

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/330241216>

Artificial Intelligent Diagnosis and Monitoring in Manufacturing

Preprint · December 2018

CITATIONS

0

READS

363

7 authors, including:



Ye Yuan

University of Cambridge

209 PUBLICATIONS 2,677 CITATIONS

SEE PROFILE



Cheng Cheng

London Business School

16 PUBLICATIONS 30 CITATIONS

SEE PROFILE



Hai-Tao Zhang

Huazhong University of Science and Technology

163 PUBLICATIONS 1,980 CITATIONS

SEE PROFILE



Han Ding

Jilin University

372 PUBLICATIONS 5,484 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Modeling, simulation and control of complex systems; pattern recognition and intelligent machines; new energy vehicles and distributed micro grid [View project](#)



Collective dynamics [View project](#)

Artificial Intelligent Diagnosis and Monitoring in Manufacturing

Ye Yuan^{1,2,*}, Guijun Ma³, Cheng Cheng², Beitong Zhou², Huan Zhao³, Hai-Tao Zhang^{1,2}, Han Ding^{1,3,*}

¹*State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, P.R. China.*

²*School of Automation, Huazhong University of Science and Technology, Wuhan 430074, P.R. China.*

³*School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, P.R. China.*

The manufacturing sector is heavily influenced by artificial intelligence-based technologies with the extraordinary increases in computational power and data volumes. It has been reported that 35% of US manufacturers are currently collecting data from sensors for manufacturing processes enhancement. Nevertheless, many are still struggling to achieve the ‘Industry 4.0’, which aims to achieve nearly 50% reduction in maintenance cost and total machine downtime by proper health management^{1,2}. For increasing productivity and reducing operating costs, a central challenge lies in the detection of faults or wearing parts in machining operations³. Here we propose a data-driven, end-to-end framework for monitoring of manufacturing systems. This framework, derived from deep learning techniques, evaluates fused sensory measurements to detect and even predict faults and wearing conditions. This work exploits the predictive power of deep learning to extract hidden degradation features from noisy data. We demonstrate the proposed framework on several representative experimental manufacturing datasets drawn from a wide variety of applications, ranging from mechanical to electrical systems. Results reveal that the framework performs well in all benchmark applications examined and can be applied in diverse contexts, indicating its potential for use as a critical corner stone in smart manufacturing.

The aim of smart manufacturing is to integrate advanced information techniques into manufacturing processes to produce such benefits as improved production quality and cost reduction^{4,5}. Unexpected manufacturing failures can halt production and lead to wastage of raw materials or system malfunctions. In recent decades, it has been envisioned that manufacturing data, including vibration, pressure, temperature, and energy data, can be used to support artificial intelligence (AI) algorithms⁶. AI algorithms have the potential to detect the locations of faults or even predict them before they occur; doing so could allow regular maintenance to be replaced by condition-based or predictive maintenance, which would be more effective in reducing unnecessary maintenance while also guaranteeing the reliability of the machinery⁷. However, conversion of the measured data from manufacturing processes into actionable knowledge about the health status of the equipment has proven challenging⁸.

In the past, the measured signals have often been processed via feature extraction⁹ to represent the complete signals manually. The extracted features are then used to train the system using standard classification and regression methods to allow predictions to be made in a case-by-case manner¹⁰⁻¹². When the features have been extracted, the next step involves translation of the fault diagnosis problem into classification and regression forms. Common methods used to implement this step include use of neural networks (NNs)¹³, support vector machines (SVMs)¹⁴, and adaptive neuro fuzzy inference systems (ANFISs)¹⁵. Use of an ANN has been a common mode of choice in applications such as medicine, industry, and power systems since 1997¹⁶. However, use of ANNs has featured less in the recent literature because it is hard to escape from a local minimum when using an ANN¹⁷. In contrast, SVMs and ANFISs are now widely used as algorithms for fault diagnosis problems. Widodo *et al.*¹⁸ demonstrated the high accuracy of an SVM in fault diagnosis and machine condition monitoring applications when compared with other intelligent methods, such as ANN and the random forest method. ANFIS combines the advantages of neural networks with those of fuzzy systems, thus offering high computational power and reasoning capabilities simultaneously. However, these methods are difficult to apply to problems that have a variety of operating conditions and require arduous fine-tuning for various scenarios. There is

thus an urgent need for a method that can simultaneously provide convenience for feature extraction and offer universality for use in diverse manufacturing applications.

