

# *An Intelligent Approach to Detecting Novel Fault Classes for Centrifugal Pumps Based on Deep CNNs and Unsupervised Methods*

*Mahdi Abdollah Chalaki, Daniyal Maroufi, Mahdi Robati, Mohammad Javad Karimi, Ali Sadighi*

*School of Mechanical Engineering  
University of Tehran  
Tehran, Iran  
asadighi@ut.ac.ir*

**Abstract**—Despite the recent success in data-driven fault diagnosis of rotating machines, there are still remaining challenges in this field. Among the issues to be addressed, is the lack of information about variety of faults the system may encounter in the field. In this paper, we assume a partial knowledge of the system faults and use the corresponding data to train a convolutional neural network. A combination of t-SNE method and clustering techniques is then employed to detect novel faults. Upon detection, the network is augmented using the new data. Finally, a test setup is used to validate this two-stage methodology on a centrifugal pump and experimental results show high accuracy in detecting novel faults.

**Keywords**—centrifugal pump, condition monitoring, classification, convolutional neural network, t-SNE, clustering.

## I. INTRODUCTION

Diagnosis of rotating machinery faults is essential since it leads to reduction of unplanned downtime and maintenance costs and improves reliability and safety of operations. Centrifugal pumps are one of the most-commonly used equipment in a wide range of industries. Researchers have done extensive research on implementing online condition monitoring methods in order to detect faults effectively.

These methods comprise data-driven approaches such as Multi-layer Perceptron (MLP) [1], Random forests [2], Support Vector Machines [3], etc., which were used alongside extracted statistical features and signal-processing techniques. Recently, various deep learning approaches have shown remarkable results on classification tasks [4], [5]. These algorithms mainly use labeled raw time-domain sensor signals as input and learn the pattern within this data to extract main features and classify the faults.

In spite of these impressive results, there are remaining challenges that limit the application of these methods in real-world plants. The utilization of most existing methods is limited by some open issues such as domain adaptation problems, imbalanced datasets and missing labels. Different approaches are taken to address each of these issues in recent studies [4], [6], [7]. However, the problem of novel faults was not considered by any of them.

To develop statistical models capable of classifying machine faults, sufficient amount of labeled data is required. Such a data is not readily available in industry settings. Alternatively, labeled data can be generated in the laboratory by introducing various faults and using them for model development. However, novel faults not seen by the model will result in misclassifications and poor performance of the monitoring system. Anomaly detection algorithms, on the other hand, can distinguish healthy state of the system from any anomalous behavior [8], [9]. Nevertheless, different faults cannot be discerned using these methods. In this paper, we propose a new approach that combines a supervised convolutional neural network (CNN) model with an unsupervised clustering network. Consequently, not only does the system detect previously observed faults, but it also groups new faults into new clusters and continuously monitors their types.

## II. AN INTRODUCTION TO CNNs

Convolutional Neural Networks (CNNs) are capable of learning high-level features from input data, which allows them to provide accurate classification. There are three types of layers in convolutional neural networks: convolutional, pooling, and fully connected. In most classification and regression applications, the last layer is usually a fully connected layer. A CNN is based on three ideas: shared weights, local receptive fields, and pooling. In contrast to fully connected neural networks, in a CNN only a small localized area of the input data is connected to each node of a convolutional layer. Thus, it reduces the number of parameters of the network and makes it easier to train.

In a CNN architecture, a block consists of a convolution layer and a pooling layer. A number of these blocks followed by fully-connected layers form a deep network that is used for feature extraction and classification. Finally, a softmax function predicts the probability of each class [10], [11]. Mathematically, it is determined by:

$$y = f(Wa + b_c) \quad (1)$$

where  $y$  denotes the predicted labels;  $W$  denotes the weight matrix between  $a$ , neurons in fully-connected layer and connected nodes in the output layer;  $b_c$  represents the bias, and  $f$  is the softmax activation function. The training of the CNN is done by minimizing the cross-entropy loss, which is calculated as:

$$H(r, p) = \sum_i r_i \log(p_i) \quad (2)$$

According to the true label,  $r$  is either 0 or 1, and  $p$  is the output probability. The error between predicted probabilities and true labels can be measured by cross-entropy loss. Finally, weights will be updated after each iteration using a backpropagation algorithm based on gradient descent methods.

