Fog computing and convolutional neural network enabled prognosis for machining process optimization

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

Original citation & hyperlink:
Liang, Y, Li, W, Lu, X & Wang, S 2019, 'Fog computing and convolutional neural network enabled prognosis for machining process optimization', Journal of Manufacturing Systems, Part A, pp. 32-42.
https://dx.doi.org/10.1016/j.jmsy.2019.05.003

DOI 10.1016/j.jmsy.2019.05.003
ISSN 0278-6125

Publisher: Elsevier

Fog Computing and Convolutional Neural Network Enabled Prognosis for Machining Process Optimization

Y.C. Liang, W.D. Li[+], X. Lu, S. Wang
Faculty of Engineering, Environment and Computing, Coventry University, CV1 5FB, U.K.
[+] Email of corresponding author: weidong.li@coventry.ac.uk

Abstract

Cloud enabled prognosis systems have been increasingly adopted by manufacturing industries. The effectiveness of the cloud systems is, however, crippled by the high latency of data transfer between shop floors and the cloud. To overcome the limitation, this paper presents an innovative fog enabled prognosis system for machining process optimization. The system functions include: (1) dynamic prognosis - Convolutional Neural Network (CNN) based prognosis is implemented to detect potential faults from customized machining processes. Pre-processing mechanisms of the CNN are designed for partitioning and de-noising monitored signals to strengthen the performance of the system in practical manufacturing situations; (2) an innovative fog enabled prognosis architecture for machining process optimization – it consists of a terminal layer, a fog layer and a cloud layer to minimize data traffic and improve system efficiency. Under the architecture, monitored signals during machining collected on the terminal layer are processed using the trained CNN deployed on the fog layer to efficiently detect abnormal situations. Intensive computing activities like training of the CNN and system re-optimization responding to detected faults are carried out dynamically on the cloud layer to leverage its computation powers. The system was validated in a UK machining company. With the system deployment, the efficiency of energy and production was improved for 29.25% and 16.50% on average. In comparison with a cloud system, this fog system achieved 70.26% reduction in the bandwidth requirement between shop floors and cloud, and 47.02% reduction in data transfer time. This research, sponsored by EU projects, demonstrates that industrial artificial intelligence can facilitate smart manufacturing practices effectively.

Keywords: Fog computing, Convolutional Neural Network, prognosis, machining process

1. Introduction

During Computerized Numerical Control (CNC) machining processes, deviations on dimensional accuracy and surface roughness would lead to product defects or wastes. The root causes of deviations could be from poor tooling or abnormal equipment conditions, resulting in unsatisfactory precision, leaving scratch marks on machined components, generating severe vibration or high temperature during machining. Thus, it is significant to design prognosis systems to perform efficient analysis on condition data acquired from machines and processes in order to detect potential failures in early stages, leading to preventive maintenance and adaptive optimization [1]. Recently, cloud enabled prognosis systems have been developed by taking advantage of the computing and storage capabilities of cloud centers [2], as well as being reinforced by the state-of-the-art artificial intelligence (AI) algorithms to improve the accuracy and reliability of prognosis analysis [3]. Conversely, the performance of the systems is hindered by the high latency of data transfer between shop floors and the cloud. To overcome the limitation of the cloud model, a fog (From cOre to edGe) computing model based on the "divide and conquer" strategy has been researched [3-5]. Under the model, most of computing and intelligent reasoning can be conducted on edge devices locally for efficient analysis. Only necessary information is transferred to the cloud for intensive computation. Therefore, the bandwidth requirement is minimized, and the agility of detecting fault features during early stages is enhanced. Also, the data leakage issue pertaining to uploading all operation data into cloud centers is mitigated.

Though several fog enabled manufacturing systems have been recently reported ([6-9]), the systems are ineffective to meet the requirements of practical manufacturing processes. Research gaps and industrial requirements on fog enabled prognosis are summarized below:

- In practical manufacturing environments, fault features are hidden in fluctuating signals and disturbed by non-Gaussian noises in the signals. The occurrence of faults is uncertain in long-term process execution, requiring effective AI algorithms (e.g., deep learning) to mine fault features from large-scale monitored data. The above processes generate intensive computation during prognosis analysis.
- It is imperative to design a suitable fog enabled prognosis architecture to facilitate the intensive computation of related algorithms to support real-world applications. To leverage the capacities of fog devices effectively, it is vital to design pre-processing mechanisms to partition and de-noise monitored signals to facilitate the intensive computation of prognosis. Industrial trials and case studies will be essential to verify the advantage of the model to implement industrial artificial intelligence.

To address the above requirements, this paper presents an innovative fog and deep learning enabled system for machining process prognosis and optimization. The functions and contributions of the system are summarized below:

- Convolutional Neural Network (CNN) based prognosis is designed to achieve high detection rates for customized and varying machining processes. In this aspect, the novelty includes: (1) pre-processing mechanisms for the CNN to de-noise monitored data and enhance the computing efficiency effectively, and (2) suitable structures of the CNN designed to improve computation efficiency of prognosis.

- To optimize data transfer, monitored data acquired on equipment are processed on a fog layer to identify faults efficiently using the trained CNN deployed locally. Intensive computations such as the training process of the CNN and re-scheduling optimization to address abnormal situations are carried out on a cloud layer.
- The system was deployed into a UK machining company. Case studies were carried out to demonstrate the applicability of the system to manufacturing practices. With the deployed system, the energy and production efficiency was improved approximately for 29.25% and 16.50%. In comparison with cloud solutions, 70.26% reduction on bandwidth requirement, and 47.02% reduction of time spent for data transfer were achieved. To the best of our knowledge, it is the first time to develop a fog enabled prognosis system that is seamlessly integrated with deep learning algorithms and validated in practical machining processes.

2. Literature Survey

Generally, developed fog architectures are comprised of three layers: a terminal layer, a fog layer and a cloud layer [5]. The terminal layer is close to physical environment with the Internet of Thing (IoT) devices, such as sensors, smart readers, etc. The fog layer, which consists of Wi-Fi routers, gateways, etc., is usually located on the edge of the sensor network. A gateway has computing capabilities to process majority of acquired data with low latency, so that only the most useful data are sent to the cloud layer for complex computation. In recent years, there are increasingly reported research works of fog computing supported manufacturing. Wu et al. [7] proposed a fog framework for manufacturing optimization. The random forest algorithm were embedded to predict tool wear. Lin and Yang [8] developed a fog system to process data collected by sensors at logistics centers. The discrete monkey algorithm and genetic algorithm were applied to identify the best good-quality solutions efficiently. Wan et al. [9] designed a fog structure to optimize the energy consumption and workload of machines in a manufacturing shop floor. O'Donovan et al. [10] designed a fog architecture integrating machine learning algorithms. The execution time of both fog and cloud layers were analyzed to showcase the benefits of designing the fog layer. A particle swarm optimization algorithm was implemented on fog. Mohamed et al. [11] developed a fog structure to support multi-robot applications. However, the aforementioned systems in general lack validation based on complex industrial scenarios. Relevant analyses and results are mainly based on simulation.

In manufacturing, diagnosis and prognosis on manufacturing equipment and processes have been an active research field. In recent years, owing to the rapid progress of artificial intelligence, it has become an active trend to apply deep learning and Big Data analytics into this research field (several recent surveys were reported in [1, 2, 12-14]). Signals from manufacturing systems, such as current, vibration or acoustic emission, were continuously collected, and Big Data analyses were carried out in the time domain, the frequency domain, or the time-frequency domain. The principal component analysis (PCA) and k-nearest neighbor (k-NN) method were used to detect faults for bearing systems ([15]) and industrial processes [16], or the grading level of manufacturing systems [17]. The support vector machine (SVM) method was applied for fault prediction of gearboxes [18] and ship propulsion systems [19]. Zhang et al. [20] developed a fault detection system for wind turbines by using the random forest algorithm in combination with the extreme gradient boosting (XGBoost) algorithm. Abdullah [21] utilized discrete wavelet transform based artificial neural networks for fault detection of transmission lines. In [22], frequency features were extracted from vibration signals using a state-space model. A Cosine distance was designed to compare healthy frequency features and monitored features to detect early faults for a machining center. In recent years, deep learning algorithms, including CNN, RNN (Recurrent Neural Network), Stacked Autoencoders, etc., were applied for fault detection of manufacturing equipment or processes. Ince et al. [23] designed a 1-D CNN for motor fault diagnosis. Xia et al. [24] designed a CNN for fault detection for two cases, one for the motor drive system provided by the Case Western Reserve University, and another one for gearboxes. Features from vibration signals in the time and frequency domains were summarized for analysis.

