

Article



Manufacturing Quality Prediction Using Intelligent Learning Approaches: A Comparative Study

Yun Bai¹, Zhenzhong Sun¹, Jun Deng¹, Lin Li², Jianyu Long¹ and Chuan Li^{1,*}

- ¹ School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China; baiyun@dgut.edu.cn (Y.B.); sunzz@dgut.edu.cn (Z.S.); dengj@dgut.edu.cn (J.D.); longjy@dgut.edu.cn (J.L.)
- ² School of Computer Science and Network Security, Dongguan University of Technology, Dongguan 523808, China; linli@ieee.org
- * Correspondence: chuanli@dgut.edu.cn or c.li@ismmm.org

Received: 26 November 2017; Accepted: 27 December 2017; Published: 30 December 2017

Abstract: Under the international background of the transformation and promotion of manufacturing, the Chinese government proposed the "Made in China 2025" strategy, which focused on the improvement of a quality-based innovation ability. Moreover, predicting manufacturing quality is one of the crucial measures for quality management. Accurate prediction is closely related to the feature learning of manufacturing processes. Therefore, two categories of intelligent learning approaches, i.e., shallow learning and deep learning, are investigated and compared for manufacturing quality prediction in this paper. Specifically, the feed forward neural network (FFNN) with one hidden layer and the least squares support vector machine (LSSVM) with no hidden layers are selected as the representatives for shallow learning, and the deep restricted Boltzmann machine (DRBM) and the stack autoencoder (SAE) are chosen as the representatives for deep learning. The manufacturing data is collected from a competition about manufacturing quality control in the Tianchi Data Lab of China. The experiments show that the deep framework overwhelms the shallow architecture in terms of mean absolute percentage error, root-mean-square error, and threshold statistics. In addition, the prediction results also indicate that the performances depend on the length of the training data. That is, the bigger the sample size is, the better the performance is.

Keywords: manufacturing quality prediction; made in China 2025; intelligent learning; comparative study

1. Introduction

To achieve the transformation and upgrade of China's manufacturing, the "Made in China 2025" plan [1] proposed a basic guideline with innovation-driven, quality first, green development, structure optimization, and talent-oriented objectives. Therefore, quality, as the lifeline in manufacturing, has attracted the attention of manufacturers and researchers. To control and improve manufacturing quality, many techniques are implemented into the manufacturing process. Among them, manufacturing quality prediction, as one of the effective ways to control and improve manufacturing quality, has been developed using various data mining techniques.

Statistical quality control [2] based on cause–effect relationships, e.g., linear regression [3], non-linear regression [4], inference learning [5], and expert systems [6], has been widely used to assess the quality performance of manufacturing processes. The successful application of these approaches is attributed to certain stable or constant production processes, which thus makes them unsuitable for the fast-increasing complexity and high-dimensionality of modern manufacturing. To address this issue, artificial intelligence (AI) is stepping into the academic field of these researchers due to its self-learning ability without taking into account manufacturing processes [7–10]. Artificial neural networks (ANNs) and machine learning (ML) are two typical representatives of AI techniques,

and have achieved successful application in manufacturing quality prediction, e.g., self-organizing neural networks [11], back propagation neural networks (BPNNs) [12], radial basis function neural networks [13], probability neural networks [14], support vector machines (SVMs) [15], and extreme learning machines [16]. Affected by multiple parameters from multi-stage manufacturing processes, ANN and ML modeling exhibit feature learning difficulties and network calculation complexities due to their "shallow" architecture, i.e., the model has one hidden layer or none at all (a traditional ANN has one hidden layer and classical ML is based on a kernel function without a hidden layer). To improve prediction accuracy, it is thus imperative to enhance the feature learning capability using a "deep" representation technique.

In 2006, the deep learning (DL) technique was proposed [17] and it has become a hot research topic in AI. It has been proven to be effective for many fields, e.g., fault diagnosis [18], pattern recognition [19], and time series forecast [20,21]. Compared with the "shallow" models, DL has many hierarchical levels in a hidden layer, that is, the information representation is delivered from lower levels to higher levels, which makes the information representation more abstract and nonlinear for the higher levels. Through representations by the hierarchical levels, the "deeper" feature of multi-parameter manufacturing quality can be fitted by regression models sufficiently [22]. To our best knowledge, there has been little literature that has reported on applications for manufacturing quality prediction using the deep framework. Therefore, the DL technique can provide a possibility for manufacturing quality prediction.

This paper attempts to make a comparison of two feature learning patterns to investigate their performances for predicting manufacturing quality, including the feed forward neural network (FFNN), the least squares support vector machine (LSSVM), the deep restricted Boltzmann machine (DRBM), and the stack autoencoder (SAE). To reveal the feature learning capacity of the four models, two kinds of manufacturing data with multiple parameters are involved.