### III. PROPOSED METHODOLOGY

As mentioned before, although deep learning methods have shown great success in classifying the faults that they have been trained on, they fail to detect new types of faults. Here, a new network is proposed to address this problem.

This proposed network consists of two stages: In the first one, a Deep Convolutional Neural Networks with Wide First-layer Kernels (WDCNN) [12] is used to extract features and classify faults, and in the second stage faults are examined for their novelty. If a novel fault is encountered, the monitoring system could automatically detect it, make a new label and modify the current network to include the new class.

#### A. Stage 1

In the first stage, a network capable of classifying the faults of the system is built. The structure of this neural network is heavily borrowed from [12]. Raw vibration signals are used as inputs to this network. In the first convolutional

layer, feature extraction and high-frequency noise suppression are conducted by using wide kernels. The network structure is visualized in Fig. 1.

Batch normalization [13] and Dropout [14] layers after each convolutional layer are employed to decrease the training process duration and prevent overfitting. Additionally, categorical cross entropy is used to measure loss function and the Adam optimizer [15] is employed instead of the classic SGD to train the model.

After a successful training stage, this convolutional neural network can classify faults into a certain number of classes determined by our training dataset. As shown in Fig. 1, the last fully connected layer is made up of 64 neurons that can extract the features of each sample. These features play an essential role in the second stage.

#### B. Stage 2

In this stage, the dimensions of the features extracted in the previous stage are reduced by applying the t-SNE method [16]. In our case, the features can be described as a 64-dimensional vector, whose dimension could be reduced to two. We then use these two-dimensional features to form clusters. As it is shown in Fig. 2, for 600 training samples (from 6 different labels), each class forms a unique cluster after this process.

If the machine faces a faulty condition of any kind, the system will start collecting 100 new samples. By averaging over the network predictions for these new samples, it can be concluded that there exists a problem. In order to determine if these samples form a new cluster, they will be clustered with previous samples. K-means algorithm is used in this process to cluster the samples [17].

K-means clustering is an unsupervised learning technique that minimizes intra-cluster variances to divide  $n$  unlabeled samples into a pre-determined number of clusters with the nearest mean.

The goal here is to determine if the newly collected samples create a distinct cluster or not. Thus, firstly we have to try different number of clusters to partition samples into, and

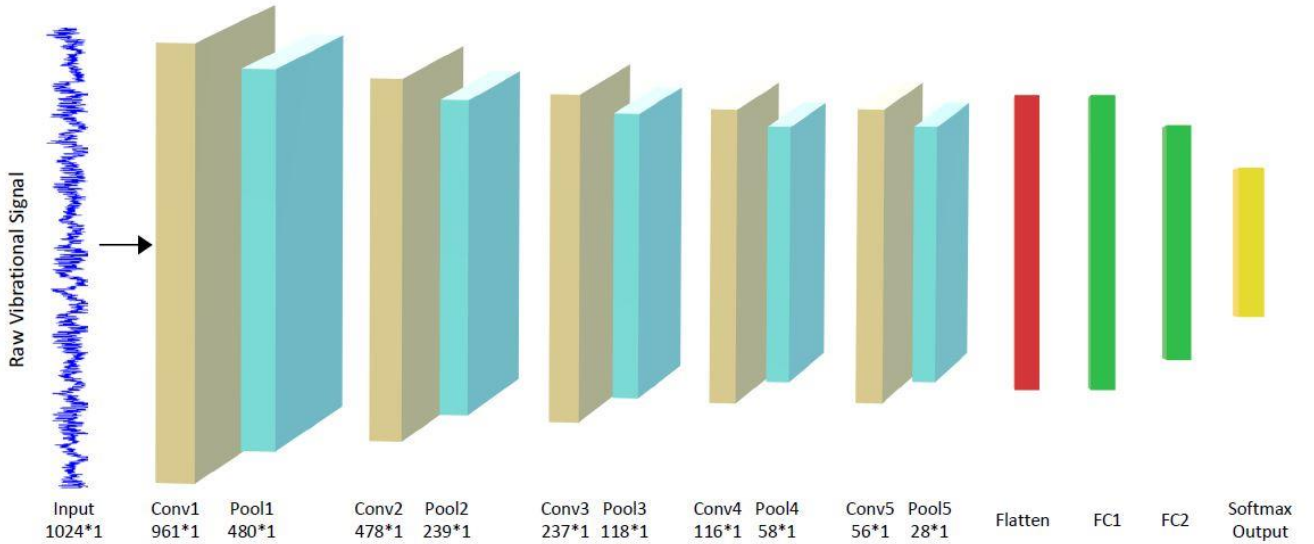


Fig. 1. Architecture of the WDCNN model. “Conv” stands for a Convolutional layer, “Pool” for a Max-Pooling layer, and “FC” for a Fully-Connected layer.