Based on the above survey, it is identified that prognosis systems would be more effective in processing complex industrial situations by integrating the fog model and deep learning algorithms seamlessly. Suitable mechanisms and system structures need to be developed to improve the computation efficiency of the system.

3. System Framework and Workflow

For the system, power signals will be collected for prognosis and optimization of machining processes. According to the research of Liu et al. [27] and Sealy et al. [28], power signals from CNC machines can indicate the working and tooling conditions of machines. Power sensors are easy to deploy and the data sampling rate is much lower than that of vibration or acoustic sensors, which has been verified by experiments [35]. Meanwhile, energy consumption for machining processes can be also calculated based on power signals in order to achieve sustainable optimization of the machining system. Therefore, for this research, power signals, which are analyzed in the time domain, are used for both abnormal condition detection and multi-objective (including energy-efficient) re-scheduling optimization for detected faults.

As aforementioned analysis, the system has been designed based on the fog computing paradigm to provide a smart solution for dynamic prognosis and optimization of machining processes. The system is comprised of three layers to optimize the computation efficiency and latency of data transmission. Data transfer in the system between layers are through the MQTT protocol, taking the advantage of its lightweight communication [29], and information exchange in dual-direction simultaneously [30]. The system structure is illustrated in Fig. 1. The three layers and a coordinator between layers are briefly introduced below. Details will be elaborated in the following Section 4.

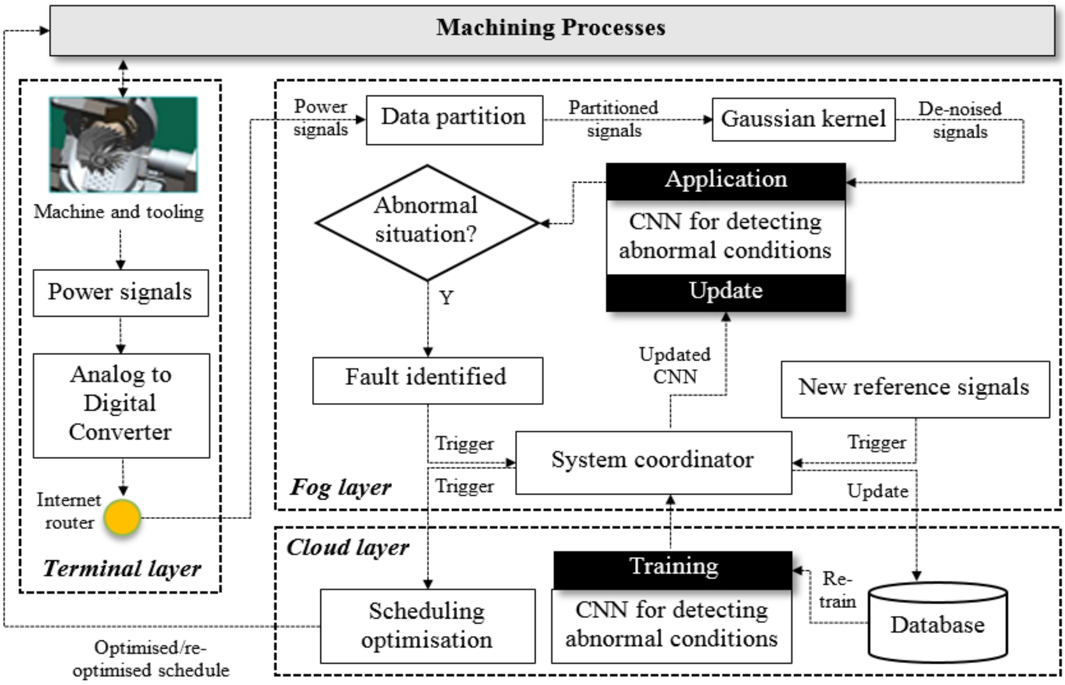


Fig. 1. Fog enabled system for prognosis and scheduling re-optimization.

(1) Terminal layer: This layer is integrated with physical machine equipment via sensor devices, circuits and routers. Machining processes will follow an optimized schedule sent from the cloud layer. During entire machining processes, power signals of machines are continuously collected via power sensors and transmitted to a fog layer for further processing. When abnormal conditions of machines and tooling (e.g., severe tool wear, tool breakage, spindle failure) are detected on the fog layer, a re-schedule optimization will be triggered on the cloud layer and the generated optimized schedule will be sent to this physical layer for dynamic production adjustment.

(2) Fog layer: A trained CNN is deployed on the fog layer for efficient detection of abnormal conditions. To facilitate the efficiency and effectiveness of computing in the CNN, power signals during machining processes are first partitioned into individual stages of the machining process for each component. An algorithm of Gaussian kernel smoothness is embedded for de-noising the signals to facilitate detection. The CNN will be trained on the cloud layer and the training process will be updated for new patterns of signals when new types of components are arranged for machining in production lines. Based on the design, abnormal situations are detected using the trained CNN on fog without relatively long transmission time of the power signals to the cloud layer for judgement and decision making. Thus, signal data pertaining to operations will be kept within companies. Machines can be stopped quickly for maintenance and adjustment when potential abnormal conditions occur.

(3) Cloud layer: The cloud layer has databases to store reference signals for each machined components (called reference signals in the paper) and typical abnormal conditions for such components. The reference signals for machined components are built based on historical data [31], [32], [35]. The CNN is re-trained when new components are arranged and reference signals are received. The trained CNN is transmitted back to the fog layer for deployment after the update. On the cloud layer, there is a multi-objective optimization algorithm to be triggered for re-scheduling when necessary.

(4) System coordinator: It is arranged between the fog and cloud layers to coordinate tasks as follows: (i) updating the knowledge database on the cloud layer for reference signals of newly machined components; (ii) re-training of the CNN using the new references; (iii) updating the trained CNN on the fog layer; (iv) triggering the scheduling optimization for the situation of abnormal situations during machining processes, and sending the optimized schedule to the terminal layer for production adjustment.

4. Detailed Design and Algorithms

4.1 Terminal layer

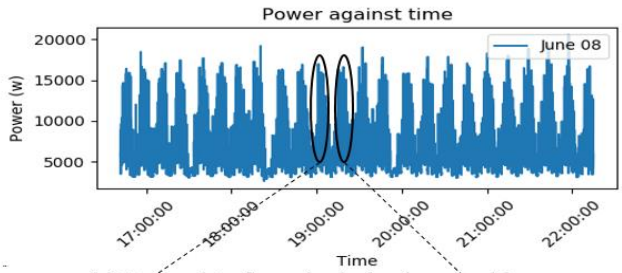
The terminal layer is closely connected to machining equipment and sensor devices. Power sensors are installed in the machines to continuously monitor power. Collected power data are converted from Analog to Digital by an Analog to Digital converter (ADC). The converted raw data are transmitted to the fog layer through the MQTT protocol. When an abnormal situation identified on the fog layer, a new optimized schedule calculated on the cloud layer will be sent to the terminal layer for production adjustment accordingly.