Deep learning algorithms¹⁹ have recently been shown to exceed human performance in Go²⁰, Atari games²¹, and other fields. When compared with traditional machine learning algorithms, the advantage of deep learning is that it enables automatic feature extraction from raw data and can thus eliminate any dependence on prior knowledge¹⁹. The convolutional neural network (CNN), as an important type of neural network, obtained remarkable results in ImageNet in 2012²² and has gradually become a representative method that is used in medical diagnosis²³, image recognition²⁴ and speech recognition²⁵ applications. Hence, we here propose the use of the CNN framework together with experimental mechanical data to solve various manufacturing problems, including fault diagnosis, and condition monitoring.

The work in this paper is designed to transform these key problems into a unified supervised learning framework. In particular, it proposes a general end-to-end framework, i.e., a CNN that can extract features automatically and solve the problems accurately. The performance is verified using some open competition datasets including Case Western Reserve University's bearing data²⁶ and hydraulic system data²⁷, airplane girder simulation damage data²⁸, broken tool data, and the bearing²⁸, tool wear, and gearbox data²⁹ that were collected via our experiments. All these data were converted into classification problems. National Aeronautics and Space Administration (NASA) tool wearing data³⁰, battery data³¹ and the Centre of Advanced Life Cycle Engineering (CALCE) battery data³² were converted into regression problems. After simple truncation and filter-based pre-processing, we substituted the data into a multilayer CNN model for training and testing.

Rolling bearing fault detection and classification is used here as an illustrative example. Rolling bearings are vital components in many types of rotating machinery, ranging from simple electrical fans to complex machine tools. More than half of machinery defects are generally related to bearing faults³³. Typically, a rolling bearing fault can lead to machine shutdown, chain

damage, and even human casualties³³. Bearing vibration fault signals are usually caused by localized defects in three components: the rolling elements; the outer race; and the inner race. When bearings near the end of their lifetimes, instances of deformation, cracking, and burning among these components may cause spindle deviation and cause further serious damage to the mechanical system.

A bearing data set provided by the Case Western Reserve University (CWRU) data centre²⁶, which is regarded as a benchmark for the bearing fault diagnosis problem, was used to validate the effectiveness of our proposed framework. An experimental platform (illustrated in Figure 1(d)) was used to conduct the signals to be used for defect detection on bearings of three different fault diameters (7 mils, 14 mils, and 21 mils (1 mil=0.001 inches)). Vibration signals in different conditions from the inner race, the outer race, and the rolling elements for all fault diameters were acquired using accelerometers. We solved this fault diagnosis problem by classifying the fault types as representations of the following three problems: a) binary classification (normal plus fault conditions); (b) four-way classification (normal plus three main fault conditions); and (c) ten-way classification (normal plus three main fault conditions for each of the fault diameters).

The dataset originally consisted of four normal samples and 52 faulty samples. Each sample contained a different number of time-course measurements. Because a faulty signal represents a stationary condition, we reshaped the samples here to ensure that each sample had 6000 time-course measurements consistently. In total, 1320 samples were reconstructed from the original dataset. To evaluate the performance of the model, the entire pre-processed dataset was obtained and then randomly divided into 90% training data (1188 samples) and 10% validation data (132 samples); and all portions of the data were ordered arbitrarily. Figure 1(a)–(c) show the classification results. These results show that all three models achieved 100% (i.e., 132 of 132 validation samples) fault classification, also the result is consistent over different randomization.

For further evaluation of the classification results, we used the following three assessment metrics to evaluate the classification performance with validation data: (a) precision, (b) recall, and (c) accuracy, which are defined as follows:

$$\text{precision} = \frac{TP}{TP+FP} \times 100\%, \quad (1)$$

$$\text{recall} = \frac{TP}{TP+FN} \times 100\%, \quad (2)$$

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%, \quad (3)$$

where the abbreviations TP, FP, FN, and TN denote the numbers of true positives, false positives, false negatives, and true negatives, respectively. In our four-way and ten-way classification, we regarded the first class as the positive class while others are negative classes for computing these metrics. Across the three classification tests, the defined assessment metrics all achieved results of 100%.

These results demonstrate that, without prior knowledge, measurement data suffice to classify fault types accurately and thereby provide pinpoint fault localization, which makes the fixing process efficient. In addition, because of the high sampling frequency (12 kHz) used and the high efficiency of the proposed CNN model, the fault types can be categorized correctly within 0.5 s: thus the proposed algorithm can localize faults in near-real time.