The rest of the paper is organized as follows. Section 2 introduces the FFNN, the LSSVM, the DRBM, and the SAE, respectively. Section 3 presents the application data. Section 4 gives the results with relevant discussion. Section 5 concludes this study.

2. Methodologies

As stated in the Introduction, both shallow and deep learning belong to the ANN and related machine learning algorithms. The significant difference is the structure depth (Figure 1), i.e., shallow learning includes only one hidden layer or none at all, and deep learning contains more than one hidden layer.



Figure 1. Different structure schematic diagrams for feature learning. (**a**) shallow learning framework, and (**b**) deep learning framework.

From Figure 1, one can clearly find that deep learning adopts a cascade of many hidden layers for feature extraction and transformation, and higher level features are derived from lower level features to form a hierarchical representation. Hence, deep learning can be regarded as an intensified version of shallow learning. To investigate learning performance, four typical approaches are introduced briefly in the following subsections, i.e., FFNN with one hidden layer, LSSVM with no hidden layers, and DRBM and SAE with many hidden layers.

2.1. Feed Forward Neural Network

The classical FFNN propagates inputs through a network with one input, one hidden, and one output layer to make a prediction (Figure 1a). In the FFNN architecture, the artificial neurons are organized as layers, the information strictly flows forward, and the errors of the network are propagated backwards. The expressions of the FFNN are as follows [23]

$$h_{j} = f_{hidden}(\sum_{i=1}^{m} w_{ij} x_{i}), y_{k} = f_{output}(\sum_{j=1}^{n} w_{jk} h_{j}),$$
(1)

where x_i (i = 1, 2, ..., m) represents the inputs, h_j (j = 1, 2, ..., n) represents the outputs of the hidden layer, y_k (k = 1, 2, ..., p) represents the outputs, w_{ij} and w_{jk} represent the weight matrix between two adjacent layers, respectively, and $f_{hidden}(.)$ and $f_{output}(.)$ are transfer functions in the hidden layer and the output layer, respectively. To update the weights w effectively, a back propagation algorithm (BP), a well-known method, is used for training the FFNN [24].

2.2. Least Squares Support Vector Machine

For a given dataset, the goal of the LSSVM for regression is to find an optimal relationship between inputs **x** and outputs **y** in the feature space $\mathbf{y} = \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{\varphi}(\mathbf{x}) + \mathbf{b}$ (Figure 1a), where $\boldsymbol{\varphi}(\mathbf{x})$ denotes the nonlinear mapping function, $\boldsymbol{\omega}$ is the weight vector, and **b** is the bias vector. Moreover, the objective function of the LSSVR is given by

$$\min J(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + \frac{\gamma}{2}\sum_{i=1}^{q}\boldsymbol{\xi}^{2}$$
(2)

where ξ is the error variance, and $\gamma > 0$ is the penalty coefficient.

Transforming this quadratic programming problem to its corresponding dual optimization problem and introducing the kernel function in order to achieve non-linearity yields an optimal regression function as [25]

$$\mathbf{y} = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \mathbf{b}$$
(3)

where *q* is the length of dataset, α_i is the Lagrange multiplier, and *k*(.) represents the kernel function.

Generally, the radial basis function (RBF) is chosen as the kernel function, and is given by

$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left[-\frac{\mathbf{x} - \mathbf{x}_i}{2\lambda^2}\right]$$
(4)

where λ is the kernel bandwidth.

2.3. Deep Restricted Boltzmann Machine

As introduced above, a DRBM is a stack of restricted Boltzmann machines (RBMs). After an RBM (Figure 2) has been learned, the activities of its hidden units can be used as the data for learning a higher-level RBM. Note that when l = 1, $h^\circ = x$ (also called visible nodes v in RBM).



Figure 2. Architecture of the restricted Boltzmann machine (RBM).

For an RBM, the energy function $E(\mathbf{v}, \mathbf{h} \mid \theta)$ taking consideration of the real data normalized into [0, 1] is given by [26]

$$E(\mathbf{v}, \mathbf{h} | \boldsymbol{\theta}) = -\sum_{m=1}^{V} \sum_{n=1}^{H} w_{mn} h_n \frac{v_m}{\sigma_m^2} - \sum_{m=1}^{V} \frac{(v_m - b_m)^2}{2\sigma_m^2} - \sum_{n=1}^{H} a_n h_n$$
(5)

where $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{a})$ is the parameter set, *w* is the symmetric weight between the hidden layers *l*-1 and *l*, **b** and **a** are their bias, σ is the standard deviation, and *V* and *H* denote the number of visible and hidden units, respectively.