4.2 Fog layer

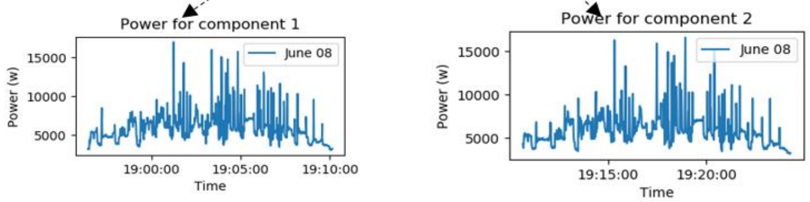
Recently, CNNs have been widely proved for high accuracy in pattern classification such as images and EEG/ECG signals [33], [34]. Considering their distinguishing characteristics, a CNN is therefore designed here for prognosis on machining processes. Furthermore, the accuracy and efficiency of the CNN will be limited if there are a lot of noises and the dataset is a bit long. In this research, pre-processing mechanisms for power signals are developed to support the CNN to process in a more effective means.

In order to better support the CNN to extract the most useful information from raw data during real-time production, monitored power signals are partitioned into individual cutting processes based on power ranges. Standby powers during cutter changes and machining breaks are lower than machining power so that power signals can be partitioned into individual cutting processes to facilitate the following CNN computing. The patterns of partitioned signals are then de-noised and smoothened based on a Gaussian kernel method to preserve the most relevant patterns.

To process monitored power signals efficiently, the power signals are partitioned into a series of separate portions according to each cutting process. A monitored power signal consists of several stages, e.g., idle, machine start-up/shut-down, machining, cutter changes, etc. The partition process is based on the power range to concentrate on the data of machining process. The partition is made based on a set of given thresholds defining the working ranges for machining. The thresholds were decided via experiments for individual machines [35]. The partition process is illustrated in Fig. 2.



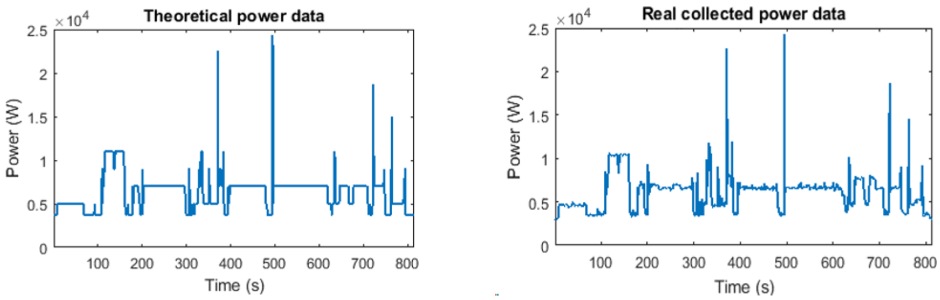
(a) Power data for a single day in a machine



(b) Power patterns for individual cutting processes partitioned from the daily data

Fig. 2. Example of partitioning power data to individual processes.

The partitioned power signals will be further smoothened to de-noise the signals in order to facilitate the CNN for detecting abnormal conditions of machines and tooling effectively. An exemplar theoretical representation of power data is shown in Fig. 3(a). In practical situations, features of power data collected during real production lines are hidden in fluctuated signals with noises (illustrated in Fig. 3(b)). The relevant patterns need to be extracted using suitable statistical methods.



(a) A theoretical power data (b) Power data during real production

Fig. 3. Theoretical and real power data during machining.

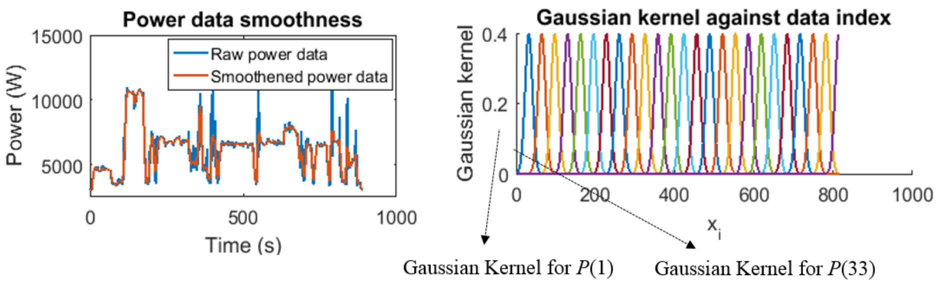
For the research of pattern extraction from fluctuated signals, some statistical methods have been developed, including Principal Component Analysis (PCA) [36], [37], [38], partial least squares [39], cluster analysis [40], [41], etc. According to the research of Navi et al. [43], PCA is the most widely used approach among the above methods to handle signal data with high-dimensional (from multiple-sensor sources), noisy and highly correlated features. However, PCA is not a suitable method to be applicable in this case, in which patterns are extracted from one-dimensional signal data. Furthermore, PCA is a linear method that is difficult to handle time varying dynamic and non-linear systems [42]. To handle low-dimensional data in a robust means, a Gaussian kernel model has been used and its robustness has been proved [43]. Based on the analysis, in this research, the Gaussian kernel model is chosen as a pre-processing mechanism for signal de-noising.

To support pattern extraction, data signals are smoothened by convolution with the Gaussian kernel as follows:

$$P_\sigma(i) = P * g_\sigma(x_i) \tag{1}$$

Where $*$ is the convolutional operator; P is raw power signal; $\boldsymbol{x_i}$ is the ith collected moment of P along the x-axis (Time); $g_\sigma(x_i)$ is the Gaussian kernel for the ith point with a kernel width σ. $P_\sigma(i)$ is the ith smoothened signal point calculated through this equation.

An example of power and some corresponding Gaussian kernels are shown in Fig. 4.



(a) Example of data smoothness (b) Example of the Gaussian kernel P(n)

Fig. 4. Example of signal smoothness with Gaussian kernels.

The Gaussian kernel with a kernel width 'σ' can be calculated below [38]:

$$g_\sigma(x_i) = \exp(-\frac{(x-x_i)^2}{2\sigma^2}) \tag{2}$$

where $(x - x_i)^2$ is the squared Euclidean distance between the i^{th} data point and all the other data points along the x-axis.

It is significant to select proper parameter σ to extract the most useful features for power signals. According to the research of Roueff and Vehel [44], σ can be decided below:

$$\sigma = \frac{I-1}{2\sqrt{\alpha \ln(10)}} \tag{3}$$

where I is the time support of the Gaussian kernel, which can decide the relative amplitude of the entire Gaussians. α is the attenuation coefficient and usually $\alpha = 2$ [44]. σ will be investigated in Section 5.

To detect abnormal situations during machining process, Artificial Neural Network (ANN) has been used for fault identification and type classification. An ANN based anomaly detection algorithm was developed to detect abnormal situations by the authors [31], [35]. However, according to the previous research of the authors, the trained ANN cannot always guarantee high accuracy. Therefore, in this research, a CNN will be considered as a good alternative to detect abnormal situations, and the CNN training is processed on the cloud layer to leverage high computing powers. When a new type of component for machining is added into production, the power signal during normal machining process for the component will be collected as reference signals (usually at beginning of production, machines and cutting tools are in good conditions, and related signals are considered as reference signals for machining the components. Faulty conditions are obtained through machining life-cycles and accumulated historical data). Fault features are generated based on the reference signals according to pre-defined rules [32], [45], [46]. The CNN will be re-trained for update when there are new reference signals for new component machining. The updated CNN will be transmitted to the fog layer for efficient detection. Some abnormal situations supported by the system are described as follows and shown in Fig. 5:

- Severe tool wear: the level of power shifts vertically significantly, but the level of power during the idle stage remains the same [45].
- Tool breakdown: power has a sudden peak and goes back to the air cutting level [45].
- Spindle failures: power has a sudden peak and an increased power level during both machining and idle stages [46].

