Sensors and Materials, Vol. 36, No. 1 (2024) 49–65
MYU Tokyo

S & M 3497

Hybrid Approach Combining Simulated Annealing and Deep Neural Network Models for Diagnosing and Predicting Potential Failures in Smart Manufacturing

Yung-Hsiang Hung, Mei-Ling Huang, Wen-Pai Wang,* and Guan-Liang Chen

Department of Industrial Engineering and Management, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan, R.O.C.

(Received May 26, 2023; accepted August 18, 2023)

Keywords: predictive maintenance, recursive feature elimination, simulated annealing, deep neural networks, feature selection

Predictive maintenance is vital in smart manufacturing because it can help reduce downtime and costs and enhance productivity and product quality. Preventive maintenance for computer numerical control (CNC) machine tools is crucial for performance optimization. The effectiveness of preventive maintenance is affected by factors such as the usage environment, maintenance plans, and records. Regular inspection and maintenance are necessary to address problems related to tool wear and spindle condition, which affect machining quality. Accordingly, artificial intelligence algorithms, including deep learning models, have been extensively used for predictive maintenance. Conventional feature engineering methods and Internet of Things-based machine health monitoring systems are effective in this domain. Recursive feature elimination is commonly used for feature selection, but it is computationally intensive. In this study, we established a hybrid approach combining simulated annealing (SA) and deep neural network (DNN) models for diagnosing and predicting potential failures or problems in smart production machines; this approach was noted to exhibit excellent performance in solving complex problems. Integrating SA and DNNs can enhance preventive maintenance, optimize CNC machining processes, and improve productivity and product quality in smart manufacturing. The key advantage of the proposed hybrid approach is its ability to optimize feature selection while reducing computational costs. Therefore, the approach has potential for advancing preventive maintenance in smart manufacturing and provides valuable insights for developing efficient production systems.

1. Introduction

Predictive maintenance involves monitoring the behavior and operating conditions of equipment or machines to predict potential failures and damage. This enables proactive maintenance, which reduces downtime and maintenance costs while maximizing productivity and product quality. In smart manufacturing, predictive maintenance is essential for the real-

*Corresponding author: e-mail: wangwp@ncut.edu.tw
https://doi.org/10.18494/SAM4529

ISSN 0914-4935

time monitoring and prediction of equipment status through automated monitoring, data collection, and analysis. It can help reduce downtime, prevent severe damage, enhance productivity, and improve product quality. Overall, predictive maintenance is crucial in smart manufacturing because it enables businesses to minimize downtime, enhance productivity and product quality, and increase their competitiveness and market share.

Computer numerical control (CNC) machine tools are typically used to process high-precision and high-demand work pieces. Implementing preventive maintenance for such tools can prolong their lifetime, boost efficiency, and enhance product quality. In general, several factors affect the effectiveness of preventive maintenance for CNC machine tools, namely, the usage environment, maintenance plan, and maintenance records. Regarding the first factor, CNC machine tools are frequently operated in production environments in which various factors, such as humidity, temperature, dust, and vibrations, can affect their performance and maintenance. For example, high temperatures and humidity can expedite the aging and damage of CNC machine tool components, and excessive dust and vibration can accelerate the wear and fatigue of machine parts. Regarding the second factor, the maintenance plan for CNC machine tools can affect the effectiveness of preventive maintenance. A maintenance plan encompasses elements such as maintenance cycles, maintenance content, and maintenance methods. Different methods yield distinct effects on the maintenance of CNC machine tools. Finally, regarding the third factor, maintenance records are crucial for the preventive maintenance of CNC machine tools. In summary, the aforementioned factors should be comprehensively analyzed and considered to formulate maintenance plans and measures that enhance the stability, reliability, and productivity of CNC machine tools.

Tool wear and spindle condition are critical factors affecting CNC machining quality. Tool wear is unavoidable during CNC machining. A machine tool becomes dull and worn out with continuous use, resulting in increased cutting force and temperature and potentially reduced surface roughness and work concentration. The effect of tool wear on CNC machining quality is determined by various factors, including the wear process, cutting material, and processing requirements. A tool with a high degree of wear exhibits reduced cutting performance, and its cutting force increases, leading to problems such as surface roughness and size deviation. A spindle is a crucial component of a CNC machine tool; it provides rotational power and carries the tool. The accuracy and stability of the spindle are crucial to the product quality and efficiency of CNC machining. Spindle wear or deviation can cause tool vibration and jumping, adversely affecting machining accuracy and surface quality.

The regular inspection and testing of tool and spindle conditions are necessary to satisfy the stringent quality requirements of CNC machining processes. In addition, prompt tool replacement and spindle maintenance should be performed to ensure the stability and accuracy of CNC machining.

In recent years, artificial intelligence (AI) has been increasingly applied in preventive maintenance.[1] In 2019, Silvestrin *et al.* evaluated conventional and deep learning approaches for predictive maintenance and observed that deep learning approaches are promising but require large amounts of data. They introduced a temporal convolutional neural network, which performed well even with limited data; nevertheless, they determined that conventional feature

engineering-based methods were generally more effective.[2] Zhao *et al.* provided an overview of deep learning applications in machine health monitoring, including auto encoders, restricted Boltzmann machines, convolutional neural networks, and recurrent neural networks. They compared the performance of these applications and discussed emerging trends in deep-learning-based machine health monitoring methods.[3] Pech *et al.* used co-occurrence and cluster analysis to systematically review the literature on predictive maintenance and intelligent sensors in smart factories. They identified current trends and proposed the concept of smart and intelligent predictive maintenance on the basis of their analysis.[4]

An Internet of Things-based CNC machine health monitoring system employs wireless sensor networks to detect various parameters and data for a CNC machine. Such a system provides real-time and accurate information about the machine's health status and corresponding preventive maintenance measures and decisions.