The conditional probability distributions *P* are as follows:

$$P(h_n = 1 | \mathbf{v}) = Sigm(\sum_{m=1}^{H} w_{mn} \frac{v_m}{\sigma_m^2} + a_n),$$
(6)

$$P(v_m = v | \mathbf{h}) = Z(v | b_m + \sum_{n=1}^{H} w_{mn} h_n, \sigma_m^2)$$
(7)

where $Z(\mathbf{b}, \sigma)$ represents a Gaussian probability density function.

To solve these functions above, Hinton [27] proposed a contrastive divergence algorithm: (1) initialize **v** using the input data, and compute **h** according to the conditional probability distributions (Equation (6)); (2) obtain reconstruction state **v**' based on Equation (7) using **h**, and repeat Equation (6) to update the hidden nodes using **v**', obtaining **h**'. The update in a weight is given as follows:

$$\Delta w_{mn} = \eta \left(\left\langle \frac{v_m}{\sigma_m^2} h_n \right\rangle - \left\langle \frac{v'_m}{\sigma_m^{2'}} h'_n \right\rangle \right) \tag{8}$$

where η is the learning rate, and < . > refers to the expectation of the training data.

Then, one can stack several RBMs together into a DRBM following the structure in Figure 1b, and this process is continued until a prescribed number of hidden layers in the DRBM have been trained.

2.4. Stack Autoencoder Network

Training an SAE for regression is similar to the DRBM [28]: (1) from the lower to top layers (layer 1 to layer *l*), operate generative unsupervised learning layer-wise on the autoencoder (AE) (Figure 3); (2) from the top to lower layers (layer *l* to layer 1), fine-tune by a supervised learning method (back propagation algorithm) to tweak the parameter sets (**w**, **b**); and (3) from the hidden (top) to output layer, perform regression using the pre-training parameter sets (**w**, **b**).



Figure 3. Schematic diagram of the autoencoder (AE).

According to Figure 3, the AE model is described as follows briefly [29]. The purpose of the AE is to reconstruct inputs \mathbf{h}^{l-1} ($\mathbf{h}^{\circ} = \mathbf{Y}$) into new representations \mathbf{r} with a minimum reconstruction error

$$RE(\mathbf{h}^{l-1}, \mathbf{r}) = -\sum_{m=1}^{M} \left[\mathbf{h}_{m}^{l-1}\log(r_{m}) + (1 - \mathbf{h}_{m}^{l-1})\log(1 - r_{m})\right].$$
(9)

To solve this problem, the encoder $f_e(.)$ and decoder $f_d(.)$ functions are operated step-by-step until they achieve the optimal parameter sets (**w**, **b**) based on a minimal loss function (Equation (11)).

$$\mathbf{h}^{l} = f_{e}(\mathbf{h}^{l-1}) = Sigm(\mathbf{w}\mathbf{h}^{l-1} + \mathbf{b}), \mathbf{r} = f_{d}(\mathbf{h}^{l}) = Sigm(\mathbf{w}^{\mathrm{T}}\mathbf{h}^{l} + \mathbf{b}^{\mathrm{T}})$$
(10)

where *Sigm*(.) means the sigmoid activation function.

$$L(\mathbf{w}, \mathbf{b}) = \sum RE(\mathbf{h}^{l-1}, \mathbf{R})$$
(11)

3. Application to Manufacturing Quality Prediction

3.1. Dataset

The data is collected from a competition about manufacturing quality control in the Tianchi Data Lab of China (https://tianchi.aliyun.com/competition/gameList.htm). They have the same technique parameters (19 process parameters as shown in Table 1) with a different setting, thus the quality index (one key-quality index with range [0, 1] as shown in Figure 4) exhibits diversity in different batches. There are two kinds of samples, one is a small sample including 100 batches (total sample (19 + 1) × 100, as shown in Figure 4a), and the other is a big sample including 1000 batches (total sample (19 + 1) × 1000, Figure 4b). These data are divided into two categories, 80% for training and 20% for testing. Note that all the data have been desensitized.

 Table 1. Statistical information of the multiple parameters in different processes.