The proposed framework can also be used for a wide range of other applications with high metrics, including greater accuracy, precision and recall (summarized in Figure 2). Here we focus on three examples:

1. Airplane girder simulation damage data for fault diagnosis²⁸: We used a classification method to monitor the condition of a girder in an airplane. A frequent problem with aircraft girders is loose bolts. Our experiments classified four different positions of bolt looseness and five different degrees of looseness at each position. The algorithm achieved 100% accuracy in the four-way classification and five-way classification experiments described above.
2. Hydraulic system condition classification²⁷: Hydraulic system condition monitoring is a classification task. We chose CNN as the base model to make predictions for different conditions.

Four condition classifications corresponding to different hazard types and levels were conducted: 1) a three-way classification for cooler condition; 2) a four-way classification for valve condition; 3) a three-way classification model for internal pump leakage; and 4) a four-way classification model for hydraulic accumulator. The algorithm achieved accuracies of 100% in both cooler condition and valve condition classifications. Meanwhile, the pump leakage and hydraulic accumulator classifications also achieved satisfactory accuracies, at 98.19% and 99.10%, respectively.

3. NASA lithium-ion battery data for state of health (SOH) estimation³¹: Another benchmark of industrial lithium-ion battery data obtained by NASA was used to estimate battery SOH. CNN models are trained for this dataset and the smallest average RMSE value 0.0172 mm was achieved with respect to smallest error³⁴ of 0.0264 mm that has been achieved in previous related work. Detailed descriptions of the data structures and the established models for these applications and several further, diverse, cases can be found in the Supplementary Information.

Here we have demonstrated the effectiveness of deep learning for use in manufacturing applications. Using a unified CNN framework, we tested deep learning across a large number of critical diagnostic tasks in a variety of applications. This entails some limitations, which are identified in Materials and Methods. The proposed end-to-end framework could achieve 24/7 coverage via the provision of customized monitors for machines (with a function comparable to that of doctors in personalized medicine), and as a result it could facilitate the development of predictive, preventive, and customized maintenance and thus ensuring that the equipment does not break down.

Materials and Methods

Datasets

The datasets used in this manuscript are of the following types: open-accessible data; competition data; experimental data collected in our lab; and real production data provided by

industrial partners with permission. These datasets are composed of sensory current signals, force signals, vibration signals or acoustic emission signals, or their combinations, which are processed for the classification or regression tasks.

Main idea

We convert practical problems into supervised classification and regression tasks and solve them using deep learning technique. An end-to-end algorithm is proposed to automatically discover the hidden features needed for learning and prediction without prior knowledge. We develop a novel architecture of CNN that performs fault diagnosis and prediction and regression based on the raw data. The framework of the proposed CNN is shown in Figure 3, in which we construct a fully automated closed-loop system: a CNN model is fed with the sensory measurements and extracts the features for classification or prediction. The results learned by the CNN are then fed back to the machine for decision making, for example, whether any maintenance action is required. Algorithm 1 provides an illustration of the proposed CNN framework, dealing with fault diagnosis problem of the bearing dataset:

Algorithm 1 Outline of CNN training for a classification case

Input : The pre-processed vibration data and the defined label

Output: Trained CNN parameters

Initialize parameters;

repeat

Forward Propagation:

do

 Conducting convolution operation with the pre-processed vibration data using Equation (4).

 Use ReLU described in Equation (5) as the nonlinear activation function. Equation (6) describes the convolutional process.

 Max pooling function (7) is employed to extract the maximum feature values.

end;

 Conventional fully-connected layer is used for classification.

 The softmax function (8) is introduced for a probability of output.

 Compute the cross entropy with the loss function (10).

Backward Propagation:

 Compute the gradient using *Adam* and update network parameters.

until Maximum iterations;

Use the trained CNN to predict the category on the test sets.

Pre-processing

We normalize the measurements in each dataset in several ways as detailed in the Supplementary Information. More specifically, for some datasets with a small number of time-

course measurements, such as CWRU bearing dataset, we divided the total features to a constant length in each sample without affecting the periodicity of the data. For some datasets with a huge number of measurements, such as the **Case 2** in the Supplementary Information, we cropped out the first thousand number of measurements in each sample without destroying the salient features of the dataset. For prediction tasks, such as the **Case 8**, dataset was transformed by a standardisation as specified in the Supplementary Information.

Tuning parameters

We propose to fine-tune the CNN model according to different classification and prediction objectives, with a fixed max-pooling size of 1×2 . To extract fewer features, stride sizes (i.e., the sliding window size) in CNN models are set to be, for example, 500 or 1000 in a data sequence with tens of thousands of dimensions, 100 or 200 in a data sequence with thousands of dimensions. The basic components of the proposed CNN model are stacked with input data, CNN layers, and a fully connected layer (including an output layer). For classification problems, the number of nodes N in the output layer are equal to the number of fault types. For regression problems, N is set to one. For detailed model parameters of difference applications, please refer to Supplementary Information.