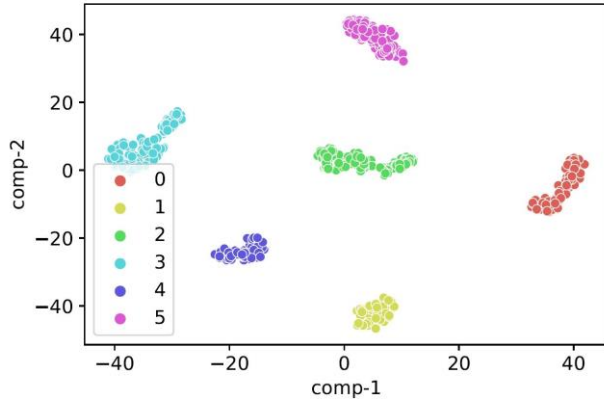


Fig. 2. Feature visualization of the last hidden FC layer of WDCNN network.

then use an evaluation method to find the optimal number of clusters.

External, relative, and internal techniques are usually used for cluster validation. Using the internal criteria, which is an unsupervised way to estimate the quality of a clustering solution, we measure how compacted and separated the clusters are. This is a practical method that can help us to determine how many clusters exist.

There are several internal criteria indices suggested in the literature including Davies–Bouldin index [18], Dunn’s Index [19], The Silhouette index [20] and indices based on Sum of Squared estimate of Errors (SSE). The proposed method uses the latest one in combination with an inflection point selection method to find the knee-point in the “Inertia Number of clusters” diagram. Inertia is defined here as the squared distance of the samples to their nearest cluster center [21]. This method is robust to the absolute value of the data points (Unlike other methods in which a limit should be set to find out if two samples are in the same neighborhood).

Knee-point analysis indicates whether a new cluster has appeared in the data. Should this occur, the system begins collecting more data. As training a new model that includes a new label requires a balanced dataset, this data collection step is necessary. After the data collection stage, we turn back to Stage 1. The next step is to modify the softmax layer of our old model and add a neuron to its neurons. After this modification, the model will be trained again with the new dataset.

#### IV. EXPERIMENTAL INVESTIGATION

##### A. Experimental Setup

To evaluate the proposed method, an experimental setup was developed as illustrated in Fig. 3. This system consists of a reservoir, a centrifugal pump, a rotameter, and two valves that control water flow. A 1-hp, single-phase induction motor drives the water pump.

Two valves are located at the suction and discharge sides of the pump. Cavitation is staged by changing the inlet valve position, and flow rate is controlled by adjusting the outlet valve.

The vibration signal is sampled at 1.6 kHz using a LIS3DSH accelerometer module that is mounted on the



Fig. 3. Experimental setup.

pump’s discharge side. This module is a three-axis accelerometer which includes an 800 Hz anti-aliasing filter. Due to the limited memory of our microcontroller, our model only uses the data from one of the axes. In addition, using data from a single axis means fewer parameters and hence, less computational resources.

The accelerometer module is connected to an STM32F4-Discovery microcontroller using Serial Peripheral Interface (SPI) protocol. Sampled data is first stored on the microcontroller memory and then transmitted to the server through two XBee modules, one installed on the main device which is mounted on the pump, and the other in the monitoring station and server. XBee modules use ZigBee protocol to communicate with each other, and UART protocol to interface with the microcontroller on the device and the Raspberry Pi board on the user end. This process is illustrated in Fig. 4.