Multi-Parameter	Proc	Range	
Parameter 1 (x_1) Parameter 2 (x_2) Parameter 3 (x_3) Parameter 4 (x_4) Parameter 5 (x_5)	Material selection	Adjustable Non-adjustable	0, 1, 2, 3, 4, 5 0, 1 [7, 30.304] [7, 30.304] 0, 1

	Multi-Parameter	Pr	Range	
	Parameter 6 (\mathbf{x}_6)Parameter 7 (\mathbf{x}_7)Parameter 8 (\mathbf{x}_8)Parameter 9 (\mathbf{x}_9)Parameter 10 (\mathbf{x}_{10})Parameter 11 (\mathbf{x}_{11})Parameter 12 (\mathbf{x}_{12})Parameter 13 (\mathbf{x}_{13})Parameter 14 (\mathbf{x}_{14})	Manufacturing	Adjustable	$\begin{array}{c} 0, 1 \\ 342, 343 \\ 0.065, 0.075, 0.28 \\ 0.4, 1 \\ 1, 1.05, 1.3 \\ 0, 0.34, 0.35 \\ 0, 1, 2 \\ 0, 1, 3, 4 \\ 0, 1 \end{array}$
	Parameter 15 (x_{15}) Parameter 16 (x_{16}) Parameter 17 (x_{17}) Parameter 18 (x_{18}) Parameter 19 (x_{19})		Non-adjustable	4, 6 [1,110] 0, 1 0, 1, 2, 3, 4, 5 3, 3.1, 3.6
0 60 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 1 1		1 0.95 0.91 0.95 0.95	

 Table 1. Cont.

Figure 4. Manufacturing quality of different batches. (**a**) Small samples with 100 batches, and (**b**) big samples with 1000 batches.

100

0.8

 0.7^{4}

3.2. Model Development

20

40

(a)

60

Production batch

80

In this subsection, the investigated models are developed using the real manufacturing data. Note that all of the data are normalized into [0, 1] firstly according to the following equation

$$Normalization = \frac{data - data_{\min}}{data_{\max} - data_{\min}}$$
(12)

200

400

600

Production batch

(b)

800

1000

where $data_{min}$ and $data_{max}$ denote the minimum and maximum of each parameter in the dataset shown in Table 1. Then, the experimental method is applied to establish four models, and the details are listed in Table 2. The optimal model with the simplest structure is identified based on the paired *t*-test results [30] except for the LSSVM (it has no hidden layers). For convenience, the models of the DRBM and the SAE are named with a sequence number (18 models in total), e.g., 1 (l = 2, hidden nodes = 10), 2 (l = 2, hidden nodes = 20), 6 (l = 2, hidden nodes = 60), 7 (l = 3, hidden nodes = 10), 12 (l = 3, hidden nodes = 60), 13 (l = 3, hidden nodes = 10), and 18 (l = 3, hidden nodes = 60). All of the results in the following experiments are the best values of ten independent runs. In addition, the computation software is Matlab 2014 with the computation environment Intel Core i5-2450M CPU @2.50 GHz, and Memory 4.00 GB.

Model	Experimental Design
FFNN	Inputs = 19, output = 1, hidden nodes = $[4, 15]$, $f_{hidden}(.)$, $f_{output}(.)$ = 'Sigm', learning rate 0.05,
LSSVM	Inputs = 19, output = 1, γ and λ is optimized by 10-cross validation [25].
DRBM SAE	Inputs = 19, output = 1, $l = [2, 3, 4]$, hidden nodes = [10, 20, 30, 40, 50, 60] (the same number in each hidden layer), dropout 0.5, learning rate 1, and iteration 200.

Table 2. Experimental design of each approach.

FFNN: feed forward neural network; LSSVM: least squares support vector machine; DRBM: deep restricted Boltzmann machine; SAE: stack autoencoder.

3.3. Performance Criteria

Three criteria, mean absolute percentage error (MAPE), root-mean-square error (RMSE), and threshold statistics (TS), are employed to assess the forecasting performances. The definitions of the three criteria are listed as follows:

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{ob_i - pr_i}{ob_i} \right|, RMSE = \sqrt{\sum_{i=1}^{N} (ob_i - pr_i)^2 / N, \ TS_a = \frac{n_a}{B} \times 100$$
(13)

where *N* is the length of the prediction, ob_i and pr_i represent the *i*-th observation and prediction, respectively, and n_a is the number of data predicted having relative error in forecasting less than a%. In this paper, TS_a is calculated for five levels of 1%, 5%, and 10%.

Moreover, a Pearson correlation analysis [31] is employed to evaluate the correlation degree of the observation and prediction.

4. Results and Discussion

4.1. FFNN Results

Figure 5 plots the MAPE using the FFNN with different hidden nodes of two cases, respectively. As shown in Figure 5, the hidden nodes with the lowest MAPE are 10 (Case 1) and 4 (Case 2) respectively, regarding the control models based on the multiple comparison procedures [31]. Through carrying out the paired *t*-test, one can choose the simplest model's structure that is not significantly different from the control model so as to obtain better generalization ability. Table 3 gives the results of the paired *t*-test at the confidence level of 5%. Note that the models in Table 3 are remarked as the hidden nodes.