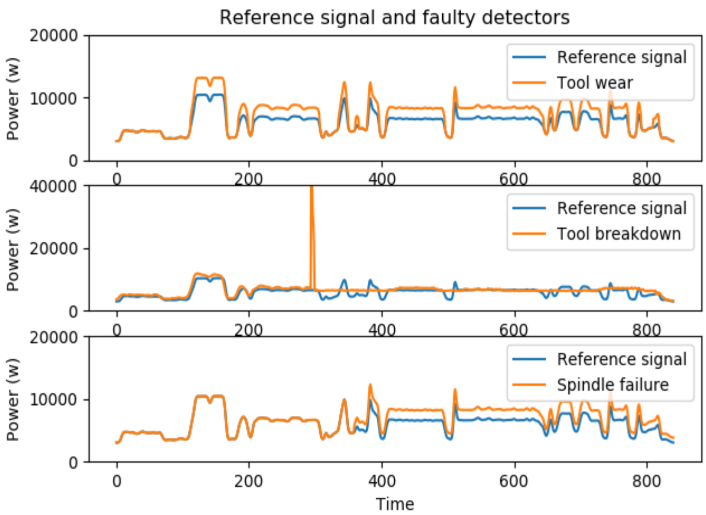


Fig. 5. Examples of reference signals and fault features.

A feature extraction method has been applied to reference signals, so that the faulty features can be amplified to achieve the higher accuracy for the CNN training. The slope and offset between every two consecutive sample points can indicate the characteristics of the curve [47]. The slope is calculated below:

$$slope_i = \lim_{w=R} \frac{P_\sigma(i+w)-P_\sigma(i)}{w} \tag{4}$$

where $P_\sigma(i+w) - P_\sigma(i)$ is the length between two sample point under the width of w; R is the sampling rate of sensor.

$offset_i$ is the y-intercept value that each line going through two sample points. The detection index ID_i is calculated based on $slope$ and $offset$ to amplify fault features as follows:

$$ID_i = \left|(slope_i - \overline{slope})(offset_i - \overline{offset})\right| \tag{5}$$

where $\overline{slope}$ and $\overline{offset}$ are the average value of all $slope$ and $offset$, respectively. The fault detection indexes are then used as input to train the CNN. The advantage of introducing the fault detection index is benchmarked in Section 5.

The CNN is trained on the cloud layer based on amplified reference signals in database. The output decides whether the component is on a normal or faulty condition. A typical CNN has two combined types of layers: convolutional layers and max pooling layers. In convolutional layers, neurons are connected as rectangular grids through a kernel filter with the same weights. In the max pooling layer, rectangular grids are subsampled to extract the core information. At the last stage, all neurons are fully connected. According to research of Rajpurkar et al. [48], the first convolutional and max pooling layer can extract the basic information of data and the following convolutional and max pooling layers can extract the information further. In this research, through experiments, the CNN with one convolutional and max pooling layer has shown good robustness and fast training speed (benchmark results are shown in case studies in Section V). The dimension of kernel filter is 2×2. The basic structure of the designed CNN for this research is shown in Fig. 6. The input/output of the CNN is shown in Table 1.

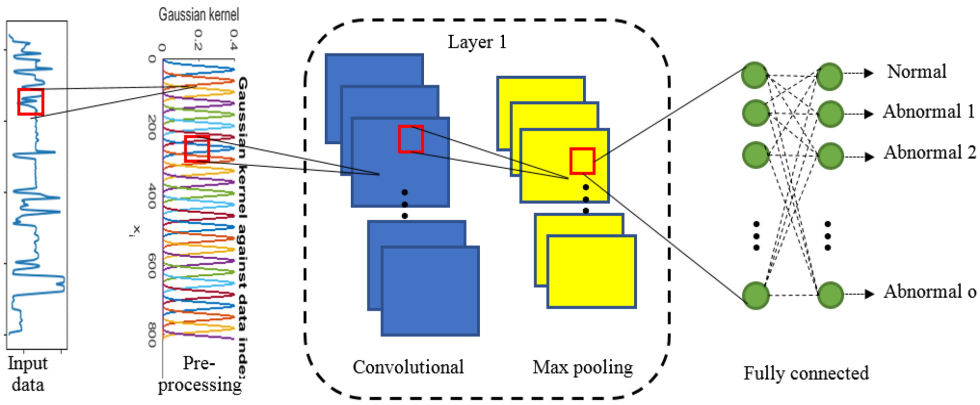


Fig. 6. Basic structure of the designed CNN.

Table 1: Input and output of the CNN.

Input vector (X)	**Output vector ($output$)**
Index 1 in power signal	Normal condition [1, 0, 0, …, 0]
Index 2 in power signal	Abnormal situation 1 [0, 1, 0, …, 0]
… …	… …
Index n in power signal	Abnormal situation o [0, 0, 0, …, 1]

In order to use the CNN, the data are usually reshaped into 2-dimensional data, which is widely used in 1-dimensional signals [48]. Fig. 7 explains an example of such data reshape.

$$[1,2,3,4,5,6,7,8,9] \xrightarrow{\text{Reshape}} \begin{bmatrix} [1, & 2, & 3] \\ [4, & 5, & 6] \\ [7, & 8, & 9] \end{bmatrix}$$

Fig. 7. An example of data reshape.

In a traditional convolutional layer, data is extracted from input data through the kernel filter as follows [43]:

$$C_{cn} = W_{cn} * X_j + b_{cn} \tag{6}$$

where C_{cn} is the output matrix after convolutional layer; W_{cn} is the weight matrix of kernel filter; X_j are the input values matrix; b_{cn} is the bias value.

Max pooling follows the convolutional layer having a lower-resolution input matrix below:

$$P_{mp} = \max_{C_{cn} \epsilon S}(C_{cn}) \tag{7}$$

where P_{mp} are the outputs after the max pooling layer in the max pooling blocks; S is the max pooling block size; All the values of P_{mp} are assembled into Matrix X_{j+1} after the j^{th} convolutional and max pooling layer, so that the size of output matrix is reduced. If there is a following convolutional and max pooling layer, X_{j+1} will be processed in the $(j+1)^{th}$ convolutional layer according to Equation (6) until reaching the fully connected layer.

After convolutional and max pooling layers, the fully-connected layer is used in the final step for classification as calculated in the following Equation (8).

$$output = W_{full} \times X_{final} + B_{full} \tag{8}$$

where $output$ is the classification result; X_{final} is the matrix after the final max pooling layer; W_{full} and B_{full} are the weight matrix and bias in fully connected layer, respectively.

The above CNN design is suitable for a linear model, so that the accuracy of model can be limited when dealing with non-linear problems. To improve the training efficiency, further improvements are made for the above CNN design based on recent research on CNN. During the training process, Batch Normalization (BN) [49], ReLU [50] and Softmax [51] are designed to achieve an optimal detection rate of abnormal situations. BN and ReLu are applied between each layer. Softmax is applied at the end of the fully connected layer to avoid extreme data. Dropout is also significant to avoid overfitting during training process for the CNN. However, it is not necessary for a lightweight CNN (e.g., the CNN in this research has a convolutional layer and a max pooling layer so that it is a lightweight CNN). Based on the above research, the CNN in this research is enhanced using BN, ReLU and Softmax (shown in Fig. 8). The advantage of the designed CNN in this research is validated in the case studies in Section 5.

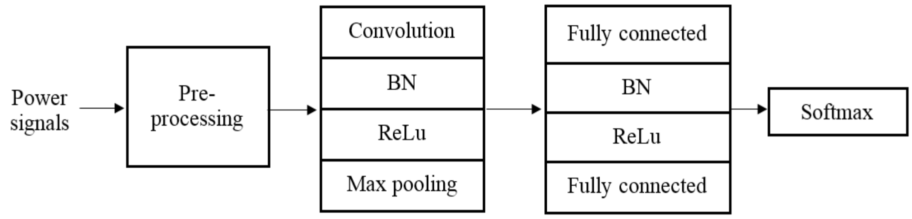


Fig. 8. BN, ReLU and Softmax introduced to enhance the CNN in this research.