Wang *et al.* proposed a multi-scale principal component analysis (PCA) model for online tool wear monitoring in the milling process.[5] In multi-scale PCA, which is based on conventional PCA, a training sample set is decomposed, and statistical indices and control limits are applied to monitor tool wear. They reported that the integration of PCA with wavelet transformation enhanced the accuracy and robustness of the model.[5] García-Nieto *et al.* proposed a particle swarm optimization (PSO)–support vector machine (SVM) model, a hybrid regression model, for predicting milling tool wear in various cutting scenarios.[6] They observed that the PSO mechanism can notably improve regression accuracy and that their model can be used to identify factors contributing to tool wear; moreover, the PSO–SVM model offers a straightforward and interpretable solution with high computational efficiency. Their study provided key findings and implications for enhancing milling machine performance.[6] These research findings underscore the importance of AI technology in preventive maintenance for CNC machines and offer practical solutions.

Recursive feature elimination (RFE) is a widely adopted feature selection technique in machine learning. This technique entails iteratively eliminating the least important features until a desired level of performance is achieved.[7–9] Yan and Zhang proposed a SVM–RFE+CBR correlation bias reduction algorithm for RFE feature selection in gas sensor data; they observed that this algorithm outperformed the original SVM–RFE and other algorithms in terms of classification accuracy.[7] To evaluate the algorithm's performance, they conducted experiments on synthetic and breath analysis datasets. Additionally, they examined an ensemble method to enhance their algorithm's stability and analyzed the rankings of features to gain relevant insights.[10] However, for large datasets with numerous features, RFE can be computationally intensive. Simulated annealing (SA) is a metaheuristic optimization algorithm that is inspired by the physical annealing process in metallurgy. It is commonly used for the global optimization of complex functions. Li *et al.* proposed a hybrid approach combining a genetic algorithm and SA to optimize process planning in a dynamic workshop environment.[11] They modeled process planning as a combinatorial optimization problem with constraints and evaluated the machining cost on the basis of several factors. Their results indicated that the hybrid approach was highly satisfactory in optimizing process planning for prismatic parts. Deep neural networks (DNNs) are effective tools for solving complex problems, such as image and speech recognition, and they

consist of multiple hidden layers between input and output layers.[12,13] SA can be used to optimize the weights of DNNs or search for the optimal subset of features to serve as inputs to DNNs.[14] The primary advantage of using SA to optimize DNNs is its ability to escape local minima and find better global optima, thereby improving the accuracy of the DNNs while reducing computational costs. Accordingly, this study presents a hybrid approach combining SA and DNN models for diagnosing and predicting potential failures or problems in smart production machines. Overall, the advantages of using SA to optimize DNNs and the hybrid SA-DNN approach for diagnosing and predicting failures in smart production machines include escaping local minima and finding global optima, reducing computational costs, and enhancing the diagnostic and predictive capabilities of the models.

2. Materials and Methods

2.1 Dataset

In this study, we used the UCI AI4I dataset,[15] which contains 10000 data points with six features: UDI, Product ID, Type, Air Temperature, Process Temperature, Rotational Speed, Torque, and Tool Wear. The dataset provides information about products (Table 1). Product ID is denoted by the letter L, M, or H, representing low-quality (50% of all products), medium-quality (30% of all products), or high-quality (20% of all products) variants, respectively; each variant is associated with a specific serial number. Air Temperature (K) and Process Temperature (K) are generated through a random walk process and then adjusted through the addition of a standard deviation of 2 and 300 K, respectively. Furthermore, Rotational Speed (rpm) is derived on the basis of a power of 2860 W and exhibits a normal distribution, but it is subject to noise overlay. Torque (Nm) values are all positive (approximately 40 Nm with a standard deviation of 10 Nm) and are normally distributed. Finally, Tool Wear (min) is determined on the basis of the H, M, and L quality variants; specifically, for the H, M, and L quality variants, the tool wear time is determined to be 5, 3, and 2 min, respectively.

The UCI AI4I dataset also includes a "machine failure" label, encompassing five independent failure modes. Tool wear failure (TWF) occurs at a random tool wear time in the range of 200–240 min, where the tool is either replaced or fails, with 69 replacements and 51 failures. Heat dissipation failure (HDF) occurs when the difference between air and process temperatures is lower than 8.6 K and the rotational speed of the tool is lower than 1380 rpm. Power failure (PWF) occurs when the product of Torque and Rotational Speed drops below 3500 W or exceeds 9000 W. Overstrain failure (OSF) occurs when the product of Tool Wear and Torque exceeds a certain threshold. Random failures (RNFs) also have a 0.1% probability of occurring during any given process. Table 2 presents an overview of this information.

2.2 Dimensionality reduction and feature ranking

Reducing the dimensionality of a feature space R is a common problem in machine learning, particularly in the context of classification, because it helps overcome the risk of over fitting. The goal of dimensionality reduction is to reduce the number of dimensions d while retaining a

Table 1
Instants and histograms for each variable.

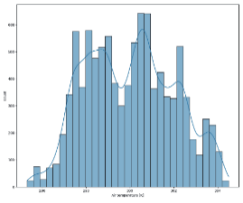
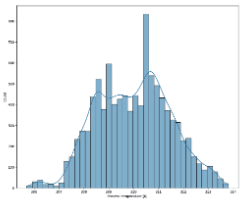
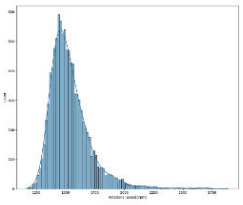
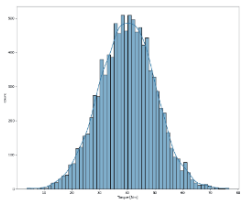
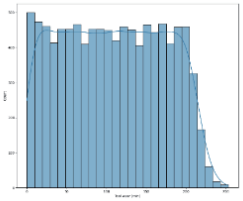
Variables	Instants	Histograms	Shapes of Outliers
Air temperature (K)	10000		Skewness: 0.11 Kurtosis: −0.84
Process temperature (K)	10000		Skewness: 0.02 Kurtosis: −0.50
Rotational speed (rpm)	9582		Skewness: 1.99 Kurtosis: 7.39
Torque (Nm)	9535		Skewness: −0.01 Kurtosis: −0.01
Tool wear (min)	9535		Skewness: 0.03 Kurtosis: −1.17

Table 2
Machine failure types and variables.