The Raspberry Pi board first stores raw data in an InfluxDB database, and then does the pre-processing task. In this task, the data is reshaped to our favorable form, get normalized and then stored in the database again. At last, this pre-processed data is sent to the CNN model for continuous evaluation. Finally, the pre-processed data batches and their corresponding predicted labels are shown in a Grafana dashboard for online monitoring purposes of the end-user. Fig. 5 schematically shows the software architecture.

##### B. Dataset Description

Our dataset consists of 6 types of health conditions; a healthy condition and five different faulty types. Each sample contains 1024 data points. Additional details are shown in Table I.

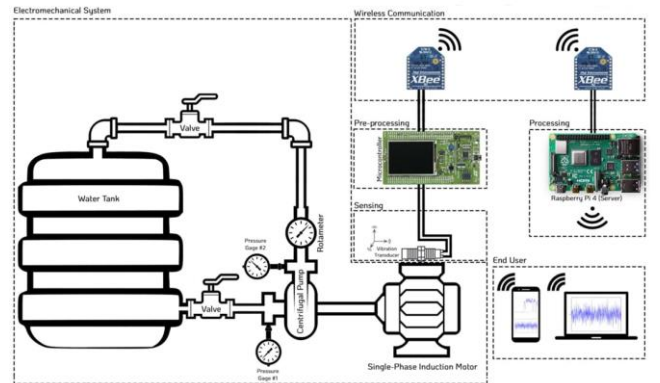


Fig. 4. Schematic picture of the Experimental setup and its communication with the user-end part.

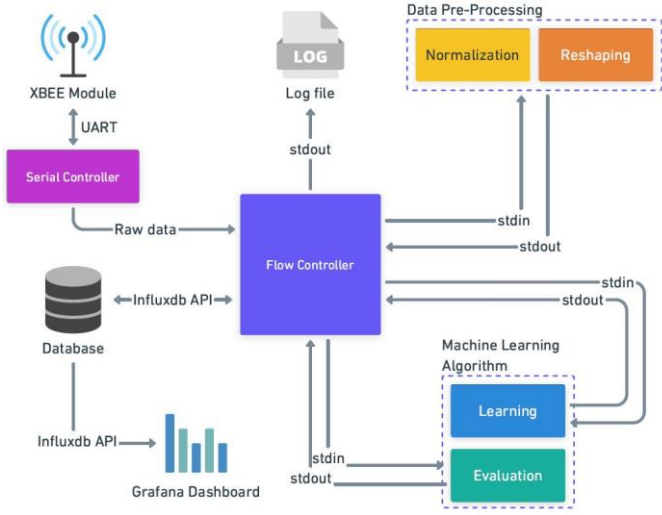


Fig. 5. Software architecture.

TABLE I. DESCRIPTION OF CENTRIFUGAL PUMP DATASET

Type	Train	Test	Flow range(L/min)	Label
Healthy	1460	360	170 - 220	0
Low flow rate	1120	280	80 - 120	1
Cavitation	1440	360	120 - 170	2
Major defect	1440	360	150 - 200	3
Minor defect	1040	360	130 - 160	4
Crack	1280	320	150 - 200	5

Fig. 6 provides a better understanding of the impeller-related faults by illustrating vane tip defects (corresponding to the third and fourth labels) and impeller crack fault (corresponding to the fifth label). In addition, low flow rate

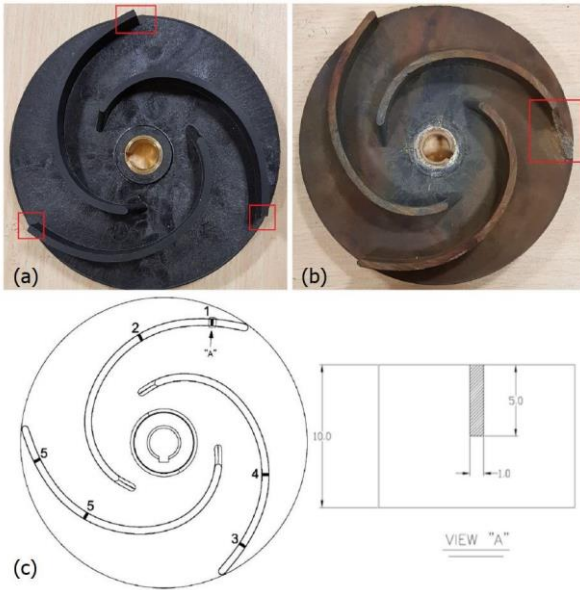


Fig. 6. (a) vane tip minor defect (b) vane tip major defect (c) impeller crack fault [8].

and cavitation conditions are adjusted by using the valves that were previously mentioned.