Figure 5. Experimental results using the FFNN with different hidden nodes. MAPE: mean absolute percentage error.

From Table 3, one can find that for Case 1, the models with 11-15 hidden nodes are considered not significantly different from the control model (Significance > 0.05), and those with 4–9 hidden nodes are significantly different from the control model (Significance < 0.05). Therefore, the model with 10 hidden nodes should be selected as the optimal model in this paper. The training time is 2.92 s.

For Case 2, the models with 4–5 hidden nodes are not significantly different, and the models with 6–15 are significantly different from the control model. The model with four hidden nodes should be selected as the optimal model in this paper. The training time is 4.04 s. Figure 6 shows the prediction results using the optimal FFNN for two cases, respectively.



Table 3	Pairod	t_tast rasult	ts of the	FENIN
Table 5.	raneu	t-test resur	is of the	FFININ.



Figure 6. Prediction results using the optimal FFNN. (a) 19-10-1 for Case 1, and (b) 19-4-1 for Case 2.

4.2. LSSVM Results

Figure 7 plots the prediction results using the LSSVM optimized by the 10-cross validation method for two cases, respectively. The training times of the two cases are 5.65 min and 12.22 min, respectively.



Figure 7. Prediction results using the optimal LSSVM. (a) $\gamma = 10.622$, $\lambda^2 = 50.018$ for Case 1, and (b) $\gamma = 9.565$, $\lambda^2 = 25.016$ for Case 2.

4.3. DRBM Results

Figure 8 plots the MAPE using the DRBM with different hidden structures of two cases, respectively. According to Figure 8, model numbers 7 (Case 1) and 8 (Case 2) have the lowest

MAPE, thus the hidden structures 10-10-10 and 20-20-20 are chosen as the control model for the paired *t*-test. Table 4 gives the results of the paired *t*-test at the confidence level of 5%.



Figure 8. Experimental results using the DRBM with different hidden structures.

As shown in Table 4, for Case 1, the control model is significantly different from the models 1–6, 9–10, 14, and 16, hence model 7 has the simplest structure. The training time is 3.08 s. For Case 2, the control model is not significantly different from models 12 and 13, hence model 8 has the simplest structure. The training time is 8.28 s. Figure 9 shows the prediction results using the optimal DRBM for two cases, respectively.

Sample	Control Model	Paired Model	Significance (Asymptotic)	Paired Model	Significance (Asymptotic)
		1	0.049	11	0.124
		2	0.021	12	0.190
		3	0.050	13	0.071
		4	0.035	14	0.011
Case 1	7	5	0.024	15	0.100
		6	0.018	16	0.084
		8	0.097	17	0.090
		9	0.007	18	0.052
		10	0.002		
		1	0.039	11	0.039
		2	0.005	12	0.120
		3	0.000	13	0.205
		4	0.002	14	0.000
Case 2	8	5	0.001	15	0.001
		6	0.003	16	0.007
		7	0.004	17	0.005
		9	0.015	18	0.001
		10	0.000		

Table 4. Paired *t*-test results of the DRBM.



Figure 9. Prediction results using the optimal DRBM. (a) 19-10-10-1 for Case 1, and (b) 19-20-20-20-1 for Case 2.

4.4. SAE Results

Figure 10 plots the MAPE using the SAE with different hidden structures of two cases, respectively. According to Figure 10, model numbers 9 (Case 1) and 2 (Case 2) have the lowest MAPE, thus the

hidden structures 30-30-30 and 20-20 are chosen as the control model for the paired *t*-test. Table 5 gives the results of the paired *t*-test at the confidence level of 5%.



Figure 10. Experimental results using the SAE with different hidden structures.

As shown in Table 5, the control models (model 9 for Case 1 and model 2 for Case 2) have the simplest structure following the selection principle aforementioned (The training times of the two cases are 6.19 s and 12.08 s, respectively). Figure 11 shows the prediction results using the optimal SAE for two cases, respectively.

Table 5.	Paired	<i>t</i> -test resu	lts of the	e SAE.