Failure Type	Data type	Variable
Tool wear failure	[0, 1]	TWF
Heat dissipation failure	[0, 1]	HDF
Power failure	[0, 1]	PWF
Overstrain failure	[0, 1]	OSF
Random failures	[0, 1]	RNF
Machine failure	[0, 1]	NAN

sufficient quantity of relevant information. By iteratively eliminating the least important features, RFE improves model performance.

The RFE algorithm involves the following steps:

1. A machine learning model is used for feature selection.
2. The model is trained using the entire dataset to obtain the initial feature importance ranking (weights w_i are optimized using a cost function C).
3. The model is fitted using all the features ranked according to their importance.
4. The least important features are removed, and the model is fitted again using the reduced dataset.

Steps 3 and 4 are repeated until the desired number of features is selected. The RFE algorithm is flexible and can be used with any of the machine learning models that provide a feature importance score, such as decision tree, random forest, and DNN models. By selecting the most important features iteratively, the RFE algorithm simplifies complex models and improves interpretability.[16]

2.3 Multiclass classification problems

In this study, we explored the use of neural networks for multiclass classification tasks. The input space X can be defined as a d-dimensional real space R, and the output space Y can be defined as a set of K categories {1, ..., 5}. A training set S, which is independent and identically distributed, is sampled from $X \times Y$ based on distribution D. The objective is to learn a prediction model $X \times Y \rightarrow R$ from the training set, which produces an output vector $f(x, k)$ for each instance $x \in X$, indicating its likelihood of belonging to category k. The final classification is achieved using the following expression:

$$\arg\max x_{k \in y} y = f\left(x, k\right). \tag{1}$$

DNNs are a class of artificial neural networks (ANNs) composed of multiple layers of interconnected nodes or neurons. DNNs are based on the concept of deep learning, which involves training ANNs with multiple layers. A deeper architecture enables the modeling of more complex relationships between input and output data; hence, DNNs are suitable for handling intricate tasks.[17]

Consider a DNN architecture with h hidden layers and assume that ReLU activation functions are used.[3] For this DNN, the input layer is a d-dimensional vector, denoted as the ith layer, whereas the output layer is a k-dimensional scalar, denoted as the hth layer. In this context, n_i denotes the number of neurons in the ith layer. The output of the network can be defined as

$$y_i = z_i^o \left(\sum_{j=1}^{h} w_{ji}^o \times z_m^h \left(\sum_{m=1}^{n} w_{jm}^h x_m + b^h \right) \right) + b^o, \tag{2}$$

where represents the activation function of the ith neuron in the output layer y_i, w_{ji}^o represents the weight of connection to the jth neuron in the output layer, z_m^h represents the activation

function of the jth neuron in the hidden layer, w_{jm}^{h} represents the weights of the mth neurons in the hidden layer, x_m represents the input matrix, and b^h and b^o represent the biases of the hidden and output layers, respectively. The performance of the DNN is evaluated using the *RMSE*, which is expressed as

$$RMSE = \sqrt{\frac{\sum_i \left(y_i - t_i \right)^2}{r-1}}, \tag{3}$$

where y_i denotes the desired output, t_i denotes the actual output, and r denotes the number of training samples.

2.4 DNN optimization using SA

SA is a probabilistic optimization algorithm that can be employed to optimize the parameters of a DNN by minimizing the loss function $\mathcal{L}(\cdot)$. The basic principle of this algorithm is to randomly perturb the network parameters and then evaluate the change in loss function. If the change is beneficial, the new parameters are accepted and the process is repeated. If the change is not beneficial, the new parameters are accepted with a certain probability; this probability depends on a temperature parameter, which governs the algorithm's acceptance probability for worse solutions. The acceptance probability can be expressed as

$$P = \exp\left(\frac{-\left(E(S') - E(S) \right)}{KT} \right), \tag{4}$$

where $\exp(\cdot)$ is an exponential function, $E(S)$ is the energy of the current state, and $E(S')$ is the energy of the new state; the division by KT is used to scale the energy difference between the current state and the proposed state according to the temperature. The cooling schedule equation $T_i = T_0/(1 + \alpha \times i)$ represents the decrease in temperature over time, where T_0 represents the initial temperature and α represents the rate of cooling.

SA involves the following steps:

1. The parameters of the DNN are initialized with random values.
2. The performance of the DNN is evaluated using a validation set and a loss function.
3. The parameters of the DNN are perturbed by adding random noise.
4. The performance of the perturbed DNN on the validation set is evaluated using the loss function.
5. If the performance of the perturbed DNN is higher than that of the original DNN, the new parameters are accepted.
6. If the performance of the perturbed DNN is lower than that of the original DNN, the new parameters are accepted with a certain probability; this probability depends on a temperature parameter.
7. Steps 3–6 are repeated for a certain number of iterations or until convergence.

SA is used to optimize DNN models, primarily because it can help escape local minima in the loss function and obtain a better global optimum. However, in SA, convergence can be slow, and the choice of temperature schedule and noise level can substantially affect the algorithm's performance. Therefore, SA hyperparameters should be carefully tuned to achieve optimal performance.

2.5 Methods

In this study, we developed a novel approach based on deep learning for failure mode classification. We first examined the data structure and type by using feature encoding techniques to handle string-type samples and facilitate model integration. Statistical analyses were then performed, including kurtosis and skewness calculation, and outliers were detected using the quartile range method. Next, noncontributing outliers were removed to enhance model learning. The RFE algorithm was applied to rank the importance of the feature variables, further enabling the selection of a subset of significant factors for model construction. To address the problem of data imbalance due to the underrepresentation of certain failure types, we employed the adaptive synthetic sampling (ADASYN) algorithm.[18] This algorithm generates synthetic samples for the minority class based on feature weights while preserving the original data structure.