Convolutional neural networks

CNNs consist of convolutional layers, pooling layers, and fully connected layers with a final N -way prediction layer. The convolution layer uses a number of filters to discretely convolve with the input data. We define a vector $K \in \mathbb{R}^m$ of weights, a vector $I \in \mathbb{R}^k$ of raw data, and a constant b of a bias. In a convolutional process, stride is the distance between two sub-convolution windows, and we define it as parameter d . We define a sub-vector of I , i.e., $I^{(i)} = [I^{1+(i-1)d}, I^{2+(i-1)d}, \dots, I^{m+(i-1)d}]^T$ ($i = 1, 2, \dots, \frac{k-m}{d} + 1$). The idea of a one-dimensional convolution is to take the product between the vector K and the sub-vector $I^{(i)}$ of raw data, which reads as follows:

$$S^{(i)} = I^{(i)} * K + b = \sum_{j=1}^m I^{j+(i-1)d} K^j + b, \quad (4)$$

where K^j is the j^{th} element of vector K , $j=1,2,\dots,m$. When conducting a convolutional process, the number of filters (different filters have different initial values K) and b is set to determine the depth of the convolutional results. Since the process of convolution between each filter and data uses weight sharing, the number of training parameters and complexity of the model are greatly reduced. As a result, computational efficiency is improved.

An activation function named Rectified Linear Unit (ReLU) is followed by each convolutional layer, which has the following form:

$$f(S^{(i)}) \triangleq \max(0, S^{(i)}) . \quad (5)$$

ReLU prevents the saturation nonlinearities with respect to other functions when optimizer calculates gradient descent, meanwhile guarantees the sparsity in convolutional networks. To conclude, the entire convolution process for a sub-vector $I^{(i)}$ can be described as:

$$Y^i = f\left(\sum_{j=1}^m I^{j+(i-1)d} K^j + b\right), \forall i = 1, 2, \dots, \frac{k-m}{d} + 1, \quad (6)$$

leading to the convolutional results $Y = [Y^1, \dots, Y^i, \dots, Y^{\frac{k-m}{d}+1}]^T$.

Data is then fed into the pooling layer with the aim of down-sampling the number of parameters. The commonly used pooling method are max pooling, average pooling, and L_2 -norm pooling. Due to the improved performance achieved in practice, max pooling is chosen here:

$$M^i = \max_{\ell=1}^p Y^{\ell+(i-1)e}, \forall i = 1, 2, \dots, \frac{k-m}{d} + 1, \quad (7)$$

where p is the pooling size, and e is the stride size in max pooling.

After convolution and pooling, the data is fed into a fully connected layer. Data is transformed into a one-dimensional structure to facilitate data processing in the fully connected layer. The final layer is the output of the model with size N .

For classification problems, the activation function is selected as softmax in classification problem. The probabilities of the faulty types of the i_{th} training measurement are calculated by

the softmax classifier $P(y^{(i)} = n|X^{(i)}; W^{(i)}) =$

$(P(y^{(i)} = 1|X^{(i)}; W^{(i)}), \dots, P(y^{(i)} = N|X^{(i)}; W^{(i)}))^T$ are defined as follows:

$$P(y^{(i)} = n|X^{(i)}; W^{(i)}) = \frac{e^{(W_n^{(i)})^T X^{(i)}}}{\sum_{j=1}^N e^{(W_j^{(i)})^T X^{(i)}}}, \quad (8)$$

where vector $X^{(i)}$, $i=1,2,\dots,q$, represents the penultimate layer in fully connected layers (output layer is the last layer in fully connected layers), matrix $W^{(i)}$ is the weights for different faulty types in the i th training measurement and $W_j^{(i)}$ is the j th column of matrix $W^{(i)}$.

For regression, the estimated result $y_{\text{estimate}}^{(i)}$ of the i th training measurements is represented as follows:

$$y_{\text{estimate}}^{(i)} = \frac{1}{1 + e^{-(W^{(i)})^T X^{(i)}}}, \quad (9)$$

where the definition of vector $X^{(i)}$, $i = 1, \dots, q$, is the same as Equation (8), while vector $W^{(i)}$ is the weights for the penultimate layer in fully connected layers of the i th training measurements.