When abnormal situations are identified, the machine will be temporarily disabled. The disabling time $T_{disable}$ is based on the problem complexity decided by engineers. Managed by the system coordinator, $T_{disable}$ will be sent to the terminal layer when the production schedule is re-scheduled on the cloud layer. Fog nodes usually have limited computing power and storage. Therefore, only an updated CNN is deployed on the fog layer to identify the abnormal situations without storing reference signals. Reference signals are stored in databases on the cloud layer. If new components need to be machined in the production lines, the system coordinator will be triggered and the power data signals will be collected as reference signals. As thus, the CNN will be re-trained with new reference signals on the cloud layer. The advantages of bandwidth saving and low latency under this design will be demonstrated in Section V. Meanwhile, the re-scheduling optimization algorithm requiring intensive computing will be deployed on the cloud layer and triggered by the system coordinator when necessary.

4.3 Cloud layer

Cloud centers have the advantage of high performance of computing powers and storage spaces. Reference signals are also stored in databases on the cloud. The scheduling optimization algorithm and CNN training generally require iterations/epochs to calculate optimum results. Therefore, they are processed on the cloud layer. At the beginning of each machining cycle, an optimization algorithm is applied to provide the optimum scheduling to minimize energy consumption, makespan and machine utilization rate by using the Fruit Fly Optimization (FFO) algorithm [35]. The optimized schedule is transmitted to the terminal layer for production control. When the cloud layer receives new reference signals, the database is updated and the CNN is re-trained with the updated reference signals, so that the CNN is able to detect and identify newly abnormal situations in the machining cycle. The updated CNN is transmitted to the fog layer for abnormal situation detection.

When any abnormal situation is identified, the machine is disabled for maintenance. The disabling time $T_{disable}(M_k)$ of Machine M_k is based on the type of abnormal situations. A multi-objective optimization mode is established for re-scheduling [35]. The total energy consumption of Machine M_k in the production line is calculated:

$$E_{total}(M_k) = E_{machine}(M_k) + E_{wait}(M_k) + E_{disable}(M_k) \tag{9}$$

where $E_{total}(M_k)$ is the total energy consumption during all the machining phases of machine M_k; $E_{machine}(M_k)$, $E_{wait}(M_k)$ and $E_{disable}(M_k)$ represent the energy consumption during machining, waiting and disabling phases, respectively.

The total time during all phases of all machines: $Makespan$, which is the maximum time used by all the machines in the production lines, can be calculated below:

$$Makespan = \underset{k=1}{\overset{n}{Max}} \left(T_{total}(M_k)\right) \tag{10}$$

The balanced utilization of production is calculated below:

$$\mu = \frac{\sum_{k=1}^{n} T_{total}(M_k)}{n} \tag{11}$$

$$Utlization\_level = \sqrt{\sum_{k=1}^{n} (T_{total}(M_k) - \mu)^2} \tag{12}$$

To optimize E_{total}, $Makespan$ and $Utlization\_level$ at the same time, Nadir and Utopia points are employed to normalize the objectives. The fitness function is calculated weighted sum of the three objectives as follows:

$$fitness = min(w_1 \cdot NE + w_2 \cdot NT + w_3 \cdot NU) \tag{13}$$

where NE, NT and NU are the normalized value of E_{total}, $Makespan$ and $Utlisation\_level$, respectively; w_1, w_2 and w_3 (weight sum = 1) are the weights for the three objectives.

The fitness is optimized by using the improved FFO algorithm to calculate a scheduler or a re-schedule from previous research [35]. The details for the optimization algorithm can be found out in [35].

4.4 Time complexity analysis

For the system, computation tasks are conducted on the fog layer and cloud layer, so that the time complexity for the system is analyzed for the two layers.

Fog layer

On the fog layer, the Gaussian kernel and detection index are calculated through matrix multiplication. The time complexity can be calculated below:

$$TX_1 = O(n^3) \tag{14}$$

where TX_1 is the time complexity of the Gaussian kernel and detection index; n is the input size of the CNN.

According to Ke et al. [52], the computation process of the convolutional layer would be the most complex one in the CNN. The time complexity of the convolutional layer on the fog layer is calculated below [53]:

$$TX_2 = O(\sum_{l=1}^{d} n_{l-1} \times {s_l}^2 \times n_l \times {m_l}^2) \tag{15}$$

where TX_2 is the time complexity of the convolutional layer on the fog layer; l is no. of a convolutional layer in the CNN; d is the total number of convolutional layers; n_l is the number of filters in the lth layer; n_{l-1} is also known as the number of input channels of the lth layer; s_l is the length of the filter; m_l is the spatial size of the output feature map.

<u>Cloud layer</u>

On the cloud layer, both the optimization algorithm and CNN training are performed. The time complexity of optimization can be calculated below [54]:

$$TX_3 = \mathrm{O}(n_g \times n_0 \times {n_p}^3) \tag{16}$$

where TX_3 is the time complexity of optimization; n_g is the number of generations in the algorithm; n_0 is the number of optimization objectives; n_p is the population size.

According to the research of Bianchini and Scarselli [55], the maximum time complex for training a CNN is:

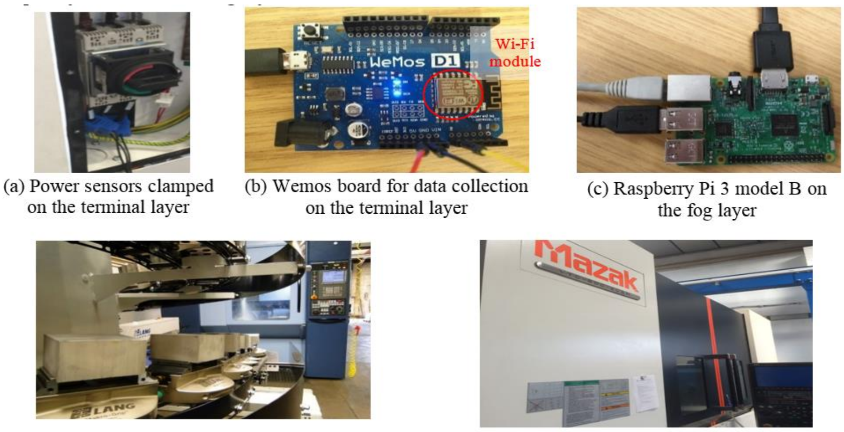
$$TX_4 = O(4^{h^2}(nh)^{n+2h}) \tag{17}$$

where TX_4 is the time complexity of the CNN; h is the total number of hidden neurons in all the layers; n is the number of the input signals.

5. System Deployment and Analysis

5.1 System setup and deployment

For the system, on the terminal layer, power sensors are mounted with machining equipment. Power data are collected by the Wemos Wi-Fi board (https://www.wemos.cc/), which is a cost-effective platform with an Analog to Digital converter and an embedded Wi-Fi module. Monitored power data are transmitted to the fog layer via Wi-Fi. The fog layer consists of an Internet router and Raspberry Pi (https://www.raspberrypi.org/), Raspberry Pi is located right beside the Wemos Wi-Fi board to receive data within the shop floor. Received data on the fog layer are processed by the trained CNN deployed into the Raspberry Pi. Reference signals are stored on the cloud layer to train the CNN. Managed by the system coordinator, the CNN is re-trained when necessary, and the trained CNN is transmitted back to the fog layer for re-deployment. Fig. 9 illustrates the system and some machining centers for this deployment. Table 2 shows the specification of the Raspberry Pi used on the fog layer and the cloud server.



(a) Power sensors clamped on the terminal layer

(b) Wemos board for data collection on the terminal layer

(c) Raspberry Pi 3 model B on the fog layer

(d) Loading/unloading tower for raw materials and the MX520 CNC machine

(e) The Mazak machine

Fig. 9. Deployment of the fog computing system into machining processes.

