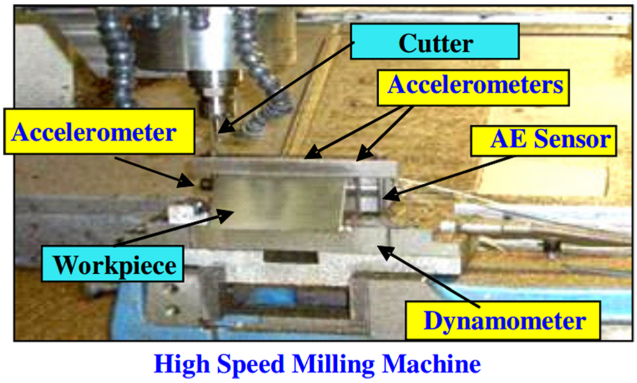


Fig. 5 The milling machine testbed

The probability of the neurons in the RBM being 0 or 1 is defined based on the energy of the neuron. Taking a hidden neuron (h_j) as an example, this energy is given as

$$E_j = \sum_{i=1}^{n} w_{ij} v_i + c_j \tag{2}$$

Similarly, the energy of visible neuron (v_i) is calculated as

$$E_i = \sum_{j=1}^{m} w_{ij} h_j + b_i \tag{3}$$

To simplify the subsequent discussion, we denote $\theta = \{\mathbf{w}, \mathbf{b}, \mathbf{c}\}$ as the set of parameters in the RBM. The architecture of the RBM is explicit, including visible and hidden neurons named $\mathbf{v}$ and $\mathbf{h}$, respectively, and a set of parameters (θ), including the weights and biases of the neurons.

The probability of the hidden neuron having a value of 1 can be calculated as follows:

$$P\left(h_j = 1|\mathbf{v}\right) = \sigma(j) = \frac{1}{1 + e^{-E_j}} \tag{4}$$

Similarly, in terms of visible neurons,

$$P\left(v_j = 1|\mathbf{h}\right) = \sigma(i) = \frac{1}{1 + e^{-E_i}} \tag{5}$$

Considering all the neurons, the probability that the model assigns to a visible vector ($\mathbf{v}$) is obtained by summing over all of the possible hidden vectors as follows:

$$\begin{aligned} P(\mathbf{v};\theta) &= \sum_{\mathbf{h}} P(\mathbf{v},\mathbf{h};\theta) = \frac{1}{\sum_{\mathbf{h}}\sum_{\mathbf{v}} \exp(-E(\mathbf{v},\mathbf{h};\theta))} \sum_{\mathbf{h}} (-E(\mathbf{v},\mathbf{h};\theta)) \\ &= \frac{1}{\sum_{\mathbf{h}}\sum_{\mathbf{v}} \exp(-E(\mathbf{v},\mathbf{h};\theta))} \sum_{\mathbf{h}} \exp\left(\mathbf{v}^T\mathbf{w}\mathbf{h} + \mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{h}\right) \end{aligned} \tag{6}$$

Table 3 Extracted features

Cutting force (x, y, z dimensions)	Vibration (x, y, z dimensions)	Acoustic emission
Maximum	Maximum	Maximum
Minimum	Minimum	Minimum
Mean	Mean	Mean
Standard deviation	Standard deviation	Standard deviation

To determine the values of the neurons, a random number (u) ranging between 0 and 1 is compared to $P(h_j = 1)$ to determine the value of h_j using the following criteria:

$$h_j = \begin{cases} 1, P\left(h_j = 1\right) \geq u \\ 0, P\left(h_j = 1\right) < u \end{cases} \tag{7}$$