In the training process, to minimize the difference between the predicted scores and the ground labels in the training data, we need to design a proper loss function and specify an optimizer. Two loss functions used for classification and regression are cross-entropy L_{ce} and least-squares L_{ls} , respectively, which are described in Equations (10) and (11):

$$L_{\text{ce}} = -\frac{1}{q} \sum_{i=1}^q \sum_{j=1}^N 1\{y^{(i)} = j\} \log \frac{e^{(W_j^{(i)})^T X^{(i)}}}{\sum_{j=1}^N e^{(W_j^{(i)})^T X^{(i)}}} + (1 - 1\{y^{(i)} = j\}) \log \left(1 - \frac{e^{(W_j^{(i)})^T X^{(i)}}}{\sum_{j=1}^N e^{(W_j^{(i)})^T X^{(i)}}} \right), \quad (10)$$

$$L_{\text{ls}} = \frac{1}{q} \sum_{i=1}^q (y^{(i)} - y_{\text{estimate}}^{(i)})^2. \quad (11)$$

Specially, the term $1\{y^{(i)} = j\}$ in Equation (10) is the logical expression that always returns either zeros or ones. Meanwhile, $y^{(i)}$ and $y_{\text{estimate}}^{(i)}$ in Equation (11) are real output and estimated output of the i th training measurement. Once the cost function has been chosen, we use standard

optimizers such as stochastic gradient descent (SGD) or Adam for parameter training in back-propagation to update weights. The final CNN model weights refresh until the predefined maximum iteration to yield a lower loss.

Limitation of the proposed method

Limitations include the following:

- a. the method has a few hyper parameters to tune in order to achieve best performance, such as kernel size, the number of strides, the number of layers and learning rate. These parameters can be tuned using standard methods such as cross validation;
- b. given the number of parameters in the constructed model, it requires a large amount of data to train;
- c. due to the intrinsic complexity of the hidden layers, it is difficult to interpret the mechanism of the learned neural network model.

Data availability statement

The bearing fault, aircraft girder, and aero engine blade processing datasets can be downloaded at Manufacutring Network Platform <http://mad-net.org:8765/explore.html?t=0.597370213951085> built by our lab. The CRWU bearing dataset is available at <http://www.eecs.cwru.edu/laboratory/bearing>. The NASA tool wear dataset can be downloaded from <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>. NASA and CALCE battery datasets are available at <http://ti.arc.nasa.gov/project/prognostic-data-repository> and <https://web.calce.umd.edu/batteries/data.htm#>, respectively. The hydraulic system dataset is available at <https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems#>. The experimental data including gearbox, and tool broken datasets are available from the corresponding author upon request.

REFERENCES

1. James Manyika *et al.* *Manufacturing the future: The next era of global growth and innovation.* McKinsey Global Institute (2012).
2. McKinsey & Company. *Developing the future of manufacturing.* (2017).
3. Isermann, R. *Fault-diagnosis systems: An introduction from fault detection to fault tolerance.* (2006).
4. Davis, J., Edgar, T., Porter, J., Bernaden, J. & Sarli, M. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Comput. Chem. Eng.* **47**, 145–156 (2012).
5. Kusiak, A. Smart manufacturing. *Int. J. Prod. Res.* **56**, 508–517 (2018).
6. Kusiak, A. Smart manufacturing must embrace big data. *Nature* **544**, 23–25 (2017).
7. Liu, R., Yang, B., Zio, E. & Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.* **108**, 33–47 (2018).
8. Jack, L. B. & Nandi, A. K. Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. *Mech. Syst. Signal Process.* **16**, 373–390 (2002).
9. Li, J. *et al.* Feature Selection: A Data Perspective. *ACM Comput. Surv.* **50**, 94 (2016).
10. Rauber, T. W., De Assis Boldt, F. & Varejão, F. M. Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Trans. Ind. Electron.* **62**, 637–646 (2015).
11. Liu, H., Liu, C. & Huang, Y. Adaptive feature extraction using sparse coding for machinery fault diagnosis. *Mech. Syst. Signal Process.* **25**, 558–574 (2011).
12. Shao, H., Jiang, H., Zhang, H. & Liang, T. Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network. *IEEE Trans. Ind. Electron.* **65**, 2727–2736 (2018).
13. Basheer, I. A. & Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **43**, 3–31 (2000).
14. Furey, T. S. *et al.* Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16**, 906–914 (2000).