Adam and SA employ distinct optimization approaches and exhibit different characteristics. Adam is highly efficient for local optimization, converging rapidly, but it may encounter difficulties in escaping local optima. On the other hand, SA emphasizes global optimization, showcasing resilience to initial conditions and a greater capacity for discovering superior global solutions. However, SA may require more iterations to achieve convergence. The choice between these algorithms depends on the specific problem, the desired optimization objectives, and the balance between exploration and exploitation in the optimization process.[19] The dataset was first preprocessed through standardization for model construction, and the deep learning model's learning rate was optimized using both the Adam and SA algorithms. Classification accuracy was assessed to determine the effectiveness of the proposed approach, and the optimal failure classification model was determined.

This study involved four main steps. First, we performed data preprocessing to handle data of different ranges and dimensions, and classified them into the same intervals and ranges to reduce the effects of features and differences on model performance. Second, we used RFE to eliminate weakly related features until the specified number of features was reached, thereby addressing the dependence among model features through repeated training. Third, we used a synthetic sampling method to increase the number of learning samples in order to prevent the replication of the same minority data. Different samples were synthesized by assigning different weights to the samples with different minorities. Finally, we applied SA to optimize the learning rate. The loss function of the deep learning models was calculated, and the original learning rate was compared with the new learning rate. When the new learning rate was less than the original learning rate, it was fully accepted; otherwise, it was accepted with a certain probability, and the weights and biases of the neural network were updated in each round. The flowchart of our approach is illustrated in Fig. 1.

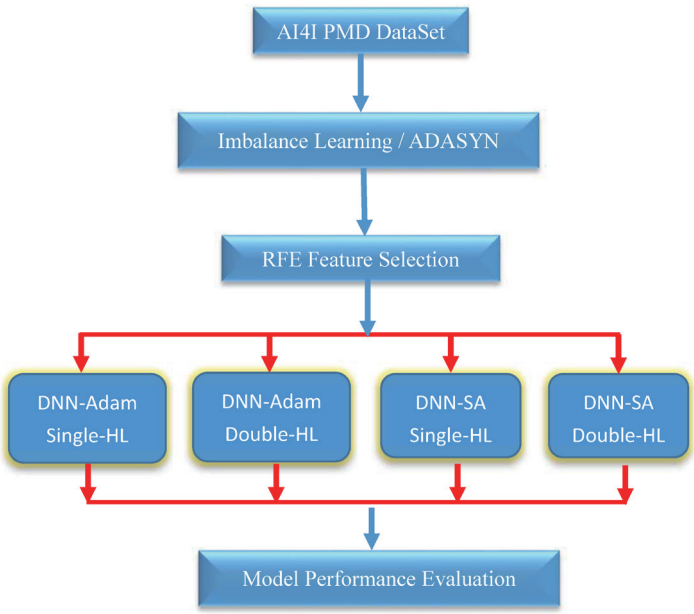


Fig. 1. (Color online) Architecture of AI4I PDM prediction model.

2.6 Performance evaluation

The area under the receiver operating characteristic (ROC) curve, precision, and recall are commonly used evaluation metrics in binary classification. A classifier's decisions can be effectively summarized and represented using a confusion matrix or contingency table, which provides a structured view of the classification outcome. A confusion matrix comprises four categories, namely, true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*), which are used to assess a classification model's performance. *TP* represents correctly labeled positive instances, *FP* represents negative instances incorrectly labeled as positive, *TN* represents correctly labeled negative instances, and *FN* represents positive instances incorrectly labeled as negative. These metrics provide valuable information about a classification model's accuracy. We used the following metrics to evaluate the predictive performance and efficacy of the failure diagnosis models in our approach:[20] ROC curve, accuracy, precision, recall, F1-score, and support.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

3. Results

In this study, we used DNN models for failure mode classification. The UCI AI4I dataset was used for the classification process in this study. The modeling process and experiments were conducted using the Python programming language. The experimental procedure involved several steps. First, the dataset was preprocessed by examining the data distribution, performing data standardization, and removing outliers. Subsequently, the RFE algorithm was employed to select a subset of important features, which were further organized into a new dataset. Weight design and sampling techniques were applied to compile the final experimental dataset. To enhance the accuracy and predictive performance of the DNN models, we leveraged preprocessing techniques and feature selection and optimization algorithms.

3.1 Data analysis and visualization

To ensure data quality, data points that fell outside the range of [Q1 − 1.5 IQR, Q3 + 1.5 IQR] were excluded. In addition, 418 failure events related to rotational speed and 47 samples related to torque were excluded, resulting in the removal of 465 data points. The dataset was determined to be imbalanced because it involved a higher representation of operational machines than faulty machines, and this imbalance may affect the training process of the deep learning models.

3.2 Optimal feature selection

In this study, we used the RFE algorithm to rank eight feature variables according to their importance in relation to machine failure. Table 3 presents the rankings and scores of the variables. The experimental results indicated that Type, Air Temperature, Process Temperature, Rotational Speed, and Torque had the lowest scores (0.9685). Furthermore, we identified the optimal subset of features by using our models' cross-validation scores. Thus, we identified a subset of five optimal feature variables, which we incorporated into the deep learning models (Fig. 2).

3.3 Dataset imbalance and ADASYN weight sampling

As discussed in Sect. 3.1, our data analysis results revealed that 97% of the data instances represented normal machine operation (Class 0), whereas only 3% represented other failure classes, indicating a highly imbalanced distribution. To address this problem in the context of multiclass classification, we employed the ADASYN weight sampling method to generate a new dataset; consequently, the sample size increased from 9535 instances to 55626 instances (Fig. 3).