The RBM training involves finding the optimum θ that fits best the training dataset. This value can be determined by performing a stochastic gradient descent on the negative log-likelihood probability of the training data. Considering the visible layer, the gradient of the negative log probability can be obtained from Eq. (6) as

$$\begin{cases} \dfrac{\partial \ln P(\mathbf{v};\theta)}{\partial w_{ij}} = \left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{model}} \\ \dfrac{\partial \ln P(\mathbf{v};\theta)}{\partial \mathbf{c}} = \left\langle h_j \right\rangle_{\text{data}} - \left\langle h_j \right\rangle_{\text{model}} \\ \dfrac{\partial \ln P(\mathbf{v};\theta)}{\partial \mathbf{b}} = \left\langle v_i \right\rangle_{\text{data}} - \left\langle h_i \right\rangle_{\text{model}} \end{cases} \tag{8}$$

where $\langle \bullet \rangle_{\text{data}}$ denotes an expectation with respect to the data distribution and $\langle \bullet \rangle_{\text{model}}$ denotes an expectation with respect to the distribution defined by the model. The former term is called the positive phase; it is calculated from given data and increases the probability of training data. The latter term is called the negative phase; it is calculated by model data and decreases the probability of samples generated by the model. However, the expectation $\langle \bullet \rangle_{\text{model}}$ cannot be easily computed. Gibbs sampling is used to obtain the approximation to the gradient. As shown in Fig. 2, Gibbs sampling starts from the given visible data ($\mathbf{v}^{(0)}$) to compute the initial expectation of hidden neurons $\mathbf{h}^{(0)}$; then, $\mathbf{h}^{(0)}$ is used to compute the first expectation of visible data ($\mathbf{v}^{(1)}$). Theoretically, infinite steps are needed to obtain an accurate value of $\langle \bullet \rangle_{\text{model}}$; however, in practice, a few steps have been shown to yield a comparable result, referred to as contrastive divergence (CD) [25]. A single sampling is typically adequate; then, $\langle \bullet \rangle_{\text{model}}$ is replaced by one Gibbs sampling.

Using Eq. (8) and single-sampled CD learning, the update rule of the parameter θ is given as follows:

$$\begin{cases} \mathbf{w}(t) = \mathbf{w}(t{-}1) + \varepsilon_{\mathbf{w}}\left(\left\langle v_i^{(0)} h_j^{(0)} \right\rangle - \left\langle v_i^{(1)} h_j^{(1)} \right\rangle\right) \\ \mathbf{c}(t) = \mathbf{c}(t{-}1) + \varepsilon_{\mathbf{c}}\left(\left\langle h_j^{(0)} \right\rangle - \left\langle h_j^{(1)} \right\rangle\right) \\ \mathbf{b}(t) = \mathbf{b}(t{-}1) + \varepsilon_{\mathbf{b}}\left(\left\langle v_j^{(0)} \right\rangle - \left\langle v_j^{(1)} \right\rangle\right) \end{cases} \tag{9}$$

where $\varepsilon_{\mathbf{w}}$, $\varepsilon_{\mathbf{c}}$, and $\varepsilon_{\mathbf{b}}$ are the learning rates of the weight, hidden bias, and visible bias, respectively.

One time of the update of θ is called an epoch; a total of ten epochs were adopted in this study, as this number of epochs has demonstrated optimal performance in previous experiments.

As shown in Fig. 3, stacking RBMs form a DBN layer-by-layer. The training procedure is a repeat of the RBM training by the CD algorithm: each hidden layer is trained using the activation probabilities of the hidden layer of the lower RBM as the input training data, and its output data are used as the training data of the upper RBM. After N hidden layers have been trained, an output layer is added for the real-valued output, and a fine-tuning process is performed using the back-propagation algorithm with the training data as output. This procedure is demonstrated by the dotted arrow in