## V. RESULTS

In the first stage of training, we just used the first five labels to train the model and the sixth label (Crack fault) is left out for evaluation of the proposed method. It is a more realistic situation where we are able to only make a limited number of faults for training. At the end of this phase, our network can successfully detect samples from each class. Fig. 7 depicts the confusion matrix at the end of the first stage.

In the second stage, the crack fault is introduced to the model. Averaging over the predictions of the last 100 samples shows that the pump is not in a healthy condition. To determine if this fault belongs to one of the existing labels, we perform the second stage explained in Section IV. These 100 new faulty samples are combined with 500 random samples from our initial dataset. The output of applying the t-SNE on these 600 samples is shown in Fig. 2.

To automatically find the number of clusters, K-means algorithm is applied with different numbers of clusters ranging from 1 to 20. This helps us to find the knee-point of this diagram, as well as the number of clusters. In our case, the procedure indicates that there are six clusters now (Fig. 8).

Having a new cluster, sufficient data is needed to train a new model. In an actual situation, this means collecting data for a determined period of time. In our experiment, we used the data from Class 5. A slight modification was also necessary (changing the softmax layer so it will have six outputs). To evaluate the proposed method, we used K-fold Cross-Validation with five folds. The results of this test on our new model are shown in Table II. As shown in Table II, the proposed method could successfully address the problem of classifying new emerging faults in a centrifugal pump. Fig. 9 illustrates confusion matrix of the re-trained model.

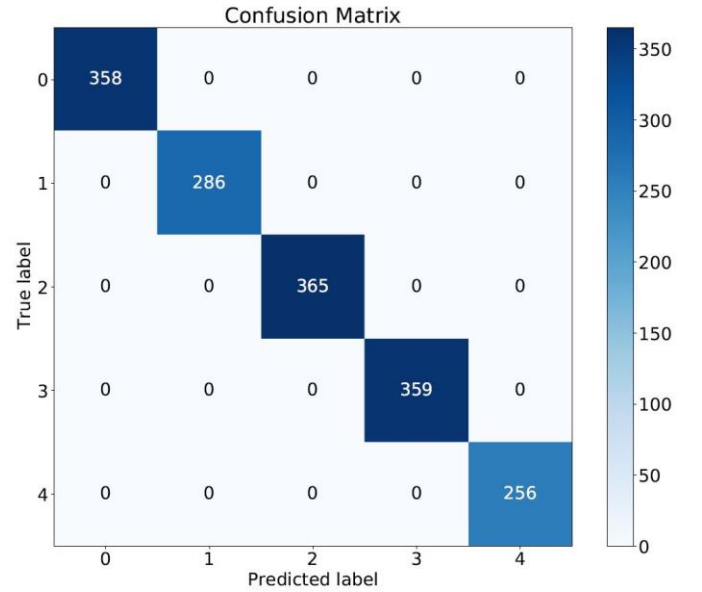


Fig. 7. Results of the WDCNN network with 5 classes.



