Data Driven Anomaly Diagnosis for Machining Processes

Liang, Y., Wang, S., Li, W. & Lu, X.

Original citation:
Liang, Y, Wang, S, Li, W & Lu, X 2019, 'Data Driven Anomaly Diagnosis for Machining Processes' Engineering, vol. 5, pp. 646-652.
https://dx.doi.org/10.1016/j.eng.2019.03.012

DOI 10.1016/j.eng.2019.03.012
ISSN 2095-8099
ESSN 2096-0026

Publisher: Elsevier

Engineering 5 (2019) 646–652

Contents lists available at ScienceDirect

Engineering

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

Research
Intelligent Manufacturing—Article

Data-Driven Anomaly Diagnosis for Machining Processes

Y.C. Liang [a], S. Wang [a], W.D. Li [a,b,*], X. Lu [a]

[a] Faculty of Engineering, Environment and Computing, Coventry University, Coventry CV1 5FB, UK
[b] School of Logistics Engineering, Wuhan University of Technology, Wuhan 430070, China

ARTICLE INFO

Article history:
Received 17 July 2018
Revised 18 December 2018
Accepted 14 March 2019
Available online 19 June 2019

Keywords:
Computer numerical control machining
Anomaly detection
Fruit fly optimization algorithm
Data-driven method

ABSTRACT

To achieve zero-defect production during computer numerical control (CNC) machining processes, it is imperative to develop effective diagnosis systems to detect anomalies efficiently. However, due to the dynamic conditions of the machine and tooling during machining processes, the relevant diagnosis systems currently adopted in industries are incompetent. To address this issue, this paper presents a novel data-driven diagnosis system for anomalies. In this system, power data for condition monitoring are continuously collected during dynamic machining processes to support online diagnosis analysis. To facilitate the analysis, preprocessing mechanisms have been designed to de-noise, normalize, and align the monitored data. Important features are extracted from the monitored data and thresholds are defined to identify anomalies. Considering the dynamic conditions of the machine and tooling during machining processes, the thresholds used to identify anomalies can vary. Based on historical data, the values of thresholds are optimized using a fruit fly optimization (FFO) algorithm to achieve more accurate detection. Practical case studies were used to validate the system, thereby demonstrating the potential and effectiveness of the system for industrial applications.

1. Introduction

Modern manufacturing is characterized as high value, low volume, and high customization, and requires zero-defect production management in order to minimize scrap rates and improve product quality and productivity. However, unexpected anomalies (e.g., machining tool breakage, machine spindle failure, or severe tool wear) can cripple the pursuit of the zero-defect target. It is critical to develop effective diagnosis systems to efficiently detect unexpected anomalies during machining processes, and thus permit appropriate adjustments to be made in order to address the anomalies [1,2]. In response to this need, the European Commission has promoted the "zero-defect manufacturing" concept in manufacturing industries. Accordingly, research projects have been funded in order to identify solutions (e.g., the Intelligent Fault Correction and self-Optimizing Manufacturing systems (IFaCOM) project). From an industrial perspective, some diagnosis systems have been developed and are deployed in factories. A popular strategy in such systems is to identify anomalies by comparing key performance indicators (KPIs) and static thresholds that have been preset by experienced engineers. However, machining processes usually occur under varying working conditions, leading to high dynamics during machining processes. Thus, diagnosis systems that are based on a static threshold setting are unable to address dynamics effectively.

In recent years, smart sensors and cyber–physical systems (CPS) have increasingly been integrated into factories to monitor the dynamic conditions of machining equipment and tooling. As a result, data-driven diagnosis systems have been actively investigated [3–5]. In such systems, intelligent and deep learning algorithms are leveraged in order to mine abnormal patterns from large data streams through time-, frequency-, or time/frequency-domain analysis [6,7]. In order to apply data-driven systems more effectively in industries, it is essential to carry out further research to improve system performance in data processing and analysis.

In this paper, a novel data-driven diagnosis system for computer numerical control (CNC) machining processes is presented. Based on this system, machining processes are continuously monitored to collect data. Analysis is conducted on the monitored data in order to dynamically detect anomalies in the machines and tooling. The innovative characteristics of the system are as follows:

(1) De-noising, normalization, and alignment mechanisms on monitored data have been designed to facilitate anomaly analyses.

* Corresponding author.
E-mail address: weidong.li@coventry.ac.uk (W.D. Li).

https://doi.org/10.1016/j.eng.2019.03.012

(2) A set of features has been defined to represent the most important aspects of the monitored data. Thresholds are used to identify anomalies based on feature comparison. A fruit fly optimization (FFO) algorithm has been applied to optimize the thresholds in order to achieve more accurate diagnosis for dynamic machining processes.

(3) The system has been validated using industrial case studies to prove its effectiveness in practical machining processes.