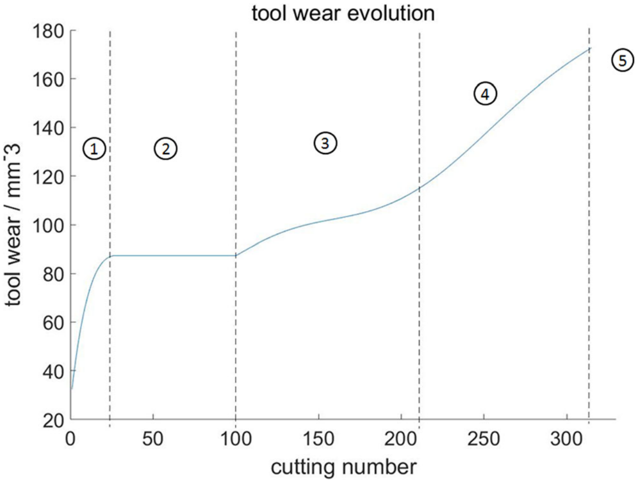


Fig. 6 Five different wear rates in the cutting process

Table 4 A comparison of the MSEs

Mean-squared error (total)						
Hidden unit number	10	11	12	13	14	15
DBN	0.017914	0.013705	0.0091717	0.0081048	0.006921	0.010889
ANN	0.023739	0.01447	0.0071075	0.024372	0.021933	0.1317
SVR	0.007775					

Fig. 3. Four DBN layers are used in this study, and for simplicity, every hidden layer is set with the same number of hidden neurons, as was used in [26].

2.2 Compared algorithms

In this study, two artificial intelligence algorithms are also compared with the DBN to demonstrate the fitness of the DBN in predicting tool wear. The two other algorithms are least-squares support vector regression (LS-SVR) and the ANN. These algorithms are briefly described below.

Originally developed by Vapnik [27], a SVM is used to construct a hyperplane in high-dimensional space and to classify data. SVR is a popular application of SVM techniques [11, 27]. SVR is typically used to estimate the relationship between input and output variables. LS-SVR training only requires the solution for a set of linear questions instead of the complex quadratic problems involved in standard SVR. LS-SVR is considered in the comparison because it is a powerful tool in tool wear prediction. In this study, the kernel function of SVR is the radial basis function (RBF)

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-p\|\mathbf{x}-\mathbf{x}_i\|), \quad p > 0 \tag{10}$$

where p is the tuning factor and is set to 0.1.

The ANN, a widely used algorithm, is also considered in the comparison. The ANN is an imitation of the human brain, which is composed of neurons [28]. An ANN is generally composed of three elements: the values of the neurons, the weights connecting the neurons from different layers, and the active function, which is a function used to convert the input of every neuron to an output, considering the weights connecting them. When training ANNs, back-propagation, a learning algorithm, is widely used to reduce the training error. Thus, a FFBP network is applied here. Figure 4 illustrates the architecture of the FFBP-NN. The visible layer accepts the input data, which are denoted as f_i. These data are conveyed from the visible layer to the hidden layer and then the output layer. The neurons are computed using a sigmoid function, in a similar manner as in the DBN. The output data are compared with a label for each record and then used in the BP process to adjust the network. In contrast to the DBN, the ANN only has one hidden layer, which represents its weakness in learning complex nonlinear relationships. Six hundred training data points are used to train the network. One traversal of all the training data is referred to as an epoch, and ten epochs are adopted as an empirical choice to achieve a balance between runtime and accuracy.

There is currently no standard or well-accepted method for selecting the number of hidden neurons [29, 30]. As part of this study, experiments are conducted with the number of hidden neurons (ranging from 10 to 25 as a typical range), which include the most possible algorithm results, along with a comparison of the corresponding ANNs. The results are shown in Figs. 7 and 8. The accuracy of the network fluctuated considerably with variations in the number of hidden neurons.

3 Experimental setup

The data used in this paper were obtained from Li et al. [31], which are part of the "prognostic data challenge 2010" dataset from the Prognostics and Health Management (PHM) Society. Experiments were performed on a high-speed CNC milling machine (Röders Tech RFM760) with a cutter spindle speed of 10,400 RPM, a feed rate of 1555 mm/min, a y cut depth (radial) of 0.125 mm, and a z cut depth (axial) of 0.2 mm. These values are shown in Table 1. The cutter material was high-speed steel, while the workpiece material was stainless steel.