Table 3
Variables' rankings and scores.

Feature	Score	Ranking
Type	0.9685	1
Air temperature (K)	0.9685	1
Process temperature (K)	0.9685	1
Rotational speed (rpm)	0.9685	1
Torque (Nm)	0.9685	1
Tool wear	0.968	2
Product ID	0.9658	3
UDI	0.9657	4

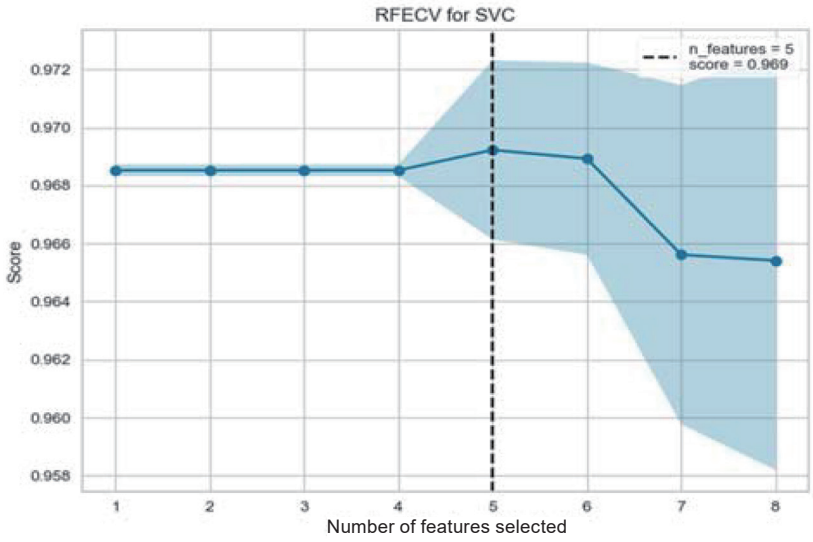
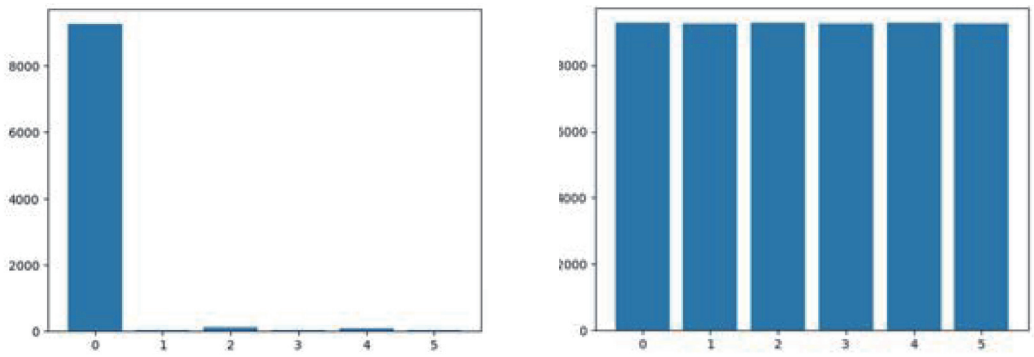


Fig. 2. (Color online) Optimal feature selection for AI4I.



(a) (b)

Fig. 3. (Color online) Imbalance data and resampling results. (a) Original dataset. (b) ADASYN weight sampling.

4. Experimental Results and Analysis

During the modeling process, we used 90% of our dataset as a training set and 10% as a test set. Two network architectures comprising a single hidden layer and double hidden layers were used to construct DNN models. The SA and Adam optimization algorithms were employed to optimize the model learning rate. The input layer of each DNN model contained five neurons along with the ReLU activation function, and the output layer contained six failure modes along with the Softmax activation function. The hidden layer comprised 100 neurons. The models were trained for 100 epochs by using a batch size of 400. The loss function was computed using cross-entropy, and the performance and accuracy of the models were evaluated.

In this study, we evaluated four experimental models (Models I–IV). Model I contained a single hidden layer and was optimized using the Adam algorithm. The experimental results revealed that the model achieved a classification accuracy of 93.58%, with a total execution time of 37.9523 s. The confusion matrix indicated that the model exhibited poorer classification results for Class0 than for other classes (ROC area = 0.97); this was primarily due to the lower recall value, indicating that some instances of Class0 were not correctly classified. Loss curve analysis revealed that the curve had not fully converged.

Model II contained a double hidden layer and was optimized using the Adam optimizer. The experimental results showed that the model achieved a classification accuracy of 96.54%, with a total execution time of 83.3118 s. The training and validation curves derived for this model revealed a more favorable performance than did those derived for Model I. Additionally, this model achieved a clearer and faster convergence than did Model I, which could be attributed to the notable decrease in distance between the training and validation curves.