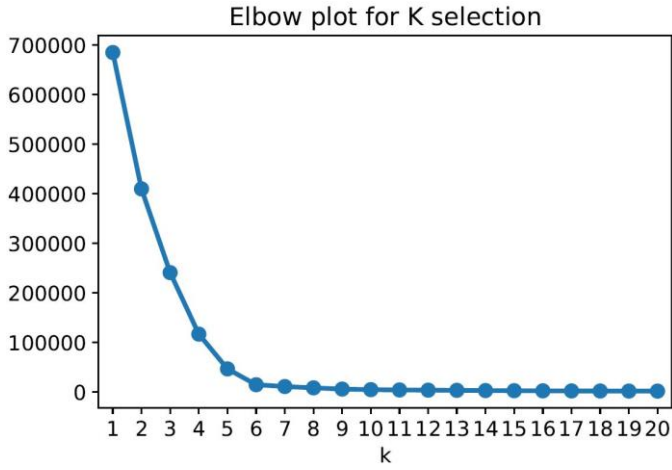


Fig. 8. Sum of Squared estimate of Errors (SSE) for different number of clusters

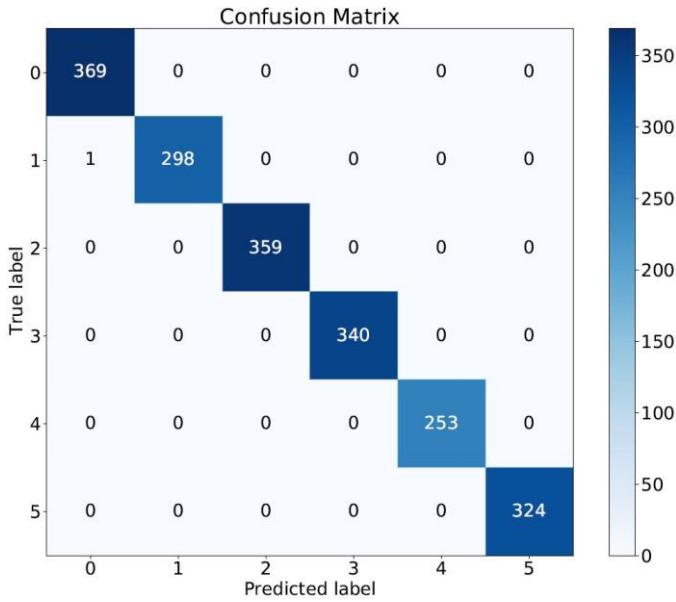


Fig. 9. Results of the re-trained network with 6 classes.

## VI. CONCLUSION

The paper presented a new two-stage methodology to fault detection of rotating machines. In the first stage, prior information about the system faults was used to develop a CNN capable of classifying defects. The second stage comprised t-SNE and unsupervised clustering algorithms to detect novel fault classes and augment the existing CNN accordingly. A hydraulic test rig was used to implement and show the performance of the proposed method where high accuracy in detecting novel faults was obtained.

## REFERENCES

- [1] A. Hajnayeb, A. Ghasemlooia, S. Khadem, and M. H. Moradi, "Application and comparison of an ann-based feature selection method and the genetic algorithm in gearbox fault diagnosis," *Expert systems with Applications*, vol. 38, no. 8, pp. 10205–10209, 2011.
- [2] B.-S. Yang, X. Di, and T. Han, "Random forests classifier for machine fault diagnosis," *Journal of mechanical science and technology*, vol. 22, no. 9, pp. 1716–1725, 2008.
- [3] P. Santos, L. F. Villa, A. Renones, A. Bustillo, and J. Maudes, "An svm-based solution for fault detection in wind turbines," *Sensors*, vol. 15, no. 3, pp. 5627–5648, 2015.
- [4] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, pp. 433–452, 2020.
- [5] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. Gao, "Deep learning and its applications to machine health monitoring: A survey. arxiv 2016," *arXiv preprint arXiv:1612.07640*, 2016.
- [6] X. Li and W. Zhang, "Deep learning-based partial domain adaptation method on intelligent machinery fault diagnostics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4351–4361, 2020.
- [7] Q. Wang, G. Michau, and O. Fink, "Missing-class-robust domain adaptation by unilateral alignment," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 663–671, 2020.
- [8] M. Jamimoghaddam, A. Sadighi, and Z. Araste, "Esa-based anomaly detection of a centrifugal pump using self-organizing map," in *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 2020, pp. 1–5.
- [9] S. M. H. Maasoum, A. Mostafavi, and A. Sadighi, "An autoencoder-based algorithm for fault detection of rotating machines, suitable for online learning and standalone applications," in *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 2020, pp. 1–6.
- [10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
- [12] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.
- [13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
- [14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
- [16] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [17] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

TABLE II. TEST ACCURACY OF THE RE-TRAINED MODEL WITH K-FOLD CROSS VALIDATION

Fold	Test Accuracy (%)
1	99.948
2	100.0
3	100.0
4	100.0
5	100.0
Average	99.989 ± 0.021

- [18] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI1, no. 2, pp. 224–227, 1979.
- [19] J. C. Dunn†, "Well-separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974. [Online]. Available: <https://doi.org/10.1080/01969727408546059>
- [20] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377042787901257>
- [21] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.