sensors

Article

Deep Learning Methods for Fault Detection and Classification in MMC-HVDC systems

Qinghua Wang [1,3], Yuexiao Yu [2], Hosameldin O. A. Ahmed [3], Mohamed Darwish [3], and Asoke K. Nandi Fellow, IEEE [3,*]

[1] School of Mechatronic Engineering, Xi'an Technological University, Shaanxi 710021, China; wqhhuazi@163.com
[2] State Grid Sichuan Electric Power Research Institute of China, Chengdu 610094, China; yuyuexiao333@hotmail.com
[3] College of Engineering, Design and Physical Sciences, Brunel University London, UK; hosameldin.ahmed2@brunel.ac.uk; mohamed.darwish@brunel.ac.uk;
* College of Engineering, Design and Physical Sciences, Brunel University London, UK; asoke.nandi@brunel.ac.uk; Tel.: +44 (0)1895 266119

Received: date; Accepted: date; Published: date

Abstract: As there is not much literature about deep learning-based fault diagnosis for modular multilevel converters (MMCs) and comparison among deep learning methods used in fault diagnosis for MMC, two deep learning methods, namely, Convolutional Neural Networks (CNN) and Auto Encoder based Deep Neural networks (AE-based DNN) as well as stand-alone Softmax classifier are explored for the detection and classification of faults of MMC-based high voltage direct current converter (MMC-HVDC). Only AC-side three-phase current and the upper and lower bridges' currents of the MMCs are used directly by our proposed approaches without any explicit feature extraction or feature subset selection. The two-terminal MMC-HVDC system is established in PSCAD/EMTDC to verify and compare our methods. The simulation results indicate CNN, AE-based DNN, and Softmax classifier can detect and classify faults with high detection accuracy and classification accuracy. Compared with CNN and AE-based DNN, the Softmax classifier behaved better in detection and classification accuracy as well as testing speed. The detection accuracy of AE-based DNN is a little better than CNN, while CNN needs less training time than the AE-based DNN and Softmax classifier.

Keywords: MMC-HVDC; fault detection; fault classification; CNN; AE-based DNN; Softmax classifier; classification accuracy; speed

1. Introduction

With the increasing application of modular multilevel converter-based high-voltage direct current (MMC-HVDC) systems, the reliability of MMC is of major importance in ensuring power systems are safe and reliable. Topology configuration redundant strategies of fault-tolerant systems are useful methods to improve reliability which can be achieved by using more semiconductor devices as switches in an SM [1] or integrating redundant SMs into the arm submodule [2]. But, it is well to remember that, fault detection is a precondition for fault-tolerant operation which is needed to be as fast and accurate as possible to ensure converter continuous service. Therefore, Fault detection and classification are one of challenging tasks in MMC-HVDC systems to improve its reliability and thus reducing potential dangers in the power systems because there are a large number of power electronic sub-modules (SMs) in the MMC circuit and each SM is a potential failure point [3,4].

The research of fault detection and classification in MMC-HVDC systems applications can be broadly categorized into three basic approaches that are mechanism-based, signal processing-based, and artificial intelligence-based [5]. All the mechanism-based methods need many sensors monitoring the inner characteristics (circulating current, arm currents, capacitor voltages, etc.). Signal processing-based methods employ output characteristics rather than inner characteristics to detect a fault. Signal processing-based methods have been deemed reliable and fast by researchers [6-9] with the advancement of signal processing methods in recent years. But both of them need suitable methods to obtain expected inner characteristics or threshold of certain derived features, such as zero-crossing current slope or harmonic content which degrades the robustness of fault detection and classification. The artificial intelligent methods do not need any mathematical models of MMC functionality and any threshold setting, yet, they can improve the accuracy of fault diagnosis due to their advantage of nonlinear representations.

A neural network as the most basic artificial intelligence method is used by many researchers. Khomfoi and Tolbert [10] propose a fault diagnosis and reconfiguration technique for a cascaded H-bridge multilevel inverter drive using principal component analysis (PCA) and neural network (NN). In this method, the genetic algorithm is used to select valuable principal components. Simulation and experimental results showed that the proposed method is satisfactory to detect fault type, fault location, and reconfiguration. Wang et al. [11] propose an artificial NN-based robust DC fault protection algorithm for MMC high voltage direct current grid. In which, the discrete wavelet transform is used as an extractor of distinctive features at the input of the ANN. Furqan Asghar et al. [12] present NN-based fault detection and diagnosis system for three-phase inverter using several features extracted from the Clarke transformed output as an input of NNs. Merlin et al. [13] design thirteen artificial NNs for the voltage-source converter-HVDC systems to detect a fault condition in the whole HVDC system based only on voltage waveforms measured at the rectifier substation.