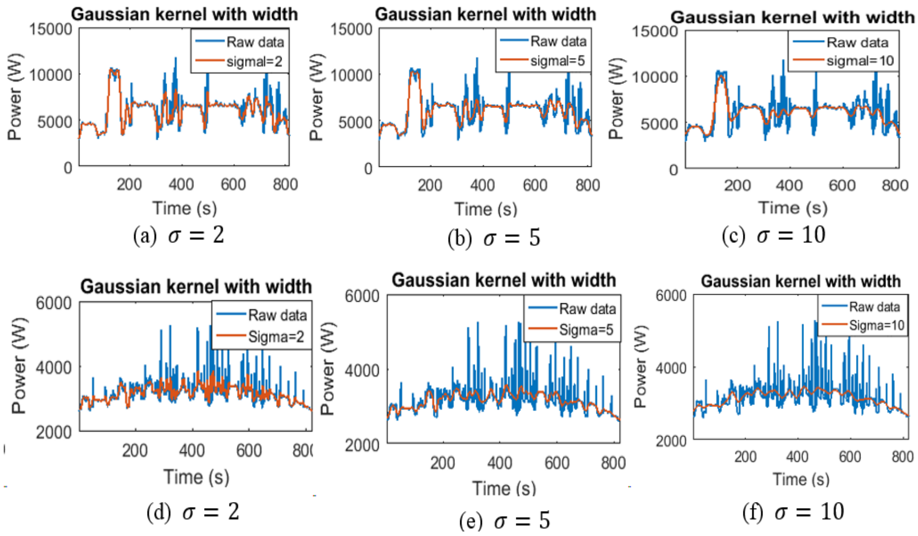
Table 2: Specification of the Raspberry Pi and cloud server.

Hardware	**Connectivity**	**Processor**	**Memory**	**Storage**
Raspberry Pi	2.4GHz/5GHz IEEE 802.11	1.4GHz	1GB	SD card (16GB)
Cloud server	2.4GHz/5GHz IEEE 802.11	4.2Ghz	32GB	1TB

The fog enabled prognosis system was deployed in a UK company for this industrial trial. The company specializes in precision machining for automotive, aeronautical and nuclear components. Two CNC machine centers (MX520, MAZAK VTC-800/20SR) were monitored for approximately three months. Raw materials towers and loading/unloading robotic arms were integrated into the production line to implement automation processes. The production is automatic, working for 6 days per week (from Monday to Saturday). Operators/maintenance engineers were scheduled to check the status of machining processes.

5.2 Analysis on Gaussian kernel modelling and fault index

In the system, the Gaussian kernel is applied to monitored raw power signals according to Equations (1)-(3). Based on this pre-processing mechanism, the identification process of the most useful features from machining processes by the CNN will be much quicker and more accurate. Via experiments on a number of samples and statistical analysis, when $\sigma = 10$, the smoothened signal can keep the most of the machining features. Fig. 10 illustrates the smoothness with different values of σ.

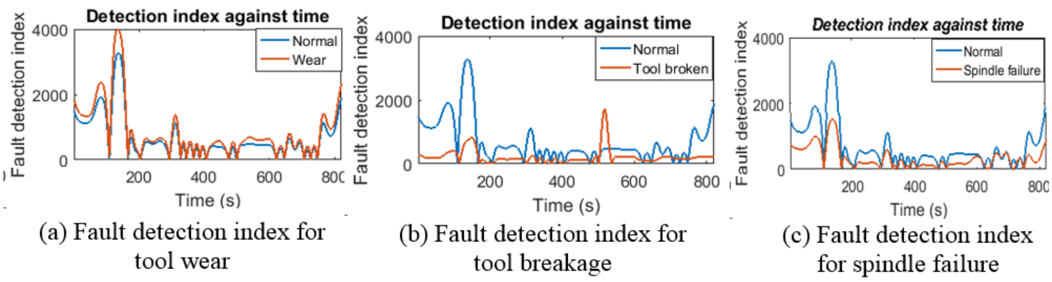


(a) $\sigma = 2$ (b) $\sigma = 5$ (c) $\sigma = 10$

(d) $\sigma = 2$ (e) $\sigma = 5$ (f) $\sigma = 10$

Fig. 10. Smoothened signals by the Gaussian kernel with different width.

As shown in Fig. 11, an example of fault detection indexes is created according to Equations (4)-(5). As shown in Table 3, the Mean Absolute Percentage of Error (MAPE) between a faulty signal and a reference signal is calculated for both smoothened data and fault detection index. It can clearly indicate that the faulty features can be significantly magnified to facilitate feature identification by the CNN process.



(a) Fault detection index for tool wear (b) Fault detection index for tool breakage (c) Fault detection index for spindle failure

Fig. 11: Fault detection index.

Table 3: MAPEs for smoothened data and fault detection index.

	Tool wear	**Tool breakage**	**Spindle failure**
Smoothened data	83.66	86.75	23.68
Fault detection index	260.23	699.99	522.28
Improvement in magnification	211.06%	706.90%	2105.57%

In order to show the effectiveness of introducing the Gaussian kernel and fault detection index to improve the training process of the CNN, both signal data with/without smoothness and index are utilized for comparison. The benchmark results are shown in Fig. 12. The training process of the CNN with smoothened and indexed data achieved 98% accuracy in 12 epochs. The training process only with indexed data achieved 98% accuracy in 21 epochs. Other training processes were not able to achieve 98% accuracy in 100 epochs. As thus, the pre-processing mechanisms based on smoothened and indexed data can accelerate the training speed of the CNN.

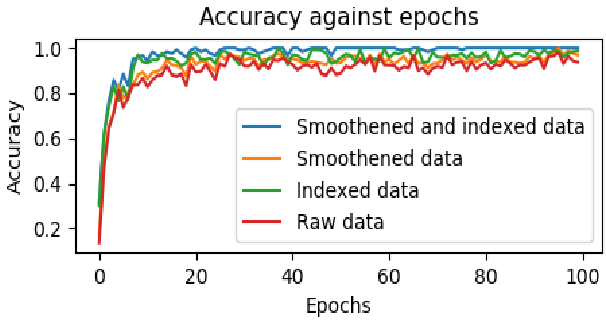
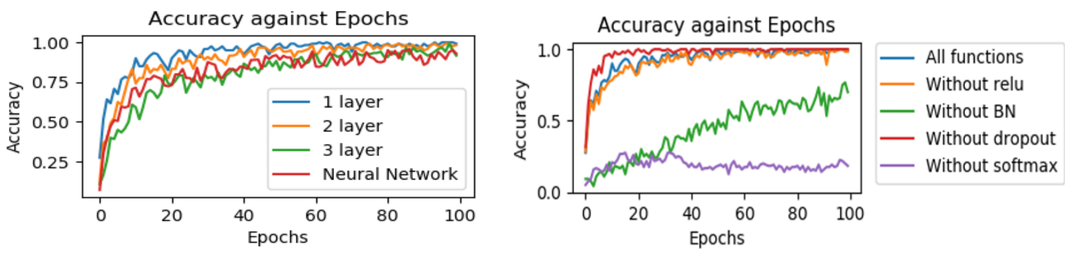


Fig. 12. Benchmarks on the CNN training with processed and raw signal data.

5.3 Analysis on different structures of the CNN design

In order to validate the structure design of the CNN in this research, benchmark analyses were carried out by executing training different structures of the CNN and Artificial Neural Network (ANN) for 5 times each. The training processes on ANN and different structures of the CNN with different convolutional and max pooling layers were compared and illustrated in Fig. 13(a). The training processes of the CNN with/without the functions of BN, Relu, Dropout and Softmax were compared and illustrated in Fig. 13(b). In this case study, the accuracy of ANN (92.91%) is much lower than that of CNN (98%). Therefore, ANN cannot satisfy the requirement of real production with low fault-tolerance. It can be observed clearly that the CNN, which is designed with one convolutional and max pooling layer with BN, Relu and Softmax functions but without dropout, can achieve 98% accuracy in 12 epochs (the comparisons are illustrated in Tables 4 and 5).



(a) Benchmark of Neural Network and CNN with different convolutional and max pooling layers

(b) Benchmark of CNN with different functions

Fig. 13. Comparison on training the CNN with different design.