15. Jang, J. S. R. ANFIS: Adaptive-Neural-Based Fuzzy Inference System. *IEEE Trans. Syst. Man Cybern.* **23**, 665–685 (1993).
16. Krogh, A. What are artificial neural networks? *Nat. Biotechnol.* **26**, 195–197 (2008).
17. Afram, A., Janabi-Sharifi, F., Fung, A. S. & Raahemifar, K. Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy Build.* **141**, 96–113 (2017).
18. Widodo, A. & Yang, B. S. Support vector machine in machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process.* **21**, 2560–2574 (2007).
19. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
20. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
21. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
22. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **60**, 1097–1105 (2012).
23. Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
24. Li, W., Field, K. G. & Morgan, D. Automated defect analysis in electron microscopic images. *npj Comput. Mater.* **4**, 1–9 (2018).
25. Abdel-Hamid, O. *et al.* Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **22**, 1533–1545 (2014).
26. Download a Data File | Bearing Data Center. Available at: <http://csegroups.case.edu/bearingdatacenter/pages/download-data-file>. (Accessed: 28th August 2018)
27. Helwig, N., Pignanelli, E. & Schutze, A. Condition monitoring of a complex hydraulic system using multivariate statistics. in *Conference Record - IEEE Instrumentation and Measurement Technology Conference* 210–215 (2015).
doi:10.1109/I2MTC.2015.7151267

28. Mad Net. Available at: <http://mad-net.org:8765/explore.html?t=0.597370213951085>. (Accessed: 28th August 2018)
29. He, Q., Guo, Y., Wang, X., Ren, Z. & Li, J. Gearbox Fault Diagnosis Based on RB-SSD and MCKD (In Chinese) . *Zhongguo Jixie Gongcheng/China Mech. Eng.* **28**, 1528–1534 (2017).
30. A. Agogino and K. Goebel (2007). BEST lab, UC Berkeley. "Milling Data Set ", NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA. Available at: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>. (Accessed: 28th August 2018)
31. B. Saha and K. Goebel (2007). 'Battery Data Set', NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA. Available at: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>. (Accessed: 1st September 2018)
32. He, W., Williard, N., Osterman, M. & Pecht, M. Prognostics of lithium-ion batteries based on Dempster-Shafer theory and the Bayesian Monte Carlo method. *J. Power Sources* **196**, 10314–10321 (2011).
33. Tao, J., Liu, Y. & Yang, D. Bearing Fault Diagnosis Based on Deep Belief Network and Multisensor Information Fusion. *Shock Vib.* **2016**, 1–9 (2016).
34. Dong, G., Chen, Z., Wei, J. & Ling, Q. Battery health prognosis using brownian motion modeling and particle filtering. *IEEE Trans. Ind. Electron.* **65**, 8646–8655 (2018).

Acknowledgements This work was supported by National Natural Science Foundation of China through projects 91748112, 5135004. We thank Prof. Bin Li, Dr. Bo Luo, Prof. Tao Zhou, Prof. Sijie Yan for early insightful discussion. We thank Prof. Shenfang Yuan, Prof. Jihong Chen, Prof. Limin Zhu, Prof. Ke Li for kindly sharing their valuable data for us. We thank Mr. Anthony Haynes, and

Mr. David MacDonald from Liwen Bianji, Edanz Editing China (www.liwenbianji.cn/ac), for editing the English text of a draft of this manuscript.

Author Contributions Y.Y. conceptualized the algorithm. G.M. developed the algorithms under the supervision of H.D. and Y.Y.. H.D. G.M. C.C., B. Z. and Y.Y. analyzed the experimental data. All authors designed and discussed the study and wrote the paper.

Author Information The authors declare no competing financial interests. Readers are welcome to comment on the online version of the paper. All data and computer code needed to evaluate the conclusions in the paper are available from the corresponding author upon request.

Correspondence and requests for materials should be addressed to Y.Y. (yue@hust.edu.cn) and H.D. (dinghan@hust.edu.cn).