2. Literature review

In the past, physics- and model-based diagnosis approaches have been the dominant approaches. In recent years, by leveraging the rapid progress that has been made in smart sensors, data analytics, and deep learning technologies, data-driven algorithms have been developed to enhance the effectiveness and performance of diagnosis (e.g., Boltzmann machines, support vector machines (SVMs), convolutional neural networks (CNNs), etc.). Hu et al. [8] developed a method that combined a deep Boltzmann machine algorithm with a multi-grained scanning forest ensemble algorithm to mine faults for industrial equipment. Tian et al. [9] designed a modified SVM to diagnose faults in steel plants; in this method, the data dimension is reduced by a recursive feature elimination (RFE) algorithm in order to speed up computation. Zheng et al. [10] proposed composite multiscale fuzzy entropy (CMFE) and ensemble support vector machines (ESVMs) to extract nonlinear features and classify rolling bearing faults. However, redundant and irrelevant features from data were used, which could reduce the true detection rate significantly and increase the computational time. Wu and Zhao [11] proposed a deep CNN model to detect chemical process faults. However, deep CNN usually requires a high computation time. Madhusudana et al. [12] developed a decision tree technique (J48 algorithm) to detect faulty conditions for face milling tools. In that work, a set of discrete wavelet features were extracted from sound signals by utilizing a discrete wavelet transform (DWT) method. The limit of this research is that the decision tree structure and threshold are difficult to define. Lu et al. [13] proposed a dual reduced kernel extreme learning machine method to diagnose aero-engine faults. Wen et al. [14] proposed a new CNN based on LeNet-5; the proposed CNN was tested for motor bearing, as well as for self-priming centrifugal pump and axial piston hydraulic pump fault detection with an accuracy between 99.481% and 100%. Wen et al. [15] proposed a new deep transfer learning based on a sparse auto-encoder for motor-bearing fault detection with 99.82% accuracy. Wen et al. [16] proposed a new hierarchical convolutional neural network (HCNN), with an accuracy between 96.1% and 99.82%. These works are summarized in Table 1.

According to surveys by García et al. [17] and Pan and Yang [18], the following research gaps remain in the further improvement of the efficiency of data-driven algorithms:

(1) It is imperative to design suitable preprocessing technologies for monitored data to ensure the best diagnosis accuracy and efficiency.

(2) Deep learning algorithms usually require a long training time to achieve high accuracy. It is also difficult and costly to acquire sufficient faulty data patterns for algorithm training.

(3) Thresholds to classify different faults are usually preset by experienced engineers. This is not an optimal solution for the increasingly dynamic environments that exist in modern production processes.

3. System framework

Power data from the control motors in CNC machines can indicate the working conditions of the machine and tooling [19,20]. Moreover, in comparison with vibration sensors or acoustic sensors [21,22], power sensors are more cost-effective in deployment. Thus, in this system, which is empowered by a wireless sensor network (WSN) mounted onto CNC machines, power data are collected to support anomaly diagnosis of the machine and tooling [5]. The structure of the system is illustrated in Fig. 1. The functions are explained below:

(1) **Data repository:** A big data infrastructure has been configured and deployed for collecting, storing, and visualizing real-time monitored data during machining processes [5].

(2) **Data preprocessing:** Considering the veracity of the monitored data, preprocessing mechanisms have been designed. These mechanisms include: ① Partitioning the data into time-series datasets according to individual machining processes, ② using a Gaussian kernel model [23,24] to de-noise the fluctuated information from the monitored data in order to facilitate further processing, ③ applying normalization to ensure that the scale of the monitored data is suitable for analysis, and ④ performing alignment based on a cross-covariance method [5] to rescale the power data with standard and faculty reference patterns in order to facilitate anomaly identification.

(3) **Feature representation and anomaly identification:** A set of features has been defined to support anomaly analysis and diagnosis. Thresholds for feature comparisons are used for anomaly identification. The system is open to new anomalies and is dynamically updated during machining processes.

(4) **Threshold optimization:** An optimization algorithm has been designed to determine optimized thresholds based on historically monitored data.

Table 1
Summary of proposed methods.

Method	De-noising, normalization, and alignment	Application	Accuracy	Disadvantage	Ref.
Deep Boltzmann machine	No	Industrial diagnosis	29.85%–93.67%	High computation time	[8]
Modified SVM and RFE	No	Steel plates diagnosis	80.74%	Crippled detection rate	[9]
CMFE and ESVMs	No	Rolling bearing diagnosis	100.00%	Crippled detection rate	[10]
CNN	No	Chemical process diagnosis	91.00%	High computation time	[11]
Decision tree	No	Milling tool diagnosis	81.00%	Structure and threshold difficult to define	[12]
Extreme learning machine	No	Aero-engine diagnosis	90.00%	High computation time	[13]
CNN based on LeNet-5	No	Motor bearing, etc.	99.481%–100.00%	High computation time	[14]
Sparse auto-encoder	No	Motor bearing	99.82%	High computation time	[15]
HCNN	No	Motor bearing, etc.	96.10%–99.82%	High computation time	[16]