Table 5 A comparison of the R^2 values

Coefficient of determination						
Hidden unit number	10	11	12	13	14	15
DBN	0.97117	0.97794	0.98524	0.98696	0.98886	0.98248
ANN	0.96179	0.97671	0.98856	0.96077	0.9647	0.78804
SVR	0.96246					

Table 6 A comparison of the training times

Training time (s)						
Hidden unit number	10	11	12	13	14	15
DBN	0.087351	0.10028	0.08858	0.11285	0.11994	0.1041
ANN	0.062962	0.066766	0.063753	0.061384	0.066667	0.062798
SVR	0.25375					

The workpiece was prepared through face milling to avoid the effects of hard regions in the original skin layer. The sensor installation layout is shown in Fig. 5. A three-component dynamometer was mounted between the workpiece and machining table to record the cutting force. Three piezo accelerometers were also mounted to the workpiece to measure the vibrations from the machine tool during the cutting period in the *x*-, *y*-, and *z*-directions [31]. Besides, an AE sensor was used to monitor the high-frequency stress waves generated by the cutting process. These signals from the different sensors represent the data source of the models. A detailed description of the meaning of these data is provided in Table 2.

The sensor signals were acquired at a sampling rate of 50 kHz by a DAQ board. Every 15 s, data was called a record, including approximately 200,000 signals. Each cutting test included 315 records, and there were three individual cutting tests. Two of the tests were used as training data, and the remainder was used as testing data. The cutters were considered to be worn to a specific stage at the end of each cutting test, which is a criterion to end the test. The total size of the data was approximately 8 GB.

For the data pre-processing, four statistical features were extracted from the raw data, as shown in Table 3. Generally, there are five stages in the wear procedure, called initial wear, slight wear, moderate wear, severe wear, and worn-out [32]. In Fig. 6, they are marked as stages 1 to 5, and in this study, the experiment was stopped when tool was worn out so there were no records in stage 5. It is demonstrated in previous research that the wear rate differs in different processing stages [33], which is also shown in Fig. 6. Considering this, a time stamp was added as the last feature. Furthermore, the input of RBM needs to range from 0 to 1, statistic features and the time stamp were normalized, and these were the input of DBN. To train the DBN, a label is needed for each record, which is the wear of the cutter. The wear data were normalized as well.

4 Results and discussion

Three predictive models were used to make the comparison, including the SVR, the ANN, and the DBN. For simplicity, certain records were omitted. Six hundred records were used as training data, and 300 records were used as testing data.

Two accuracy evaluation standards were adopted: the coefficient of determination (expressed as R^2) and the mean-squared error (MSE). The time required for training and predicting was an important criterion in this paper, as low runtimes are required for online prediction.

Because the number of hidden neurons may influence the performance of networks, experiments were conducted with varying numbers of neurons in the hidden layer on both the ANNs and DBNs. There is no hidden neuron in SVR; thus, the SVR results do not fluctuate. Ten iterations were performed for the ANN and DBN.

In this study, the computation was performed using MATLAB (MathWorks, 2016). Tables 4, 5, 6, and 7 illustrate the prediction results in data form, whereas Figs. 7, 8, 9, and 10 expressed the data as lines. The DBNs achieved a minimum MSE of 0.0069, whereas the ANNs and SVR achieved minimum values of 0.0071 and 0.0078, respectively. For R^2, the DBNs achieved a maximum value of 0.9889, whereas the ANNs and SVR achieved values of 0.9886 and 0.96246, respectively. There is no significant difference in accuracy prediction between the best performances of these three algorithms. However, because DBNs have more hidden layers than do ANNs, additional time is required to train the networks. Both the DBNs and ANNs have shorter training times than do SVR because of their superior architectures. The comparison is clearly visualized in Fig. 9. It shows that SVR required more time than the other two algorithms, and training with DBNs was slightly more time-consuming

Table 7 A comparison of the prediction times

Prediction time (s)						
Hidden unit number	10	11	12	13	14	15
DBN	0.00022904	0.00096396	0.00025117	0.001882	0.00087253	0.0018945
ANN	0.0013704	0.0015722	0.0024428	0.00052961	0.00034548	0.00024027
SVR	0.042562					