Sample	Control Model	Paired Model	Significance (Asymptotic)	Paired Model	Significance (Asymptotic)
		1	0.046	11	0.022
		2	0.040	12	0.110
		3	0.033	13	0.108
		4	0.037	14	0.044
Case 1	9	5	0.041	15	0.117
		6	0.046	16	0.015
		7	0.018	17	0.015
		8	0.034	18	0.014
		10	0.045		
		1	0.000	11	0.033
		3	0.000	12	0.025
		4	0.122	13	0.009
		5	0.021	14	0.013
Case 2	2	6	0.000	15	0.036
		7	0.033	16	0.001
		8	0.060	17	0.001
		9	0.010	18	0.000
		10	0.007		



Figure 11. Prediction results using the optimal SAE. (a) 19-30-30-30-1 for Case 1, and (b) 19-20-20-1 for Case 2.

4.5. Comparison Studies

As shown in Figures 6, 7, 9, and 11, one can find that: (1) the performances of the four models have clear differences, illustrating that the results are not related to the multi-parameter inputs, but related

to the inputs' feature learned by different patterns; (2) the predictions using the deep learning technique have smaller fluctuations than those using the shallow learning technique, illustrating that the parameters have little impact on the deep learning framework; and (3) all four models fail at the peak values, demonstrating that both shallow and deep learning have insufficient ability in peak information learning. To compare the models' performances from the quantification, a residual analysis and a statistical analysis are employed in the following text.

The residual analysis is plotted in Figure 12. From Figure 12a, one can find that the range of the residual errors is [-0.2, 0.2] of two cases, and there is 1 (accounting for 5%) prediction outlier (Case 1) and 15 (accounting for 7.5%) outliers (Case 2) shown in the triangle because the interval around the residual errors does not contain zero. This implies that the five residual errors caused by the unfortunate fitting, beyond the 95% confidence interval, account for 5% of the testing data. As shown in Figure 12b, one can find that the ranges of the residual errors are [-0.2, 0.15] and [-0.15, 0.2], respectively, and there is 1 (accounting for 5%) prediction outlier (Case 1) and 16 (accounting for 8%) outliers (Case 2). From Figure 12c, one can find that the ranges of the residual errors are [-0.15, 0.1]and [-0.1, 0.2], respectively, and there are 2 (accounting for 10%) prediction outliers (Case 1) and 11 (accounting for 5.5%) outliers (Case 2). As shown in Figure 12d, one can find that the ranges of the residual errors are [-0.15, 0.1] and [-0.1, 0.15], respectively, and there are 2 (accounting for 10%) prediction outliers (Case 1) and 12 (accounting for 6%) outliers (Case 2). Compared with the shallow learning architecture, the deep learning framework has smaller error fluctuations in the two cases, illustrating that deep learning has better performance over the entire testing dataset. However, the exhibition in the prediction outliers is different, that is, shallow learning is better than deep learning for small samples (Case 1) in terms of the number of the outliers, and deep learning is better than shallow learning for big samples (Case 2). This phenomenon can be attributed to the sample size, demonstrating that the feature learning ability of the deep technique is closely related to the sample size. That is, the bigger the sample size is, the better the performance is.

The evaluation criteria are summarized in Table 6. Note that PCC refers to Pearson correlation coefficient, and the labels ** and * represent 0.01 and 0.05 levels of significant correlation, respectively. As shown in Table 6, the statistical indexes of the two case applications demonstrate the following. First, in terms of the lowest MAPE and RMSE, the deep framework (DRBM and SAE) has a strong capacity for capturing the features of the manufacturing parameters and the quality sufficiently. However, the shallow architecture (FFNN and LSSVM) has a weaker capacity for feature learning and regression. Second, in terms of the highest TS, the error distributions of the deep framework are concentrated in the range of less than 5% (accounting for 90%) and 10% (accounting for 100%) for Case 1, and 5% (accounting for 92%, 92.5%) and 10% (accounting for 99.5%, 100%) for Case 2. However, the shallow architectures have good performance in TS₁ and worse performance in TS₅ and TS₁₀ compared with deep learning. Third, in terms of the PCC, the degree of correlation is higher using the shallow architecture.

C 1	NC 1.1	Performance					
Sample	Model	MAPE (%)	RMSE	TS ₁	TS ₅	TS ₁₀	PCC
	FFNN	3.323	0.044	30	75	95	0.261
C 1	LSSVM	2.939	0.035	20	85	95	0.316
Case 1	DRBM	2.242	0.031	20	90	100	0.504 *
	SAE	2.216	0.026	20	90	100	0.825 **
	FFNN	2.485	0.023	27.5	88.5	97.5	0.101
~ ^	LSSVM	2.361	0.024	26.5	91	99	0.192
Case 2	DRBM	2.306	0.019	26.5	92.5	99.5	0.348 *
	SAE	2.094	0.018	29	92	100	0.514 **

Table 6. Comparison of the prediction performances using different models.