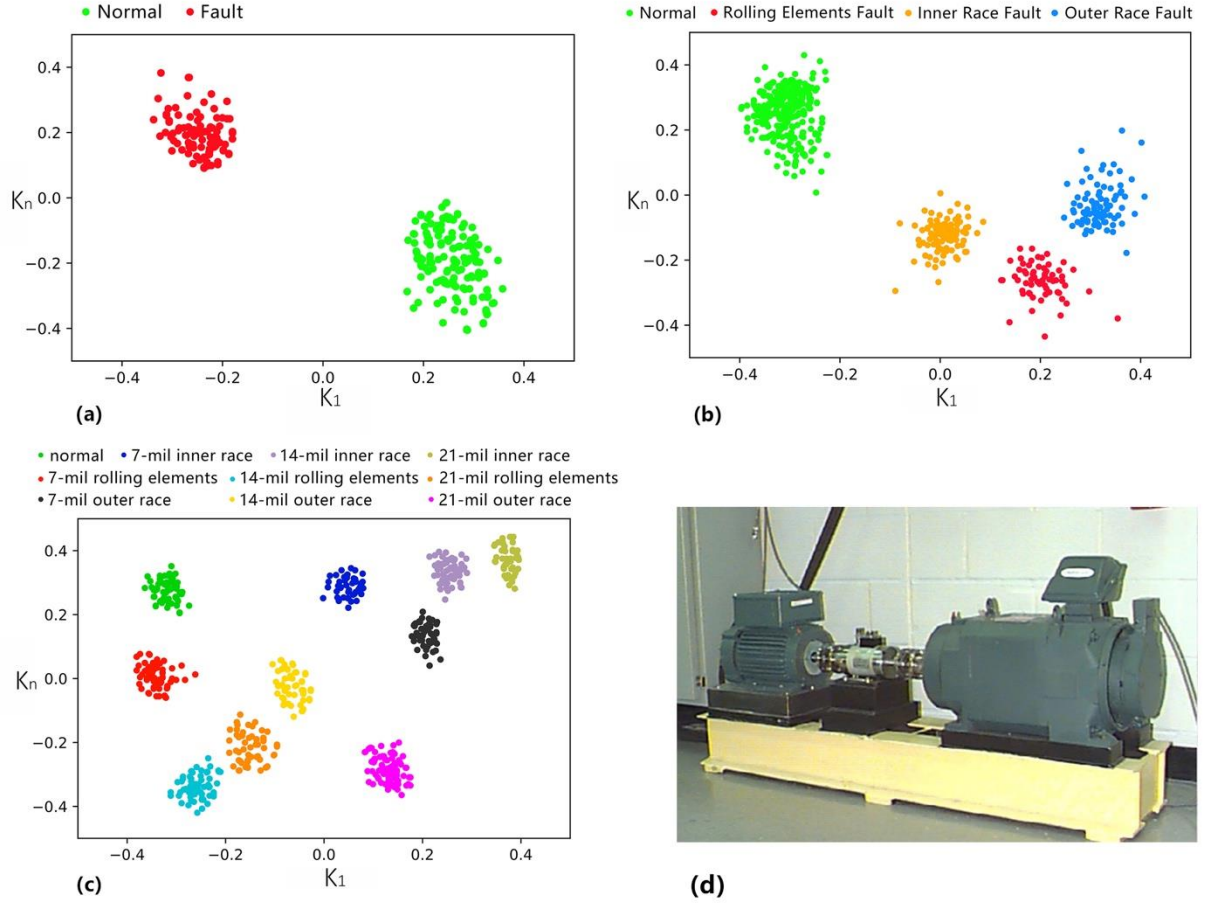


Fig. 1: Classification results on the CWRU bearing test dataset, visualized in a 2D feature space. (a) Binary classification task (normal and fault). (b) Four-way classification task (normal, ball fault, inner race fault, and outer race fault). (c) Ten-way classification task (normal and other nine types of faults on three bearing with different fault diameters). (d) The experiment platform in CWRU bearing data centre.

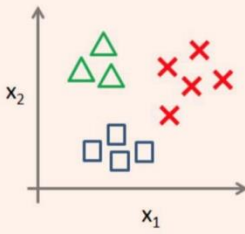

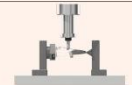

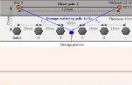



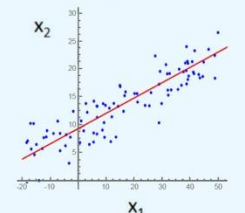



Type	Data	Platform	Result			
			Classification	Accuracy	Precision	Recall
Classification 	Rolling bearing fault classification. Dataset: CWRU ²⁶		Binary classification Four-way classification Ten-way classification	100% (132/132) 100% (132/132) 100% (132/132)	100% 100% 100%	100% 100% 100%
	Tool broken classification. Dataset: our experiment		Two-way classification	100% (53/53)	100%	100%
	Rolling bearing fault classification. Dataset: our experiment ²⁸		Four-way classification	100% (90/90)	100%	100%
	Airplane girder fault classification. Dataset: our experiment ²⁸		Five-way classification Four-way classification	100% (120/120) 100% (120/120)	100% 100%	100% 100%
	Aero engine blades processing classification. Dataset: our experiment		Four-way classification	95.92% (94/98)	95.65%	100%
	Gearbox fault classification. Dataset: our experiment ²⁹		Two-way classification Four-way classification	100% (371/371) 99.46% (369/371)	100% 100%	100% 98.60%
	Hydraulic system fault classification. Dataset: UC Irvine ²⁷		Three-way classification Four-way classification	100% (221/221) 100% (221/221)	100% 100%	100% 100%
Regression 	Tool wear value prediction. Dataset: NASA ³⁰		MSE	MAE	R ²	RMSE
	Battery state of health estimation. Dataset: NASA ³¹		0.0071	0.0671	0.8725	0.0836
	Battery state of health estimation. Dataset: CALCE ³²		4.2×10^{-6}	0.0119	0.9600	0.0172
			2.4×10^{-4}	0.0063	0.9954	0.0137