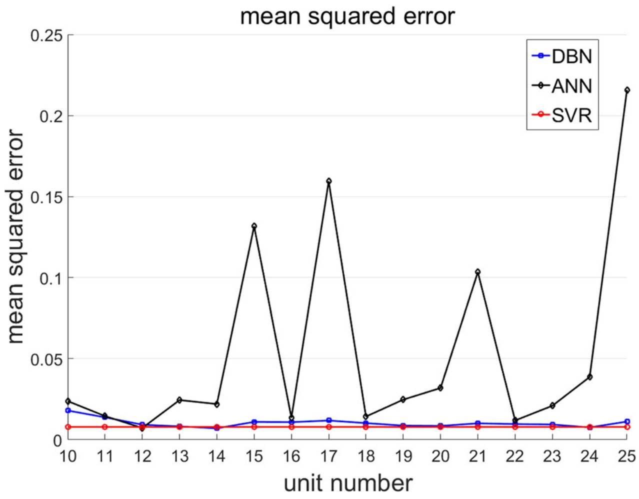


Fig. 7 A comparison of the MSEs

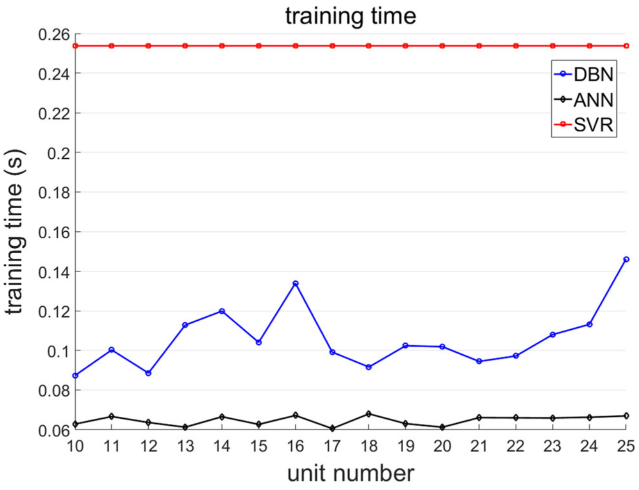


Fig. 9 A comparison of the training times

than that with ANNs. However, the difference in time is acceptable because for a particular system, the working mode is fixed, and the network can be trained off-line. Therefore, the time cost can be ignored. Regarding the prediction time shown in Table 7, DBNs and ANNs require approximately the same amount of time, and they are considerably more efficient than SVR, with a 60% shorter prediction time. Figure 10 illustrates this trend in a visual manner; namely, the DBN and ANN results are extremely similar, whereas the results for SVR are considerably higher. Therefore, both DBNs and ANNs can perform prompt online prediction.

However, considering the sensitivity of network, parameters of the networks can have an influence on predicting result. We changed the number of hidden neurons, the number of training epochs, and the learning rate of DBNs and ANNs and found that the accuracy of ANNs was strongly influenced by the number of hidden neurons while that of DBNs was relatively stable. As shown in Fig. 7, the MSE of the ANNs fluctuated with the increase in hidden neurons while DBNs exhibit a comparatively stable performance. Compared with ANNs, when networks with hidden neurons differ from 10 to 25, the DBNs decrease the MSE by 85.5% on average, which is much different from the previous comparison of the best results. Similarly, as shown in Fig. 8, the R^2 of ANNs also has violent fluctuation, which leads to a decrease in its performance. Regrettably, previous studies have shown that the method for determining the optimal number of hidden neurons in ANNs has yet to be validated [28, 34], making it difficult to avoid fluctuations in ANN. That means, it is easier to construct a DBN than an ANN for a particular system; namely, DBN has a better portability than ANN. Therefore, considering the accuracy and stability, DBN performed much better than the other two methods.