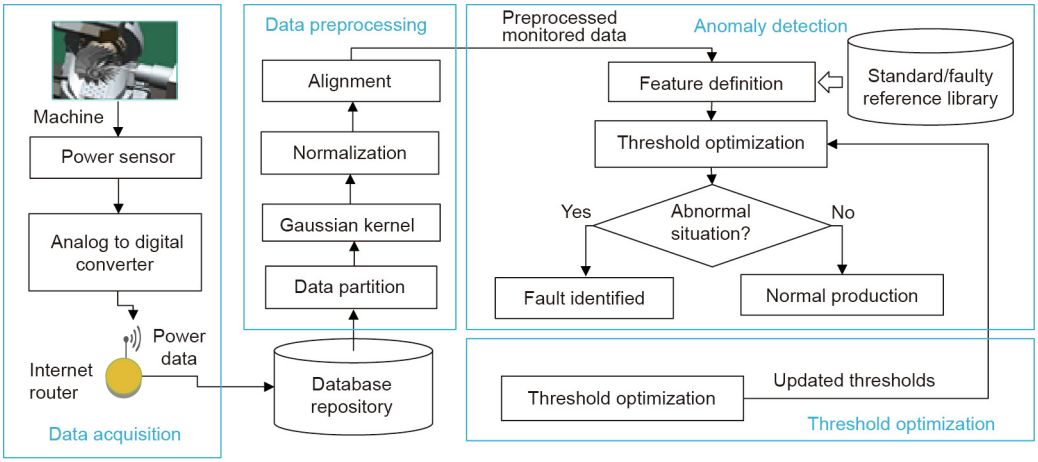

Fig. 1. System framework of CNC machining process.

4. Preprocessing monitored data

4.1. Partitioning monitored data

Monitored power data acquired during machining processes will be used for diagnosis. Power is calculated based on the following formula:

$$P(i) = [I_1(i) + I_2(i) + I_3(i)] \times V \times Factor \quad (1)$$

where $P(i)$ is the ith point of the power data along the time axis (x axis); $I_1(i)$, $I_2(i)$, and $I_3(i)$ are the three-phase currents of the power; V is the voltage of the power; and *Factor* is the quality factor of the power.

It is time-intensive and ineffective to apply analysis on all the power data collected during machining. To facilitate analysis, the monitored data are first partitioned based on machine-specific power levels to represent individual setups of the machining processes. The following steps are then applied on the partitioned monitored data to facilitate analysis further.

4.2. De-noising and smoothing monitored data

In general, monitored power data fluctuate as a result of the noises in the signals. In order to extract features effectively, it is essential to de-noise and smooth the monitored data. In this research, a Gaussian kernel-based model is designed for de-noising data. The robustness of the Gaussian kernel has been proved by Feng et al. [23] and Rimpault et al. [24]. Here, the monitored data are smoothed by a convolution computation with the Gaussian kernel. The ith point of the de-noised and smoothed power data $P_\sigma(i)$ is calculated below:

$$P_\sigma(i) = \frac{\sum_{j=1}^{n}\left[P(i) \times g_\sigma(x_j)\right]}{\sum_{j=1}^{n}\left[g_\sigma(x_j)\right]} \quad (2)$$

where n represents the total points in P (the power data); x_j stands for the jth point of P along the x axis (time); and $g_\sigma(x_j)$ is the Gaussian kernel for the jth point with kernel width σ.

An example of the above process is illustrated in Fig. 2.

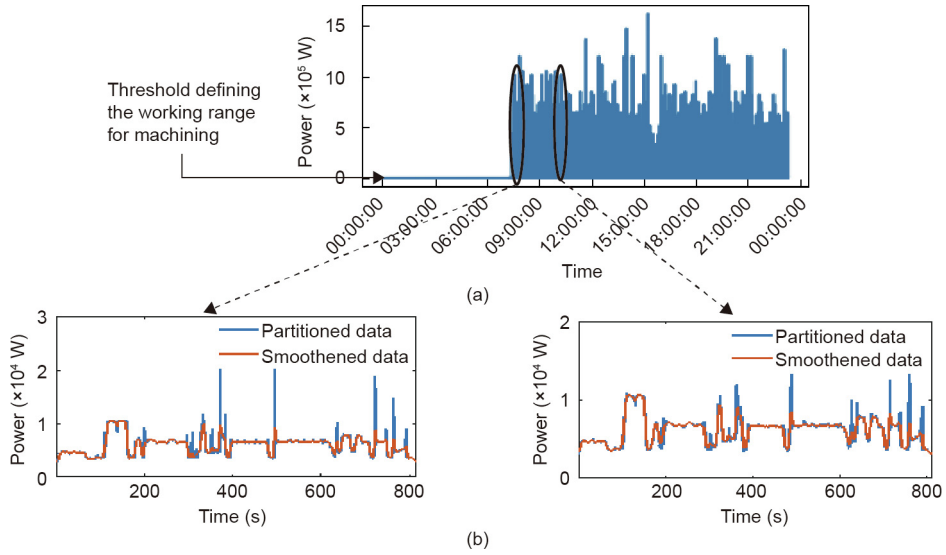

Fig. 2. Example of data partitioning and de-noising of the monitored data. (a) Acquired power data in a single day (31 May 2016); (b) power patterns (in red) for two individual processes after partitioning and de-noising.