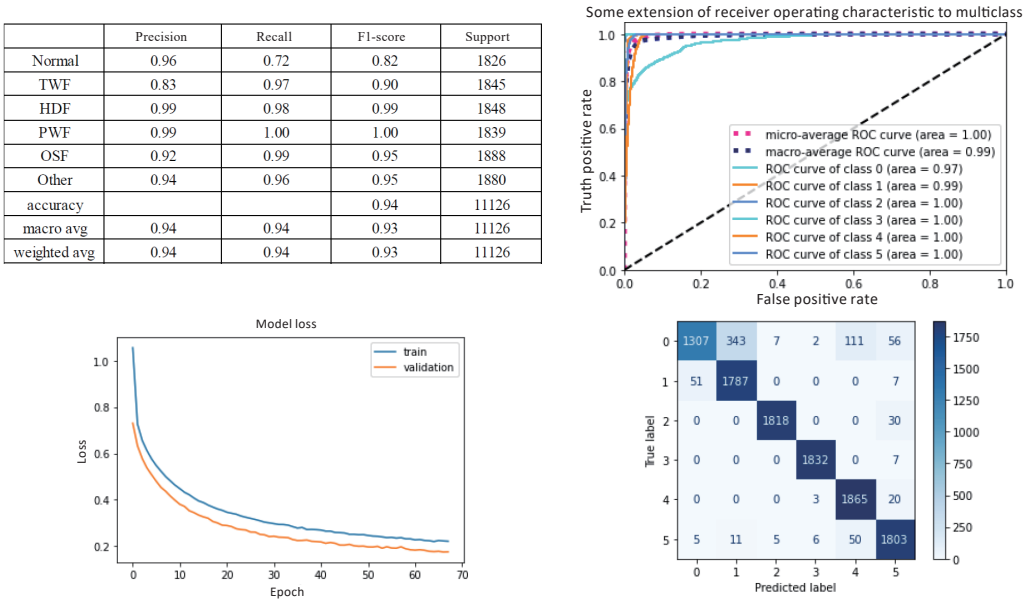
Model III contained a single hidden layer and was optimized using the SA algorithm. The experimental results revealed that the model achieved a classification accuracy of 95.37%, with a total execution time of 94.9281 s. Similar to Model I, Model III produced suboptimal classification results for Class0. However, the SA optimizer engendered higher model recall and accuracy rates and an ideal convergence.

Finally, Model IV contained a double hidden layer and was optimized using the SA algorithm. The experimental results revealed that the model achieved a classification accuracy of 97.09%, with a total execution time of 231.7983 s. Model IV achieved the highest accuracy among the four models but required a longer execution time.

The ROC curve derived for this model closely aligned with the ideal ROC curve at the end of the training process, and the gap between the two curves was minimal. In general, a smaller training loss indicates that a model is applicable to new data. Our experimental results are summarized in Table 4, and the performance indices (precision, recall, F1-score, support, ROC curve, and confusion matrix) derived for Models I–IV are presented in Figs. 4(a) and 4(b).

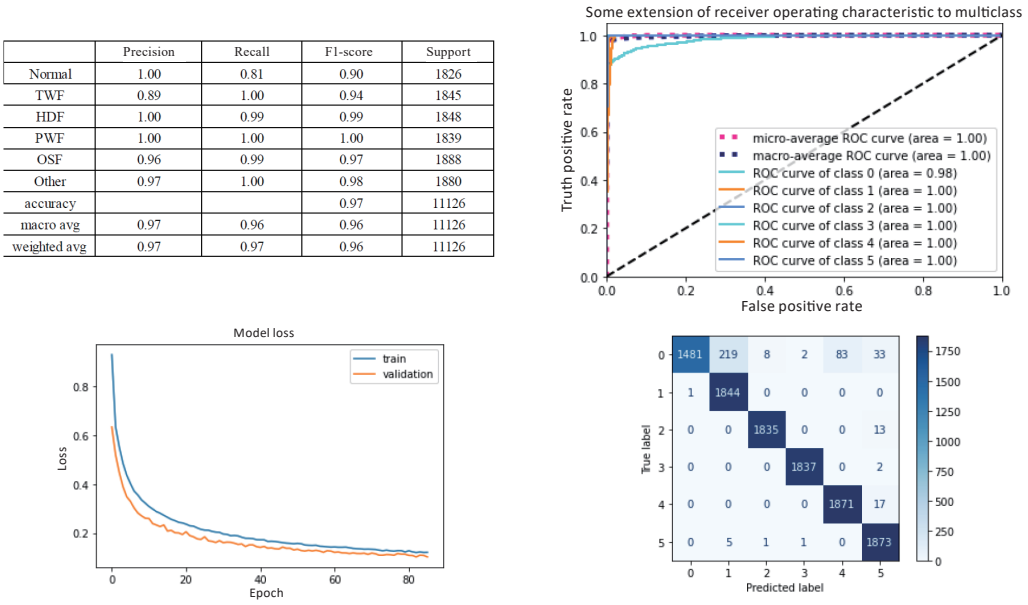
Table 4
Experimental results for single- and double-hidden-layer models.

	Adam		SA	
	SingleHL Model I	SingleHL Model II	DoubleHL Model III	DoubleHL Model IV
ROC	0.99	1.00	0.99	1.00
Recall	0.94	0.96	0.95	0.97
F1-score	0.93	0.96	0.95	0.97
Accuracy (%)	93.58	95.37	96.54	97.09
ProcessTime (s)	37.9523	92.9281	83.3118	231.7983



	Precision	Recall	F1-score	Support
Normal	0.96	0.72	0.82	1826
TWF	0.83	0.97	0.90	1845
HDF	0.99	0.98	0.99	1848
PWF	0.99	1.00	1.00	1839
OSF	0.92	0.99	0.95	1888
Other	0.94	0.96	0.95	1880
accuracy			0.94	11126
macro avg	0.94	0.94	0.93	11126
weighted avg	0.94	0.94	0.93	11126

(a)