Table 4: Benchmark on Training of the ANN and CNN with different layers.

	One layer	Two layers	Three layers	ANN
Epochs to achieve 98%	12	60	98	Did not achieve

Table 5: Benchmark on the CNN with various functions.

	with all functions	without Relu	without BN	Without dropout	without Softmax
Epochs to achieve 98%	44	50	Did not achieve	12	Did not achieve

5.4 Analysis on system performance improvement

The system separate the computation of the training and execution processes of the CNN into the fog and cloud layers respectively. This "divide and conquer" approach developed in this paper can significantly optimize the data traffic and computation efficiency in comparison with cloud solutions.

During this industrial trial, the total quantity of collected power data were 12.98GB. Owing to the fog design, only 3.36GB data were transmitted to the cloud layer. As thus, 70.26% bandwidth was saved in comparison with a cloud solution [35]. For 1Mb data, it roughly took 1.31s for monitoring signals to be transmitted to and processed on fog nodes, while it roughly took 3.96s for data transmitted and processed in a cloud center (it was also observed that data transmission to the cloud solution was even much slower when there are transmission congestions in the Internet). The improvements of the time spent for data transfer in the fog system was 47.02%. The summary of the system performance is shown in Table 6.

Table 6: Comparison of the fog system and a cloud system [35].

Method/Improvement	Total monitored data (GB)	Transmitted data to cloud (GB)
This fog based system	12.98	3.86
Cloud based system [35]	12.98	12.98
Bandwidth saving	70.26%	
Reduction of time spent for data transfer	47.02%	

The fog design presented in this research was further justified through the following computing and storage. Based on the configuration of Table 2, calculation time and memory usage for all the algorithmic elements in the industrial trials are shown in Table 7. It indicates that pre-processing and CNN for prognosis can be effectively implemented on the fog layer based on the capacity of the edge devices. However, the scheduling optimization and CNN training demonstrated high time complexity and should be arranged on the cloud. Thus, this "divide and conquer" fog architecture can ensure close interaction efficiency between prognosis and working machines, and leverage the capacity of cloud servers for intensive computation.

Table 7: Real calculation time and memory usage for complexity analysis.

Fog layer	**Gaussian kernel**	**Detection index**	**CNN for prognosis**	**Total**
Computing time (s)	0.21	0.12	0.98	1.31
Memory usage (MB)	1.37	1.10	3.76	6.32
Cloud layer	**Optimization**	**CNN training**		**Total**
Computing time (s)	3.18	1253.91		1257.09
Memory usage (MB)	20.12	291.93		312.05

5.5 Analysis on improved efficiency of machining system

For the company, before the deployment of the fog system, the production experienced various breakdown due to the issues of tooling, machines and long-time standby for manual check on machining conditions (examples before the deployment of the system are illustrated in Fig. 14). After the deployment of the fog enabled prognosis system, the efficiency of energy and production was improved approximately for 29.25% and 16.50% on average.

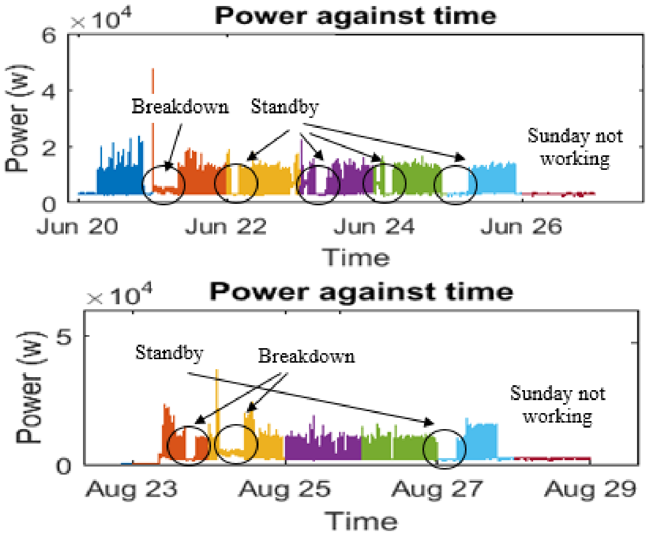


Fig. 14. Production with a number of breakdown in two weeks as examples.

6. Conclusions

This paper presents a fog and CNN enabled system for dynamic prognosis and optimization of machining processes to address practical manufacturing situations. The functions and innovations include:

(1) A three-layer fog and cloud model is developed to optimize computing resource utilization and minimize data traffic. Enabled by the fog computing model, the system can overpass the prevalent low-efficiency issue of cloud based systems rooted from the high latency of data transfer in industrial networks. Complex analysis and computation in the system, like algorithm training, data analysis,

scheduling/re-scheduling optimization, are well balanced with the design.

(2) Modern machining companies especially SMEs are characterized by low-volume and high-mix customized production. The occurrence of faults during customized machining processes is uncertain. To address complex conditions effectively, the system is augmented with a CNN to dynamically mine sensitive fault features from large-scale signals acquired in practical production. Pre-processing mechanisms are designed for effective signal de-noising to enhance the performance of the CNN in practical manufacturing situations. Benchmark experiments and analyses are conducted to determine suitable structures and parameters for the CNN design.

(3) The performance of the system was assessed in a European machining company and achieved satisfactory results. With significant improvements on bandwidth reduction and system efficiency in case studies, the effectiveness and applicability of this system to manufacturing practices were clearly indicated. The system is applicable to other manufacturing processes by training the CNN using fault features from those processes. The research provides manufacturing practitioners an innovative solution based on the state-of-the-art industrial artificial intelligence.

Currently, the training efficiency is still limited in re-training the CNN when new component patterns are introduced. The transfer learning technology can resolve the issue by training the CNN based on the parameters of the trained CNN. This topic will be researched in the future work.

Acknowledgement:

This research was carried out through the sponsorship of the EU Smarter and Cloudflow projects (PEOPLE-2013-IAPP-610675 and FP7-2013-NMP-ICT-FOF-609100). The research was also partially sponsored by the SABOT project (the Innovate UK funder). The authors acknowledge the funders and industrial partners to support the research.

References:

[1] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao and D. Siegel, "Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications", *Mechanical Systems and Signal Processing*, vol. 42, no. 1-2, pp. 314-334, 2014.

[2] R. Gao, L.H. Wang, R. Teti, D. Dornfeld, S. Kumara, M. Mori and M, Helu, "Cloud-enabled prognosis for manufacturing", *CIRP Annals*, vol. 64, no. 2, pp. 749-772, 2015.

[3] F. Tao, Q. Qi, A. Liu and A. Kusiak, "Data-driven smart manufacturing", *Journal of Manufacturing Systems*, vol. 48, pp. 157-169, 2018.

[4] E. Baccarelli, P. Naranjo, M. Scarpiniti, M. Shojafar and J. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study", *IEEE Access*, vol. 5, pp. 9882-9910, 2017.

[5] P. Hu, S. Dhelim, H. Ning and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues", *Journal of Network and Computer Applications*, vol. 98, pp. 27-42, 2017.

[6] M. Mukherjee, L. Shu and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges", *IEEE Communications Surveys & Tutorials*, pp. 1-1, 2018.

[7] D. Wu, S. Liu, L. Zhang, J. Terpenny, R. Gao, T. Kurfess and J. Guzzo, "A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing", *Journal of Manufacturing Systems*, vol. 43, pp. 25-34, 2017.

[8] C. Lin and J. Yang, "Cost-efficient deployment of fog computing systems at logistics centers in Industry 4.0", *IEEE Transactions on Industrial Informatics*, April 2018 (accepted and on-line early access).

[9] J. Wan, B. Chen, S. Wang, M. Xia, D. Li and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory", *IEEE Transactions on Industrial Informatics*, 2018 (accepted and on-line early access).

[10] P. O'Donovan, C. Gallagher, K. Bruton and D. O'Sullivan, "A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications", *Manufacturing Letters*, vol. 15, pp. 139-142, 2018.