4.3. Normalizing monitored data

Normalization is applied to the monitored data to ensure the proper scale of the data, in order to facilitate feature extraction from the data (e.g., the value of kurtosis, described in Section 5, will be extremely high without the normalization process):

$$\mathrm{NP} = P_\sigma / P_{\mathrm{ref}} \tag{3}$$

where NP is the normalized power data; P_σ is the original power data; and P_{ref} is the reference power data of the machine.

4.4. Aligning monitored data

Under practical manufacturing conditions, there may be time delays or deviations in the partitioned monitored data when machining a component, which result in misalignment with a standard reference (i.e., the power pattern when machining the same component under normal working conditions). Cross-covariance between the monitored data and the standard reference ($\sigma_{s_{\mathrm{cutting}}s_{\mathrm{standard}}}$) is applied to identify the time delay [5].

$$\sigma_{s_{\mathrm{cutting}}s_{\mathrm{standard}}}(T) = \frac{1}{N-1} \times \sum_{t=1}^{N} \left[P_{\mathrm{standard}}(t) - \mu_{\mathrm{cutting}}\right]\left[P_{\mathrm{cutting}}(t+T) - \mu_{\mathrm{standard}}\right] \tag{4}$$

where P_{standard} and P_{cutting} are the standard reference and partitioned monitored data, respectively; μ_{standard} and μ_{cutting} are the means of the time-series; N is the smaller number of the two datasets; and t and T are time deviation and standard time, respectively.

The time delay can be calculated by the following formulas:

$$X_{\mathrm{coef}} = \frac{\sigma_{s_{\mathrm{cutting}}s_{\mathrm{standard}}}(T, T = 1:N)}{\sqrt{\sigma_{s_{\mathrm{cutting}}s_{\mathrm{cutting}}}(0)\,\sigma_{s_{\mathrm{standard}}s_{\mathrm{standard}}}(0)}} \tag{5}$$

The time difference can be calculated when X_{coef} is at its maximum:

$$Difference = T, \text{ when } X_{\mathrm{coef}} \text{ is maximum} \tag{6}$$

Therefore, the aligned monitored data will be:

$$P_{\mathrm{cutting\_alignment}} = P_{\mathrm{cutting}}(t + Difference) \tag{7}$$

The same procedure is used to align the monitored data with a faulty reference P_{fault} (i.e., the power pattern when machining the same component under abnormal conditions) by replacing P_{standard} with P_{fault} in the above formulas.

5. Anomaly-detection process

During machining processes, *Feature* is defined to represent the difference between the preprocessed monitored data and the standard reference (i.e., the data pattern of machining the same component under normal working conditions). Standard references are collected during component machining under good working conditions. The diagnosis procedure, which is depicted in Fig. 3, includes the following steps:

(1) Features are represented based on a matrix of the absolute mean, kurtosis, and crest factor. The relevant definitions are provided in Table 2. In Table 2, $Feature_m$ is calculated from each piece of preprocessed monitored data and its standard reference. $Feature_2$–$Feature_m$ are calculated from each piece of preprocessed monitored data and its faulty references (where m represents an anomaly type).

(2) A series of thresholds are defined. $Threshold_1$ is used to determine normal or abnormal conditions by comparing $Feature_1$ and $Threshold_1$. $Threshold_2$–$Threshold_m$ are used to classify the type of anomaly by comparing $Feature_2$–$Feature_m$ and $Threshold_2$–$Threshold_m$, respectively. A new anomaly type will be updated into the database of the system if there is no match of anomalies.

(3) The above thresholds are optimized periodically by an FFO algorithm based on the latest historical data.

In this research, abnormal working conditions are defined based on the following rules [5]:

- **Tool wear:** Power range shifts vertically significantly, but the power range during the idle stage remains the same.
- **Tool breakage:** Power increases to a peak value and goes back to the air-cutting power range.
- **Spindle failure:** Power has a sudden peak and an increased power range during machining and idle stages.

Based on the rules and historical data, three thresholds for the above abnormal conditions can be defined: $Threshold_2$ for judging tool wear, $Threshold_3$ for judging tool breakage, and $Threshold_4$ for judging spindle failure. The process to determine the optimal thresholds is described in the following section.