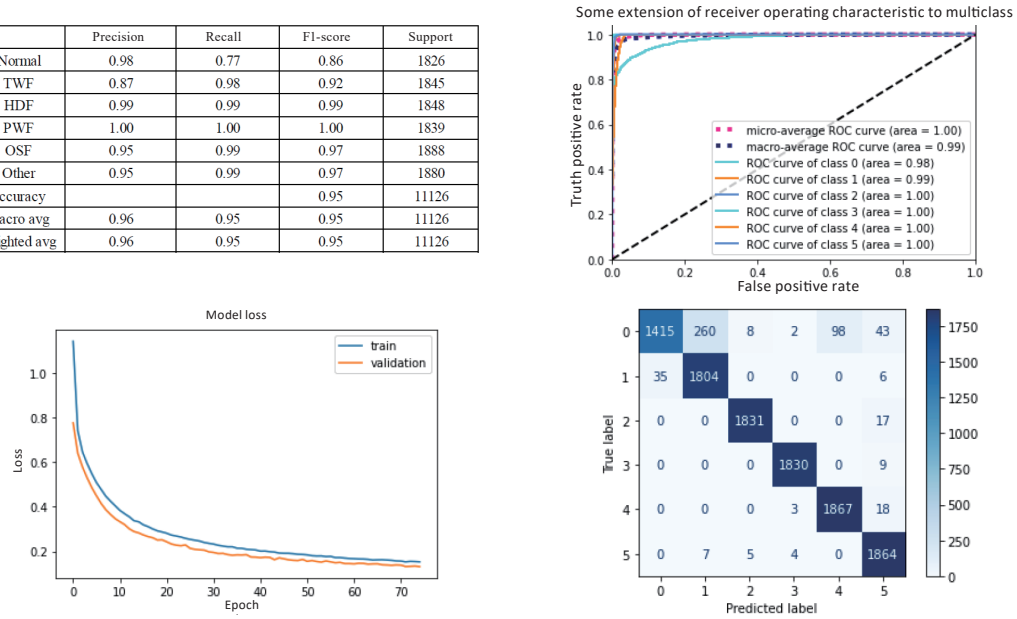


	Precision	Recall	F1-score	Support
Normal	1.00	0.81	0.90	1826
TWF	0.89	1.00	0.94	1845
HDF	1.00	0.99	0.99	1848
PWF	1.00	1.00	1.00	1839
OSF	0.96	0.99	0.97	1888
Other	0.97	1.00	0.98	1880
accuracy			0.97	11126
macro avg	0.97	0.96	0.96	11126
weighted avg	0.97	0.97	0.96	11126

(b)

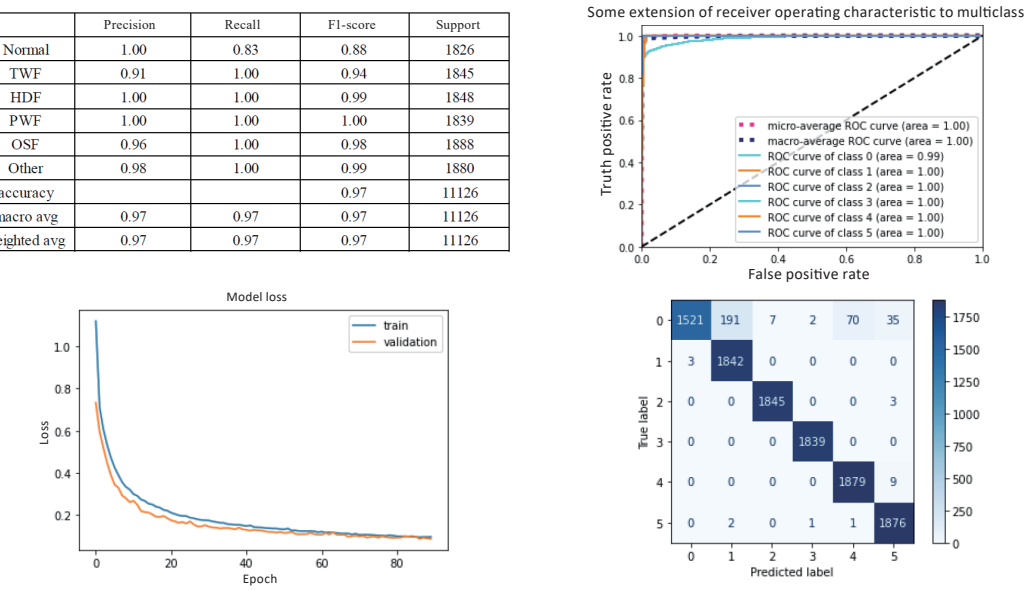
Fig. 4. (Color online) Experiments' performance evaluation. (a) Model I: Single HL Adam. (b) Model II: Double HL Adam. (c) Model III: Single HL SA. (d) Model IV: Double HL SA.

	Precision	Recall	F1-score	Support
Normal	0.98	0.77	0.86	1826
TWF	0.87	0.98	0.92	1845
HDF	0.99	0.99	0.99	1848
PWF	1.00	1.00	1.00	1839
OSF	0.95	0.99	0.97	1888
Other	0.95	0.99	0.97	1880
accuracy			0.95	11126
macro avg	0.96	0.95	0.95	11126
weighted avg	0.96	0.95	0.95	11126



(c)

	Precision	Recall	F1-score	Support
Normal	1.00	0.83	0.88	1826
TWF	0.91	1.00	0.94	1845
HDF	1.00	1.00	0.99	1848
PWF	1.00	1.00	1.00	1839
OSF	0.96	1.00	0.98	1888
Other	0.98	1.00	0.99	1880
accuracy			0.97	11126
macro avg	0.97	0.97	0.97	11126
weighted avg	0.97	0.97	0.97	11126



(d)

Fig. 4. (Color online) (Continued) Experiments' performance evaluation. (a) Model I: Single HL Adam. (b) Model II: Double HL Adam. (c) Model III: Single HL SA. (d) Model IV: Double HL SA.

5. Conclusions

The conventional manufacturing approach focuses on producing individual components, followed by processing and then quality inspection. Numerous defective products resulting from incorrect machine settings may be identified during the quality inspection phase. Even for large

production volumes, minor errors may interfere with the cost–benefit analysis. However, with the emergence of modern mass-customized products, even minor production faults can be economically burdensome. Thus, error reduction is a critical concern for contemporary businesses. In this study, we leveraged historical processing data to detect and classify potential machine downtime events. The key findings of this study are summarized as follows:

Outlier removal and encoding techniques can enable rapid data analysis, enhance model convergence, and mitigate problems related to gradient descent optimization. In addition, applying weight sampling to augment data from minority instances can effectively improve model learning capacity and recall rates. Although the accuracy of classifying normal patterns may decrease slightly, this trade-off can considerably enhance a model's ability to identify other failure patterns. Furthermore, feature selection through the elimination of low-correlation features can reduce runtime and enhance model accuracy.