[11] N. Mohamed, J. Al-Jaroodi and I. Jawhar, "Utilizing fog computing for multi-robot systems", *2018 Second IEEE International Conference on Robotic Computing (IRC)*, CA, USA, January 31- February 2, 2018.

[12] R. Liu, B. Yang, E. Zio and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review", *Mechanical Systems and Signal Processing*, vol. 108, pp. 33-47, 2018.

[13] J. Wang, Y. Ma, L. Zhang, R.X. Gao and D. Wu, "Deep learning for smart manufacturing: Methods and applications", *Journal of Manufacturing Systems*, January 2018 (accepted and on-line early access).

[14] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang and R.X. Gao, "Deep learning and its applications to machine health monitoring", *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2018.

[15] J. Tian, C. Morillo, M.H. Azarian, M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793-1803, 2016.

[16] Z. Zhou, C. Wen, C. Yang, "Fault isolation based on k-nearest neighbor rule for industrial processes", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2578-2586, 2016.

[17] F. Li, J. Wang, B. Tang, D. Tian, "Life grade recognition method based on supervised uncorrelated orthogonal locality preserving projection and k-nearest neighbor classifier", *Neurocomputing*, vol. 138, pp. 271-282, 2014.

[18] C, Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, R.E., Vasquez, "Multimoal deep support vector classification with homologous features and its application to gearbox fault diagnosis", *Neurocomputing*, vol. 168, pp. 119-127, 2015.

[19] J. Zhou, Y. Yang, S. Ding, Y. Zi and M. Wei, "A fault detection and health monitoring scheme for ship propulsion systems using SVM technique", *IEEE Access*, vol. 6, pp. 16207-16215, 2018.

[20] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost", *IEEE Access*, vol. 6, pp. 21020-21031, 2018.

[21] A. Abdullah, "Ultrafast transmission line fault detection using a DWT-based ANN", *IEEE Transactions on Industry Applications*, vol. 54, no. 2, pp. 1182-1193, 2018.

[22] B. Luo, H. Wang, H. Liu, B. Li and F. Peng, "Early fault detection of machine tools based on deep learning and dynamic identification", *IEEE Transactions on Industrial Electronics*, January 2018 (accepted and on-line early access).

[23] T. Ince, S. Kiranyaz, L. Eren, M. Askar and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067-7075, 2016.

[24] M. Xia, T. Li, L. Xu, L. Liu and C.W. de Silva, "Fault diagnosis for rotating machinery using multiple sensor and convolutional neural networks", *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101-110, 2018.

[27] Z. Liu, Y. Guo, M. Sealy and Z. Liu, "Energy consumption and process sustainability of hard milling with tool wear progression", *Journal of Materials Processing Technology*, vol. 229, pp. 305-312, 2016.

[28] M. Sealy, Z. Liu, D. Zhang, Y. Guo and Z. Liu, "Energy consumption and modeling in precision hard milling", *Journal of Cleaner Production*, vol. 135, pp. 1591-1601, 2016.

[29] K. Chooruang and P. Mangkalakeeree, "Wireless heart rate monitoring system using MQTT", *Procedia Computer Science*, vol. 86, pp. 160-163, 2016.

[30] A. Schmitt, F. Carlier and V. Renault, "Dynamic bridge generation for IoT data exchange via the MQTT protocol", *Procedia Computer Science*, vol. 130, pp. 90-97, 2018.

[31] Y.C. Liang, W.D. Li, S. Wang and X. Lu, "Big data enabled dynamic anomaly detection for machining condition monitoring", *Engineering*, 2018 (submitted).

[32] S. Wang, Y.C. Liang, W.D. Li and X. Cai, "Big data enabled intelligent immune system for energy efficient manufacturing management", *Journal of Cleaner Production*, vol. 195, pp. 507-520, 2018.

[33] V. Franc and J. Čech, "Learning CNNs from weakly annotated facial images", *Image and Vision Computing*, vol. 77, pp. 10-20, 2018.

[34] S. Banerjee and S. Das, "Mutual variation of information on transfer-CNN for face recognition with degraded probe samples", *Neurocomputing*, vol. 310, pp. 299-315, 2018.

[35] Y.C. Liang, X. Lu, W.D. Li and S. Wang, "Cyber Physical System and Big Data enabled energy efficient machining optimisation", *Journal of Cleaner Production*, vol. 187, pp. 46-62, 2018.

[36] Q. Liu, S. Qin and T. Chai, "Decentralized fault diagnosis of continuous annealing processes based on multilevel PCA", *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 687-698, 2013.

[37] Y. Guo, G. Li, H. Chen, Y. Hu, H. Li, L. Xing and W. Hu, "An enhanced PCA method with Savitzky-Golay method for VRF system sensor fault detection and diagnosis", *Energy and Buildings*, vol. 142, pp. 167-178, 2017.

[38] A. Kyprianou, A. Phinikarides, G. Makrides and G. Georghiou, "Definition and computation of the degradation rates of photovoltaic systems of different technologies with robust Principal Component Analysis", *IEEE Journal of Photovoltaics*, vol. 5, no. 6, pp. 1698-1705, 2015.

[39] J. Wu, W. Chen, K. Huang and T. Tan, "Partial least squares based subwindow search for pedestrian detection", *2011 18th IEEE International Conference on Image Processing*, 2011.

[40] Z. Yu, H. Chen, J. You, J. Liu, H. Wong, G. Han and L. Li, "Adaptive fuzzy consensus clustering framework for clustering analysis of cancer data", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 887-901, 2015.

[41] R. Yan, Z. Ma, G. Kokogiannakis and Y. Zhao, "A sensor fault detection strategy for air handling units using cluster analysis", *Automation in Construction*, vol. 70, pp. 77-88, 2016.

[42] M. Navi, N. Meskin and M. Davoodi, "Sensor fault detection and isolation of an industrial gas turbine using partial adaptive KPCA", *Journal of Process Control*, vol. 64, pp. 37-48, 2018.

[43] X. Rimpault, E. Bitar-Nehme, M. Balazinski and J. Mayer, "Online monitoring and failure detection of capacitive displacement sensor in a Capball device using fractal analysis", *Measurement*, vol. 118, pp. 23-28, 2018.

[44] F. Roueff and J. Vehel, "A regularization approach to fractional dimension estimation", *Fractals and Beyond*, World Scientific Publisher, 2018.

[45] R. Teti, K. Jemielniak, G. O'Donnell and D. Dornfeld, "Advanced monitoring of machining operations", *CIRP Annals – Manufacturing Technology*, vol. 59, pp. 717-739, 2010.

[46] D. Axinte and N. Gindy, "Assessment of the effectiveness of a spindle power signal for tool condition monitoring in machining processes", *International Journal of Production Research*, vol. 42, issue 13, pp. 2679-2691, 2004.

[47] Z. Feng, M. Zuo and F. Chu, "Application of regularization dimension to gear damage assessment", *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 1081-1098, 2010.

[48] P. Rajpurkar, A. Hannun, M. Haghpanahi, C. Bourn and A. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks", *arXiv:1707.01836*, 2018.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv:1502.03167*, 2015.

[50] K. Priddy and P. Keller, *Artificial neural networks*. Bellingham, Wash. <1000 20th St. Bellingham WA 98225-6705 USA>: SPIE, 2005.

[51] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", *arXiv:1207.0580v1*, 2012.

[52] X. Ke, W. Cao and F. Lv, "Relationship between Complexity and Precision of Convolutional Neural Networks", *Proceedings of the 2017 2nd International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE 2017)*, 2017.

[53] K. He and J. Sun, "Convolutional Neural Networks at Constrained Time Cost", *arXiv:1412.1710*, 2014.

[54] D. Curry and C. Dagli, "Computational complexity measures for multi-objective optimization problems", *Procedia Computer Science*, vol. 36, pp. 185-191, 2014.

[55] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1553-1565, 2014.