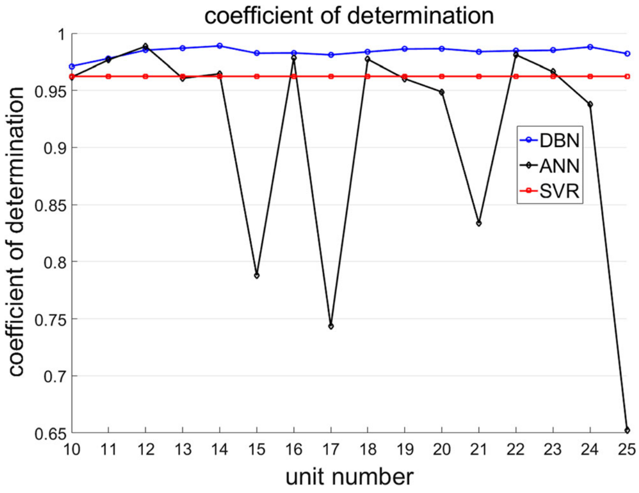


Fig. 8 A comparison of the R^2 values

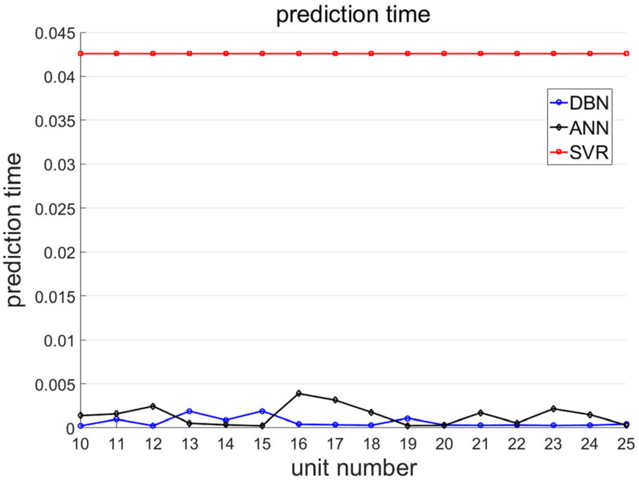


Fig. 10 A comparison of the prediction times

5 Conclusions and future work

This paper applied DBN to the prediction of cutting tool wear with multiple signals and compared SVR, DBNs, and ANNs for cutter wear prediction. The performances of these methods were evaluated by statistical standards, including the coefficient of determination and MSE, along with the training and prediction times. Six hundred records were used to train the network, and 300 records were used for testing. The ANN contained a single hidden layer, whereas the DBN contained two layers. The ANN and DBN were iterated ten times using varying numbers of hidden neurons. As a result, the DBN and ANN exhibited lower runtimes than did SVR. All three of the algorithms achieved high accuracy, but the DBN was more stable than the ANN. In summary, the DBN featured a low runtime, high accuracy, and high stability. These characteristics make it an ideal choice for tool wear prediction, particularly for online prediction with multi-sensor data.

With the development of machine learning algorithms, artificial intelligent methods can be applied to traditional machining technology. Therefore, multiple sensors are always used to collect more specific information on the cutter state, indicating that more powerful tools are needed to handle massive data. By using the DBN for tool wear prediction, this study demonstrated that the DBN achieved good performance in terms of accuracy, stability, and speed. Considering the need for online prediction, the DBN is a suitable future choice. In subsequent studies, experiments will be conducted to compare the performance of different types of deep learning networks, and practical applications of machine learning will be developed.

Funding Information This research was supported by the National Science Foundation of China (Grant No. 51605464), National Basic Research Program of China (973 Program) (2014CB049500), and Research on the Major Scientific Instrument of National Natural Science Foundation of China (61727809).