PCC: Pearson's correlation coefficient; RMSE: root-mean-square error. Labels ** and * represent 0.01 and 0.05 levels of significant correlation, respectively.



Figure 12. Residual analysis of different models for the two cases. (a) FFNN; (b) LSSVM; (c) DRBM; and (d) SAE.

Additionally, although deep learning overwhelms shallow learning according to Table 6, the network complexity and computing burden increases. Therefore, the paired *t*-test is also applied for evaluating the significant difference to investigate its feasibility. Table 7 gives the significant differences of the four models at the 5% level. As shown in Table 7, one can find that shallow learning is significantly different from deep learning, and the two sets of models (the FFNN and the LSSVM,

and the DRBM and the SAE) have no significant difference. Therefore, the deep framework can be regarded as an effective approach for multi-parameter manufacturing quality prediction.

Sample	Paired Model	Significance (Asymptotic)	Paired Model	Significance (Asymptotic)
	FFNN-LSSVM	0.522	LSSVM-DRBM	0.017
Case 1	FFNN-DRBM	0.024	LSSVM-SAE	0.072
	FFNN-SAE	0.041	DRBM-SAE	0.943
	FFNN-LSSVM	0.093	LSSVM-DRBM	0.001
Case 2	FFNN-DRBM	0.029	LSSVM-SAE	0.004
	FFNN-SAE	0.000	DRBM-SAE	0.541

Table 7. Paired *t*-test results between each model.

In conclusion, according to the qualitative analysis and the quantitative analysis, deep feature learning is beneficial to explore sophisticated relationships between multiple parameters of manufacturing and quality, and display better prediction capacity for manufacturing quality. Moreover, sample size is a vital factor affecting the deep framework's performance.

5. Conclusions

The capability of shallow and deep learning to predict manufacturing quality is tested and compared in this paper. The candidates include the FFNN with one hidden layer, the LSSVM with no hidden layers, the DRBM, and the SAE. For this purpose, the trial and error method is adopted to select the optimal model with the simplest structures (except for the LSSVM), which are specified by the paired *t*-test results. Two cases, i.e., small samples (100 batches) and big samples (1000 batches), are investigated. The comparison of the model results has shown that: (1) the performances of the deep framework consisting of two or three hidden layers are better than those of the shallow architectures in terms of the MAPE, the RMSE, the TS, and the PCC criteria; (2) the performances of the deep framework depend on the sample size in terms of the number of the prediction outliers, i.e., the bigger the sample size is, the better the performance is; and (3) the deep framework and the shallow architecture are significantly different statistically. Based on the findings of this study, it can be stated that the deep learning techniques considered can be successfully applied to establish accurate manufacturing prediction models, especially for big data. In a future study, the authors will focus on the popularization and application of the deep learning techniques in other manufacturing enterprises.

Acknowledgments: This work is supported in part by the National Natural Science Foundation of China (51775112), the Postdoctoral Science Foundation of China (2016M602459), the Research Program of Higher Education of Guangdong (2016KZDXM054), and the Research Start-Up Funds of DGUT (GC300501-08).

Author Contributions: Yun Bai proposed the research and wrote the paper. Zhenzhong Sun and Jun Deng provided the datasets and designed the experiments. Lin Li and Jianyu Long evaluated the modeling performances. Chuan Li, as corresponding author, revised the paper technically.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. The State Council of China. *Made in China* 2025; The State Council of China: Beijing, China, 2015.
- 2. Montgomery, D.C. Statistical Quality Control: A Modern Introduction, 7th ed.; Wiley: Hoboken, NJ, USA, 2013.
- 3. Hao, L.; Bian, L.; Gebraeel, N.; Shi, J. Residual life prediction of multistage manufacturing processes with interaction between tool wear and product quality degradation. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 1211–1224. [CrossRef]
- 4. Li, D.C.; Chen, W.C.; Liu, C.W.; Lin, Y.S. A non-linear quality improvement model using SVR for manufacturing TFT-LCDs. *J. Intell. Manuf.* 2012, 23, 835–844. [CrossRef]
- 5. Nada, O.A.; Elmaraghy, H.A.; Elmaraghy, W.H. Quality prediction in manufacturing system design. *J. Manuf. Syst.* **2006**, *25*, 153–171. [CrossRef]