Fig. 2: Summary of classification and regression results of different datasets. The datasets for classification problems include: CWRU bearing data; tool broken data; bearing data; airplane girder data; blades processing data; gearbox data; and hydraulic system data. The datasets for supervised regression problems include: NASA tool wear data; NASA battery data; and the CALCE data. Specifically, for the multi-classification problem, we define the first class as the positive class to calculate the precision, recall and accuracy according to Equations (1), (2) and (3).

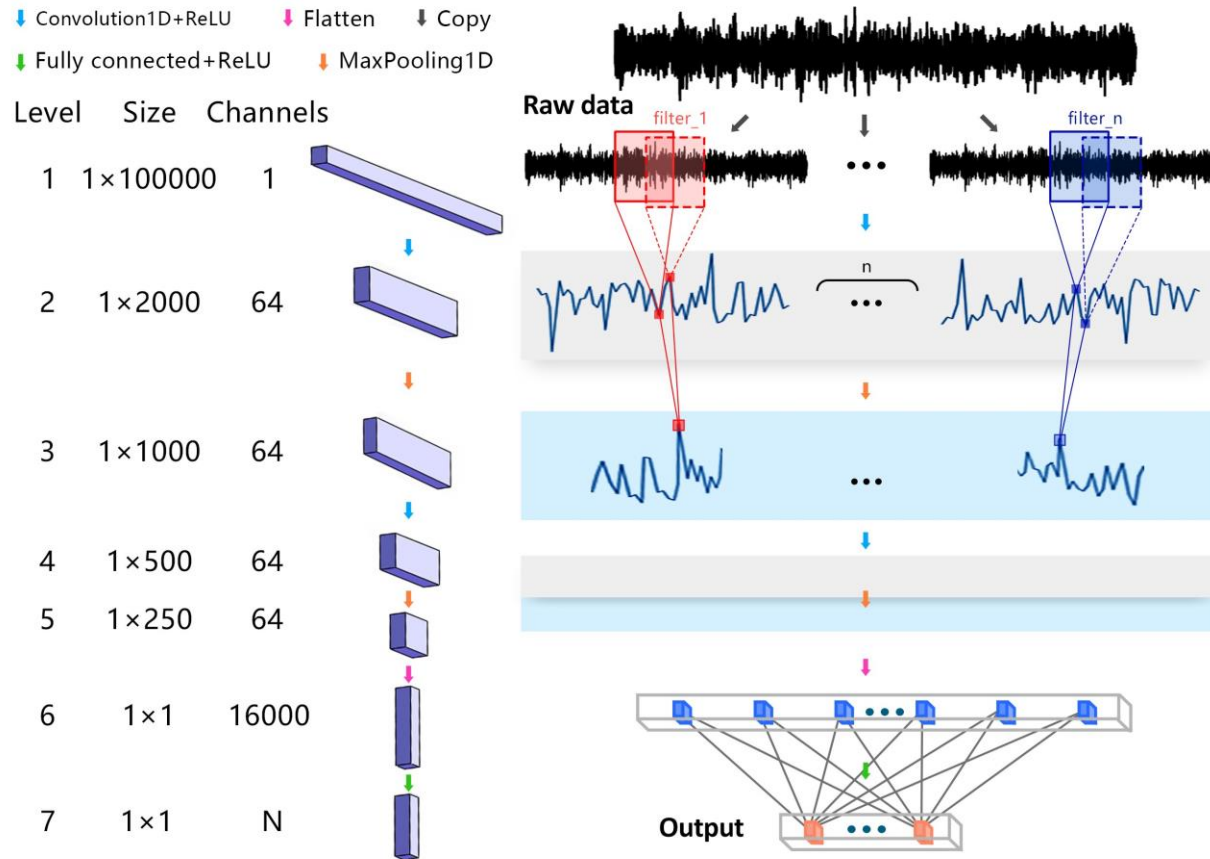


Fig. 3: An illustration of a CNN model for a classification/regression task. This framework is a fully-automated and closed-loop system. Raw data is generated by the machinery and processed and it goes through the CNN model. The output layer with $1 \times N$ size result (N is an integer for classification or equals to 1 for regression) from the CNN model is fed back to the machinery model for decision making. The input raw data is passed through convolutional layers, max pooling layers and fully connected layers as explained in the Materials and Methods.