References

1. Mobley RK (1990) An introduction to predictive maintenance. Van Nostrand Reinhold, New York
2. Malekian M, Park SS, Jun MBG (2009) Tool wear monitoring of micro-milling operations. J Mater Process Technol 209(10):4903–4914
3. Hu C, Youn BD, Kim T (2012) Semi-supervised learning with cotraining for data-driven prognostics. IEEE Conference on Prognostics and Health Management (PHM), Denver, CO, June 18–21, pp. 1–10
4. Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, Helu M (2015) Cloud-enabled prognosis for manufacturing. CIRP Ann 64(2):749–772
5. Daigle MJ, Goebel K (2013) Model-based prognostics with concurrent damage progression processes. IEEE Trans Syst Man Cybern Syst 43(3):535–546
6. Dong M, He D (2007) Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis. Eur J Oper Res 178(3):858–878
7. Si XS, Wang W, Hu CH, Chen MY, Zhou DH (2013) A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. Mech Syst Signal Process 35(1–2):219–237
8. Niaki FA, Michel M, Mears L (2016) State of health monitoring in machining: extended Kalman filter for tool wear assessment in turning of IN718 hard-to-machine alloy. J Manuf Process 24:361–369
9. Rizal M, Ghani JA, Nuawi MZ, Haron CHC (2017) An embedded multi-sensor system on the rotating dynamometer for real-time condition monitoring in milling. Int J Adv Manuf Technol:1–13
10. Wang D, Peter WT (2015) Prognostics of slurry pumps based on a moving-average wear degradation index and a general sequential Monte Carlo method. Mech Syst Signal Process 56:213–229
11. Benkedjouh T, Medjaher K, Zerhouni N, Rechak S (2015) Health assessment and life prediction of cutting tools based on support vector regression. J Intell Manuf 26(2):213–223
12. Zhang C, Zhang H (2016) Modelling and prediction of tool wear using LS-SVM in milling operation. Int J Comput Integr Manuf 29(1):76–91
13. Li N, Chen Y, Kong D, Tan S (2017) Force-based tool condition monitoring for turning process using v-support vector regression. Int J Adv Manuf Technol 91(1–4):351–361
14. Siddhpura A, Paurobally R (2013) A review of flank wear prediction methods for tool condition monitoring in a turning process. Int J Adv Manuf Technol:1–23
15. Samanta B, Nataraj C (2009) Use of particle swarm optimization for machinery fault detection. Eng Appl Artif Intell 22(2):308–316
16. Lee SS (2010) Tool condition monitoring system in turning operation utilizing wavelet signal processing and multi-learning ANNs algorithm methodology. Int J Eng Res Innov 49
17. D'Addona DM, Ullah AMMS, Matarazzo D (2015) Tool-wear prediction and pattern-recognition using artificial neural network and DNA-based computing. J Intell Manuf:1–17
18. Coppel R, Abellan-Nebot JV, Siller HR, Rodriguez CA, Guedea F (2016) Adaptive control optimization in micro-milling of hardened steels—evaluation of optimization approaches. Int J Adv Manuf Technol 84(9–12):2219–2238
19. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
20. Bengio Y (2013) Deep learning of representations: looking forward, in statistical language and speech processing. Springer, pp 1–37
21. Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
22. Fu Y, Zhang Y, Qiao H, Li D, Zhou H, Leopold J (2015) Analysis of feature extracting ability for cutting state monitoring using deep belief networks. Procedia Cirp 31:29–34
23. Tamilselvan P, Wang P (2013) Failure diagnosis using deep belief learning based health state classification. Reliab Eng Syst Saf 115:124–135
24. Jia F, Lei Y, Lin J, Zhou X, Lu N (2016) Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mech Syst Signal Process 72:303–315
25. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14:1771–1800

26. Mohamed A, Dahl GE, Hinton G (2012) Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process 20(1):14–22
27. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
28. Wang SC (2003) Artificial neural network. In: Interdisciplinary computing in java programming. Springer US, pp 81–100
29. Wu D, Jennings C, Terpenny J, Gao RX, Kumara S (2017) A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests. J Manuf Sci Eng 139(7):071018
30. Sheela KG, Deepa SN (2013) Review on methods to fix number of hidden neurons in neural networks. Math Probl Eng 2013(1):11
31. Li X, Lim BS, Zhou JH, Huang S, Phua SJ, Shaw KC, Er MJ (2009) Fuzzy neural network modelling for tool wear estimation in dry milling operation. In: Annual conference of the prognostics and health management society, pp 1–11
32. Ertunc HM, Oysu C (2004) Drill wear monitoring using cutting force signals. Mechatronics 14(5):533–548
33. Tobon-Mejia DA, Medjaher K, Zerhouni N (2011) CNC machine tool health assessment using dynamic bayesian networks. IFAC Proc 44(1):12910–12915
34. Toghyani S, Ahmadi MH, Kasaeian A, Mohammadi AH (2016) Artificial neural network, ANN-PSO and ANN-ICA for modelling the Stirling engine. Int J Ambient Energy 37(5):456–468