6. Threshold optimization

As discussed in our previous research [5], the overall detection accuracy can be decided by four factors: True Positive (*TP*), False Positive (*FP*), True Negative (*TN*), and False Negative (*FN*). *TP* indicates that an abnormal condition is correctly identified as abnormal; *FP* indicates that a normal condition is incorrectly identified as abnormal; *TN* indicates that a normal condition is correctly identified as normal; and *FN* indicates that an abnormal condition is incorrectly identified as normal. Based on these four factors, *Precision*, *Recall*, and *F* are introduced to evaluate the overall detection accuracy [26]:

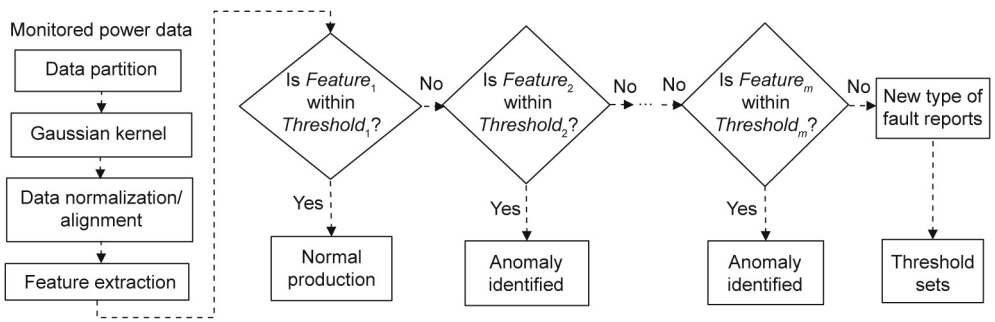

Fig. 3. The anomaly diagnosis process.

Table 2
Definitions of features and thresholds for standard reference, faulty reference, and monitored data [25].

Variables	Difference between	
	Preprocessed monitored data and standard reference	Preprocessed monitored data and fault reference
Absolute mean	$Mean_{standard} = \frac{1}{N}\sum_{t=1}^{N}\lvert P_{standard} - P_{cutting}\rvert$	$Mean_{fault} = \frac{1}{N}\sum_{t=1}^{N}\lvert P_{fault} - P_{cutting}\rvert$
Kurtosis	$k_{standard} = \frac{1}{N}\sum_{t=1}^{N}(P_{standard} - P_{cutting})^4$	$k_{fault} = \frac{1}{N}\sum_{t=1}^{N}(P_{fault} - P_{cutting})^4$
Crest factor	$C_{standard} = \max(\lvert P_{standrad} - P_{cutting}\rvert)/\sqrt{\frac{1}{N}\sum_{t=1}^{N}(P_{standard})^2}$	$C_{fault} = \max(\lvert P_{fault} - P_{cutting}\rvert)/\sqrt{\frac{1}{N}\sum_{t=1}^{N}(P_{fault})^2}$
Features	$Feature_1 = (Mean_{standard} k_{standard} C_{standard})$	$Feature_{2-n} = (Mean_{fault} k_{fault} C_{fault})$
Thresholds	$Threshold_1 = (V_{Mean_{standard}} V_{k_{standard}} V_{C_{standard}})$	$Threshold_{2-n} = (V_{Mean_{fault}} V_{k_{fault}} V_{C_{fault}})$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F = 2\frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

where *Precision* represents the proportion of correctly identified abnormal conditions against all the identified abnormal conditions; *Recall* is the proportion of correctly identified abnormal conditions against all the actual abnormal conditions; and *F* measures the overall accuracy of detection. The higher the *F* score is (where a good score is close to 1), the better the overall accuracy of detection will be.

TP, *FP*, *TN*, and *FN* are affected by the values of the four thresholds (i.e., $Threshold_1$, $Threshold_2$, $Threshold_3$, $Threshold_4$). Therefore, the value choices of the thresholds affect the *F* score.

In this research, the thresholds are optimized using an FFO algorithm through historically monitored data rather than by depending on the experience of experts. An FFO is able to avoid local optima, and has better performance than some other mainstream optimization algorithms [27,28]. In this algorithm, swarm centers are initialized to conduct a parallel search (in this research, each center is modeled as a vector of the four thresholds—i.e., $Threshold_1$–$Threshold_4$). Around each swarm center, random solutions called "fruit flies" are generated. Through smell- and vision-based strategies to calculate fitness and swarm center selection, respectively (details explained in Steps 3 and 4). The computation is iterated toward optimization.

The optimization objective is to identify the most appropriate thresholds that lead to the maximum *F* score:

$$Vector(Threshold_1, Threshold_2, Threshold_3, Threshold_4) \rightarrow F \tag{11}$$

The optimization process is described below (improvements to the typical FFO algorithm are provided in Steps 2 and 5):

Step 1. Set the maximum number of iterations T_{max}, the population size of the swarm centers *v*, and the population size of the fruit flies around each swarm center *k*.

Step 2. Randomly generate fruit flies around each swarm center according to the following formula:

$$Vector_{sub} = Vector_{center} \pm \alpha \times rand \tag{12}$$

where $Vector_{center}$ and $Vector_{sub}$ are the vectors of a swam center and of the sub-population of fruit flies around each swarm center, respectively, α is the boundary determining the search distance of the fruit flies around each swarm center, and *rand* represents a random number.

Step 3. Conduct a smell-based search to calculate the smell concentration (i.e., fitness) for each fruit fly.