Although SA may increase model runtime, it can substantially improve model accuracy (Model II-96.54%, Model IV-97.09%). For scenarios in which accuracy optimization is prioritized without the consideration of time constraints, SA should be considered for model optimization. Compared with the Adam optimizer in scenarios involving equivalent environmental parameters, SA could exhibit superior effectiveness in achieving model convergence. This can be attributed to the effective transformation of the learning rate, preventing both models from overfitting and underfitting. The oversampling of weights can enable a model to effectively recognize minority classes without compromising the integrity of the original structure. Furthermore, feature engineering can reduce irrelevant features and optimize data processing, facilitating efficient and rapid model learning.

In summary, the study's contributions include techniques for rapid data analysis, improved model convergence, enhanced recall rates, runtime reduction through feature selection, SA for accuracy optimization, weight oversampling for recognizing minority classes, and efficient feature engineering. These findings provide valuable insights for error reduction in modern manufacturing processes, enabling businesses to improve efficiency and address cost concerns associated with production faults. By addressing these research points, future studies can advance the understanding and application of outlier removal, handling imbalanced datasets, feature selection, optimization strategies, and feature engineering techniques, leading to improved performance and practical solutions for diagnosing and predicting potential failures in various domains, including smart production machines.

References

1 G. Wang, Y. Zhang, C. Liu, Q. Xie, and Y. Xu: J. Intell. Manuf. **30** (2019) 113. https://doi.org/10.1007/s10845-016-1235-9
2 L. P. Silvestrin, M. Hoogendoorn, and G. Koole: Proc. 2019 IEEE Symp. Series Computational Intelligence (SSCI, 2019) 760−767. https://doi.org/10.1109/SSCI44817.2019.9003044
3 R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao: Mech. Syst. Signal Process **115** (2019) 213. https://doi.org/10.1016/j.ymssp.2018.05.050
4 M. Pech, J. Vrchota, and J. Bednář: Sensors **21** (2021) 1470. https://doi.org/10.3390/s21041470
5 J. Wang, J. Xie, R. Zhao, K. Mao, and L. Zhang: IEEE Trans. Instrum. Meas. **65** (2016) 2527. https://doi.org/10.1109/TIM.2016.2584238
6 P. J. García-Nieto, E. García-Gonzalo, J. A. Vilán Vilán, and A. S. Robleda: Int. J. Adv. Manuf. Technol. **86** (2016) 769. https://doi.org/10.1007/s00170-015-8148-1

7 K. Yan and D. Zhang: Sens. Actuators B.**212** (2015) 353. https://doi.org/10.1016/j.snb.2015.02.025
8 H. Jeon and S. Oh: Appl. Sci. **10** (2020) 3211. https://doi.org/10.3390/app10093211
9 G. E. Hinton and R. R. Salakhutdinov: Sci. **313** (2006) 504. https://doi.org/10.1126/science.1127647
10 K. Yan and D. Zhang: Proc. 2014 Int. Conf. Medical Biometrics (IEEE, 2014) 35−40. https://doi.org/10.1109/ICMB.2014.14
11 W. D. Li, S. K. Ong, and A. Y. C. Nee: Int. J. Prod. Res. **14** (2010) 1899. https://doi.org/10.1080/00207540110119991
12 Q.V. Lee, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A.Y. Ng: Proc. 2011 28th Int. Conf. Machine Learning (2011) 265−272.
13 W.-J. Lin, S.-H. Lo, H.-T. Young, and C.-L. Hung: Appl. Sci. **9** (2019) 1462. https://doi.org/10.3390/app9071462
14 L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy: Procedia Comput. Sci. **72** (2015), 137. https://doi.org/10.1016/j.procs.2015.12.114
15 AI4I 2020 Predictive Maintenance Dataset: https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset (accessed April 2021).
16 R. Kohavi and G. H. John: Artif. Intell. **97** (1997) 273. https://doi.org/10.1016/S0004-3702(97)00043-X
17 Y. LeCun, Y. Bengio, and G. Hinton: Nature **521** (2015) 436. https://doi.org/10.1038/nature14539
18 H. B. He, Y. Bai, E. A. Garcia, and S. T. Li: Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN 2008) 1322−1328. https://doi.org/10.1109/IJCNN.2008.4633969
19 D. P. Kingma and J. Ba: Proc. 3rd Int. Conf. Learning Representations, San Diego, (ICLR 2015). https://doi.org/10.48550/arXiv.1412.6980
20 D. M. W. Powers: Int. J. Mach. Learn. Technol. **2** (2011) 37. https://doi.org/10.48550/arXiv.2010.16061

About the Authors

Yung-Hsiang Hung received his Ph.D. degree in Industrial Engineering and Management from National Chiao-Tung University, R.O.C., in 2002. He is a professor of Industrial Engineering and Management at National Chin-Yi University of Technology, Taiwan, R.O.C. His main research interests are in statistical process control, service quality management, process capability analysis, semiconductor manufacturing management, and process parameter optimization design. (hys502@ncut.edu.tw)

Mei-Ling Huang received her M.S. and Ph.D. degrees in Industrial Engineering from the University of Wisconsin–Madison and National Chiao-Tung University, respectively. She is now affiliated with the Department of Industrial Engineering & Management at National Chin-Yi University of Technology. Her research interests include quality management, quality engineering, data mining, and medical diagnosis. (huangml@ncut.edu.tw)

Wen-Pai Wang received his M.S. and Ph.D. degrees in Industrial Engineering from Kansas State University and National Chiao-Tung University, respectively. He is a professor of Industrial Engineering and Management at National Chin-Yi University of Technology. His research interests are in fuzzy set theory, product development, operations/manufacturing management, and decision theory. (wangwp@ncut.edu.tw)

Guan-Liang Chen received his M.S. degree from the Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung City, Taiwan, in 2019. His research interests are in artificial engineering and process parameter optimization design.
(leohados@gmail.com)