- 6. Hosein, K.M.; Karim, A.; Saeed, K.S.M. Development of a new expert system for statistical process control in manufacturing industry. *Iran. Electr. Ind. J. Qual. Product.* **2013**, *2*, 29–40.
- Chamkalani, A.; Chamkalani, R.; Mohammadi, A.H. Hybrid of two heuristic optimizations with LSSVM to predict refractive index as asphaltene stability identifier. *J. Dispers. Sci. Technol.* 2014, 35, 1041–1050. [CrossRef]
- Lieber, D.; Stolpe, M.; Konrad, B.; Deuse, J.; Morik, K. Quality prediction in interlinked manufacturing processes based on supervised and unsupervised machine learning. *Procedia CIRP* 2013, 7, 193–198. [CrossRef]
- 9. Bustillo, A.; Correa, M. Using artificial intelligence to predict surface roughness in deep drilling of steel components. *J. Intell. Manuf.* **2012**, *23*, 1893–1902. [CrossRef]
- 10. Yu, Y.; Choi, T.M.; Hui, C.L. An intelligent quick prediction algorithm with applications in industrial control and loading problems. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 276–287. [CrossRef]
- 11. Chen, W.C.; Tai, P.H.; Wang, M.W.; Deng, W.J.; Chen, C.T. A neural network-based approach for dynamic quality prediction in a plastic injection molding process. *Expert Syst. Appl.* **2008**, *35*, 843–849. [CrossRef]
- 12. Zhang, E.; Hou, L.; Shen, C.; Shi, Y.; Zhang, Y. Sound quality prediction of vehicle interior noise and mathematical modeling using a back propagation neural network (BPNN) based on particle swarm optimization (PSO). *Meas. Sci. Technol.* **2016**, *27*, 015801. [CrossRef]
- 13. Wannas, A.A. RBFNN model for prediction recognition of tool wear in hard turing. *J. Eng. Appl. Sci.* **2012**, *3*, 780–785.
- Li, J.; Kan, S.J.; Liu, P.Y. The study of PNN quality control method based on genetic algorithm. *Key Eng. Mater.* 2011, 467–469, 2103–2108. [CrossRef]
- 15. Liu, G.; Gao, X.; You, D.; Zhang, N. Prediction of high power laser welding status based on PCA and SVM classification of multiple sensors. *J. Intell. Manuf.* **2016**. [CrossRef]
- 16. Sun, H.; Yang, J.; Wang, L. Resistance spot welding quality identification with particle swarm optimization and a kernel extreme learning machine model. *Int. J. Adv. Manuf. Technol.* **2016**. [CrossRef]
- 17. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
- Li, C.; Sanchez, R.V.; Zurita, G.; Cerrada, M.; Cabrera, D.; Vásquez, R. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mech. Syst. Signal Process.* 2016, 76–77, 283–293. [CrossRef]
- 19. Shan, S.L.; Khalil-Hani, M.; Bakhteri, R. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* **2016**, *216*, 718–734.
- 20. Bai, Y.; Sun, Z.Z.; Zeng, B.; Deng, J.; Li, C. A multi-pattern deep fusion model for short-term bus passenger flow forecasting. *Appl. Soft Comput.* **2017**, *58*, 669–680. [CrossRef]
- 21. Lee, D.; Kang, S.; Shin, J. Using deep learning techniques to forecast environmental consumption level. *Sustainability* **2017**, *9*, 1894. [CrossRef]
- 22. Li, C.; Sanchez, R.V.; Zurita, G.; Cerrada, M.; Cabrera, D.; Vásquez, R. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing* **2015**, *168*, 119–127. [CrossRef]
- 23. Bai, Y.; Li, Y.; Wang, X.; Xie, J.; Li, C. Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions. *Atmos. Pollut. Res.* **2016**, *7*, 557–566. [CrossRef]
- 24. Lalis, J.T.; Gerardo, B.D.; Byun, Y. An adaptive stopping creterion for backpropagetion learning in feedforward neural network. *Int. J. Multimedia Ubiquitous Eng.* **2014**, *9*, 149–156. [CrossRef]
- 25. Liu, J.P.; Li, C.L. The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection. *Sustainability* **2017**, *9*, 1188. [CrossRef]
- 26. Cho, K.H.; Ilin, A.; Raiko, T. Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. *Lect. Notes Comput. Sci.* **2011**, 6791, 10–17.
- 27. Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef] [PubMed]
- 28. Längkvist, M.; Karlsson, L.; Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognit. Lett.* **2014**, *42*, 11–24. [CrossRef]

- 29. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
- 30. Pizarro, J.; Guerrero, E.; Galindo, P.L. Multiple comparison procedures applied to model selection. *Neurocomputing* **2002**, *48*, 155–173. [CrossRef]
- 31. Almeida, F.R.; Brayner, A.; Rodrigues, J.; Maia, J.E.B. Improving multidimensional wireless sensor network lifetime using Pearson correlation and fractal clustering. *Sensors* **2017**, *17*, 1317. [CrossRef] [PubMed]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).