Step 4. Conduct a vision-based search to replace the original swarm center with a fruit fly in the sub-population with the best fitness and direct the sub-population to search further.

Step 5. In a typical FFO algorithm, the search distance is always constant, which will make the search difficult to converge when the fruit fly is near to the solution. Therefore, in order to improve the algorithm, the search distance is decreased when the fitness is not improved in five iterations. (This improves the convergence speed, as the swarm can approach the solution in an easier way when the swarm is close to a solution [29].)

$$\alpha_{new} = \alpha - \frac{\alpha \times (T_{max} - 1) \times 0.8}{T_{max}} \tag{13}$$

where α_{new} is the decreased search distance when approaching an optimum result.

Step 6. Repeat the steps above until solution convergence or the maximum number of iterations T_{max} is reached.

7. Case studies

Sponsored by the EU Smarter and Cloudflow projects, a WSN was developed and deployed on the shop floor of a company in the UK. The company specializes in high-precision machining for automotive, aerospace, and tooling applications. In this case, a five-axis milling machine, MX520, was monitored. For over six months, power data (more than 10 GB) was collected and stored in a local database. A big data infrastructure based on an open source platform, Hadoop, was developed to manage the huge amount of data and to accelerate the data processing.

A part of the production line is illustrated in Fig. 4. Three current sensors (one for each phase) are clamped on the main supply of the CNC machine. The data rate for one sensor is one sample per second; hence, one sample per second is transferred to the Hadoop data server via the Wi-Fi on the shop floor. Power is then calculated

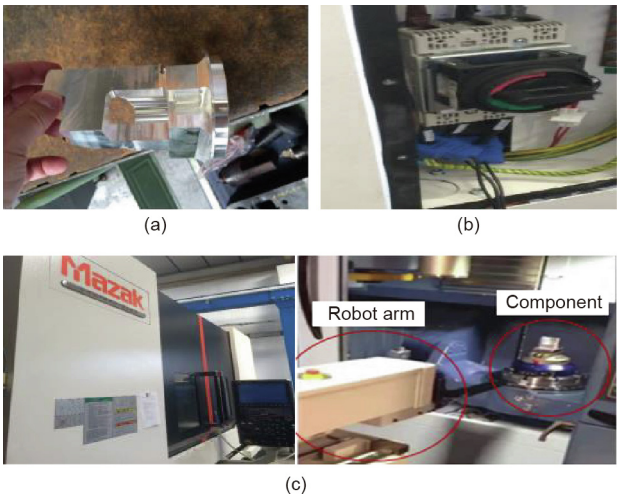

Fig. 4. CNC machining processes. (a) Machined part; (b) power measurement; (c) Mazak machine and machining processes.

based on the three-phase current, the 220 V voltage, and a power factor of 0.82.

In this case study, the FFO algorithm was designed to identify the best thresholds leading to the highest *F* score based on historical data. Table 3 shows the benchmark results for this optimization process. The optimized thresholds are (0.192, 0.032, 0.287) for $Threshold_1$, (0.632, 0.410, 0.652) for $Threshold_2$ as tool wear, (3.698, 75.363, 10.737) for $Threshold_3$ as tool breakage, and (2.412, 1.081, 0.921) for $Threshold_4$ as spindle failure. The FFO algorithm can achieve the optimal result in 23 iterations, and converges the most quickly when compared with other benchmark algorithms. It can achieve an *F* score of 1, which means that the optimized thresholds can achieve a 100% true detection rate based on the historical data. Some examples of anomaly detection and identification are explained below.

7.1. Normal production

Fig. 5 shows an analysis of the monitored data for anomaly detection. The extracted $Feature_1$ is (0.147, 0.004, 0.113), which is smaller than $Threshold_1$ (0.192, 0.032, 0.287) (the definitions of *Feature* and *Threshold* are in Table 2). Therefore, it can be classified as normal.

7.2. Anomaly situation: Tool wear

For the monitored data showed in Fig. 6(a), $Feature_1$ is (0.206, 0.042, 0.295), which is higher than $Threshold_1$ (0.192, 0.032, 0.287). Therefore, the production is classified as a fault. Thus, an anomaly diagnosis is made (Fig. 6(b)). $Feature_2$ is (0.171, 0.058, 0.250), which is smaller than $Threshold_2$ (0.632, 0.410, 0.652). Therefore, the anomaly can be classified as tool wear.

7.3. Anomaly situation: Tool breakage

For the monitored data showed in Fig. 7(a), $Feature_1$ is (0.460, 41.532, 2.303), which is higher than $Threshold_1$ (0.192, 0.032, 0.287). Therefore, the production is a fault. An anomaly diagnosis is made (Fig. 7(b)). $Feature_3$ is (1.039, 61.512, 1.744), which is smaller than $Threshold_3$ (3.698, 75.363, 10.737). Therefore, the anomaly can be classified as tool breakage.

7.4. New abnormal situation: Long-time air cutting

Fig. 8 shows an analysis of the monitored data. $Feature_1$ is (0.492, 0.441, 0.379), which is higher than $Threshold_1$ (0.192, 0.032, 0.287). Therefore, the production is a fault. However, there is no faulty reference similar to this data pattern. In this case, therefore, the data was reported to the shop floor engineers. It was found that the machine was accidently left air cutting. The pattern was then saved in order to update the faulty references.

Table 3
Comparison of optimization algorithms.

	FFO	GA	SA
Iterations to reach optimization	23	44	51
Optimized *F*	1	1	1

GA: genetic algorithms; SA: simulated annealing.

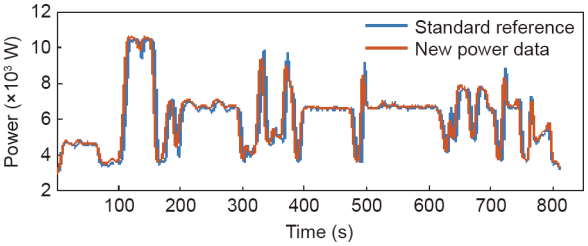

Fig. 5. Monitored data indicating a normal production condition.

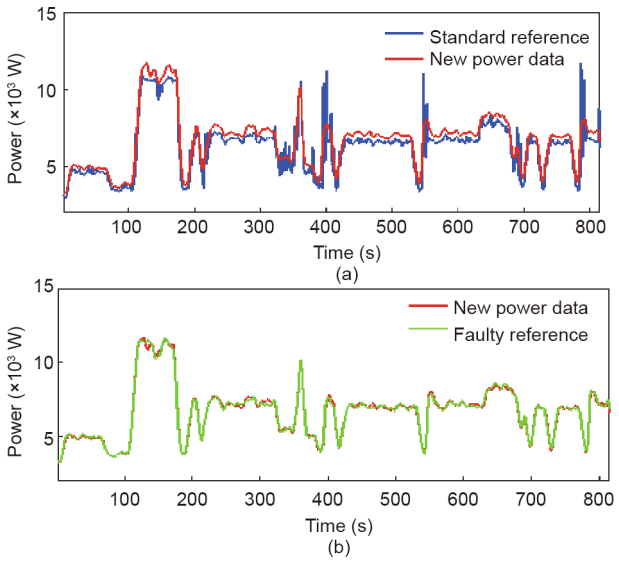

Fig. 6. Tool wear detection. (a) Fault identification; (b) fault classification.

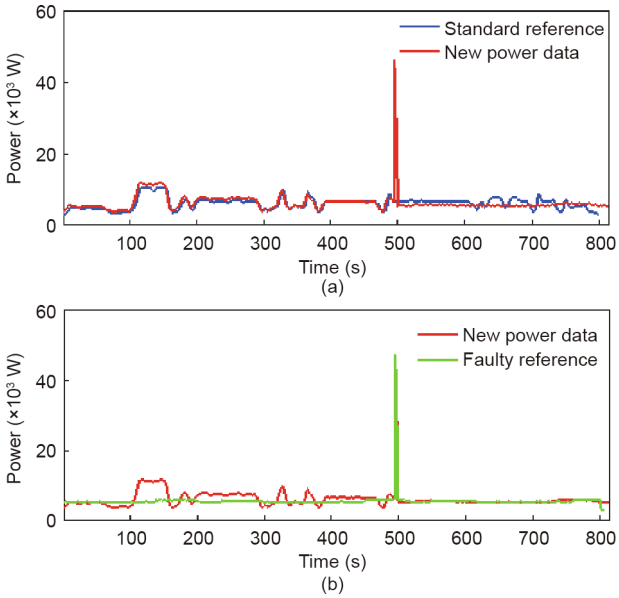

Fig. 7. Broken tool detection. (a) Fault identification; (b) fault classification.

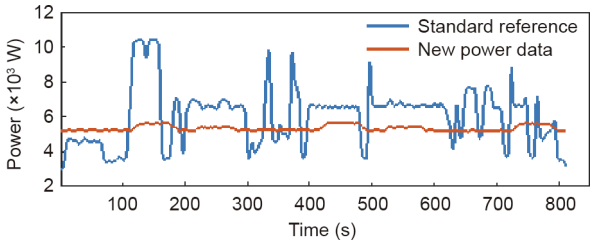

Fig. 8. New abnormal data detection.

8. Conclusions

In this research, a data-driven anomaly analysis was developed. The system was deployed in a machining company for verification under practical machining conditions. The innovations of this research are as follows:

(1) Preprocessing mechanisms, including de-noising, data normalization, and alignment, were developed to address the veracity issue in monitored data.

(2) An FFO algorithm was designed to identify optimal anomaly thresholds in order to achieve more accurate detection during dynamic machining processes.

Further investigations will be carried out in future to solidify the reliability of the system, including the following: ① A different data sampling rate will be tested to find the best system accuracy and efficiency. Furthermore, different data sources will be considered (e.g., vibration, force data, etc.) by using data fusion to enhance the prediction results. ② We will consider designing effective and efficient deep learning algorithms and computation architectures (e.g., transfer learning algorithms and edge architectures for recurrent neural networks (RNNs) and long short-term memory recurrent neural networks (LSTM RNNs), etc.), to further improve the system performance.

Acknowledgement

The authors acknowledge the funding from the EU Smarter project (PEOPLE-2013-IAPP-610675).

Compliance with ethics guidelines

Y.C. Liang, S. Wang, W.D. Li, and X. Lu declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Bayar N, Darmoul S, Hajri-Gabouj S, Pierreval H. Fault detection, diagnosis and recovery using artificial immune systems: a review. Eng Appl Artif Intell 2015;46:43–57.
[2] Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: part I: quantitative model-based methods. Comput Chem Eng 2003;27(3):293–311.
[3] Aydin I, Karakose M, Akin E. Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. Expert Syst Appl 2010;37(7):5285–94.
[4] Yang H, Li T, Hu X, Wang F, Zou Y. A survey of artificial immune system based intrusion detection. Sci World J 2014;2014:156790.
[5] Wang S, Liang YC, Li WD, Cai XT. Big data enabled intelligent immune system for energy efficient manufacturing management. J Clean Prod 2018;195:507–20.
[6] Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, et al. Cloud-enabled prognosis for manufacturing. CIRP Ann 2015;64(2):749–72.
[7] Lee J, Wu F, Zhao W, Ghaffari M, Liao L, Siegel D. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. Mech Syst Signal Process 2014;42(1–2):314–34.
[8] Hu G, Li H, Xia Y, Luo L. A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis. Comput Ind 2018;100:287–96.
[9] Tian Y, Fu M, Wu F. Steel plates fault diagnosis on the basis of support vector machines. Neurocomputing 2015;151:296–303.
[10] Zheng J, Pan H, Cheng J. Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines. Mech Syst Signal Process 2017;85:746–59.
[11] Wu H, Zhao J. Deep convolutional neural network model based chemical process fault diagnosis. Comput Chem Eng 2018;115:185–97.
[12] Madhusudana C, Kumar H, Narendranath S. Fault diagnosis of face milling tool using decision tree and sound signal. Materials Today Proc 2018;5(5):12035–44.
[13] Lu F, Jiang J, Huang J, Qiu X. Dual reduced kernel extreme learning machine for aero-engine fault diagnosis. Aerosp Sci Technol 2017;71:742–50.
[14] Wen L, Li X, Gao L, Zhang Y. A new convolutional neural network-based data-driven fault diagnosis method. IEEE Trans Ind Electron 2018;65(7):5990–8.
[15] Wen L, Gao L, Li X. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans Syst Man Cybern Syst 2019;49(1):136–44.
[16] Wen L, Li X, Gao L. A new two-level hierarchical diagnosis network based on convolutional neural network. IEEE Trans Instrum Meas. Forthcoming 2019.
[17] García S, Luengo J, Herrera F. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowl Base Syst 2016;98:1–29.
[18] Pan S, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22(10):1345–59.
[19] Liu Z, Guo Y, Sealy M, Liu Z. Energy consumption and process sustainability of hard milling with tool wear progression. J Mater Process Technol 2016;229:305–12.
[20] Sealy M, Liu Z, Zhang D, Guo Y, Liu Z. Energy consumption and modeling in precision hard milling. J Clean Prod 2016;135:1591–601.
[21] Stoney R, Donohoe B, Geraghty D, O'Donnell G. The development of surface acoustic wave sensors (SAWs) for process monitoring. Procedia CIRP 2012;1:569–74.
[22] García Plaza E, Núñez López PJ. Application of the wavelet packet transform to vibration signals for surface roughness monitoring in CNC turning operations. Mech Syst Signal Process 2018;98:902–19.
[23] Feng Z, Zuo M, Chu F. Application of regularization dimension to gear damage assessment. Mech Syst Signal Process 2010;24(4):1081–98.
[24] Rimpault X, Bitar-Nehme E, Balazinski M, Mayer J. Online monitoring and failure detection of capacitive displacement sensor in a Capball device using fractal analysis. Measurement 2018;118:23–8.
[25] Xia M, Li T, Xu L, Liu L, De Silva C. Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. IEEE/ASME Trans Mechatron 2018;23(1):101–10.
[26] Zheng X, Wang L, Wang S. A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. Knowl Base Syst 2014;57:95–103.
[27] Liang Y, Lu X, Li W, Wang S. Cyber physical system and big data enabled energy efficient machining optimisation. J Clean Prod 2018;187:46–62.
[28] Powers D. Evaluation: from precision, recall and *F*-measure to ROC, informedness, markedness & correlation. J Mach Learn Technol 2011;2(1):37–63.
[29] Du T, Ke X, Liao J, Shen Y. DSLC-FOA: improved fruit fly optimization algorithm for application to structural engineering design optimization problems. Appl Math Model 2018;55:314–39.