Although the NN based methods achieved some improvements in the diagnosis of failed converters and identification of defective switches [14,15], the prerequisite for the successful application of NNs is to have enough training data and long training time. Multi-class relevance vector machines (RVM) and support vector machine (SVM) replace a neural network to classify and locate the faults because of their rapid training speed and strongly regularized characteristic [5]. Wang et al. [16] use a PCA and multiclass RVM approach for cascaded H-bridge multilevel inverter system fault diagnosis. Wang et al. [17] propose and analyze a fault-diagnosis technique to identify shorted switches based on features generated through the wavelet transform of the converter output and subsequent classification in SVMs. The multi-class SVM is trained with multiple recordings of the output of each fault condition as well as the converter under normal operation. Jiao et al. [18] used the three-phase AC output side voltage of MMC as the fault characteristic signal, combined with PCA data preprocessing and firefly algorithm optimized SVM (FA-SVM) for MMC fault diagnosis. Zhang and Wang [19] proposes a least-squares-based ε-support vector regression scheme, which captures fault features via the Hilbert–Huang transform. Fault features are used as the inputs of ε-support vector regression to obtain fault distance. Then, the least-squares method is utilized to optimize the parameters of the model so that it can meet the demand on fault location for MMC–MTDC transmission lines.

To build the aforementioned artificial intelligence machine, feature extraction techniques such as Fourier analysis [20,21], wavelet transform [14,15], Clarke transform [12] or feature subset selection techniques such as Principal component analysis (PCA) [10,22] and multidimensional scaling (MDS) plays an important role. Sometimes to select suitable sub-features, the Genetic Algorithm (GA) [10,22,23] or particle swarm optimization (PSO) [24] are employed. It is well known that feature extraction has always been a bottleneck in the field of fault diagnosis. Moreover, the feature extraction and all the following post-operation increase the computation burden.

Deep learning methods have been explored to learn the features from the data which can be generalized to different cases. Zhu et al. [25] proposed Convolutional Neural Networks (CNN) for fault classification and fault location in AC transmission lines with back-to-back MMC-HVDC, in which, two convolutional layers were used to extract the complex features of the voltage and the current signals of only one terminal of transmission lines. Serkan Kiranyaz et al. [26] use 1-D CNN to detect and localize the switch open-circuit fault using four cell capacitor voltage, circulating current and load current signals. This method can achieve a detection probability of 0.989 and an average identification probability of 0.997 in less than 100ms. Qu et al. [27] propose CNN for MMC fault detection using each capacitor's voltage signal. Wang et al. [28] propose CNN for DC fault detection and classification using wavelet logarithmic energy entropy of transient current signal. In the past our research group proposed some related methods of NNs [29~31], AE-based DNN [32] and softmax classifier [33] for bearing fault detection and classification, but not for MMC-HVDC. Moreover, to the best of our knowledge, use of deep learning methods for MMC fault detection and classification have been very limited and there is no comparison of two deep learning methods. Furthermore, Aforementioned CNNs have achieved success, but their advantages have not been explored completely, e.g., the ability of feature extraction, the speed of processing, and its stability. In summary, up to now, there is still much room to further improve the performance of the open-circuit fault diagnosis of MMCs.

To shorten such a gap and achieve high fault classification accuracy with fewer sensors and reduced computational time for fault diagnosis of MMCs, we propose two deep learning methods and one stand-alone Softmax Classifier for MMCs faults detection and classification using raw data collected from current sensors to recognize automatically the open-circuit failures of IGBT in MMCs. The contributions of this paper are as follows:

a. Only current sensors data are used for fault diagnosis and achieved high accuracy of fault detection and classification.

b. Multichannel current signals are used instead of a single channel to improve reliability because the sensors may also cause some faults.

c. Excellent accuracy on fault detection and identification without data preprocessing or post-operation;

d. Two deep learning methods and a stand-alone Softmax Classifier are used with raw data collected by current sensors to achieve improved classification accuracy and reduced computation time.

e. Performance comparison of CNN, AE-based DNN, and Softmax Classifier in terms of fault diagnosis accuracy, stability and speed for MMC-HVDC fault diagnosis.

This paper is organized as follows. Section 2 introduces the topology and data acquisition from MMC. Section 3 proposes the framework of this paper and the design of CNN, AE-based DNN, and Softmax Classifier. The feasibility and performance of the proposed approaches are evaluated in Section 4. Section 5 compares the three deep learning methods. Conclusions are drawn in section 6.

2. MMC topology and data acquisition

The data for this study was simulated from a two-terminal model of the MMC-HVDC transmission power system using PSCAD/EMTDC [34]. It solves the differential equations of the entire power system and its controls. Figure 1 shows that each phase of the three-phase MMC consists of two arms (upper and lower) that are connected to two inductors L. Each arm contains a series of SMs, and each SM involves two IGBTs (i.e., T_1 and T_2), two diodes D, and a DC storage capacitor.

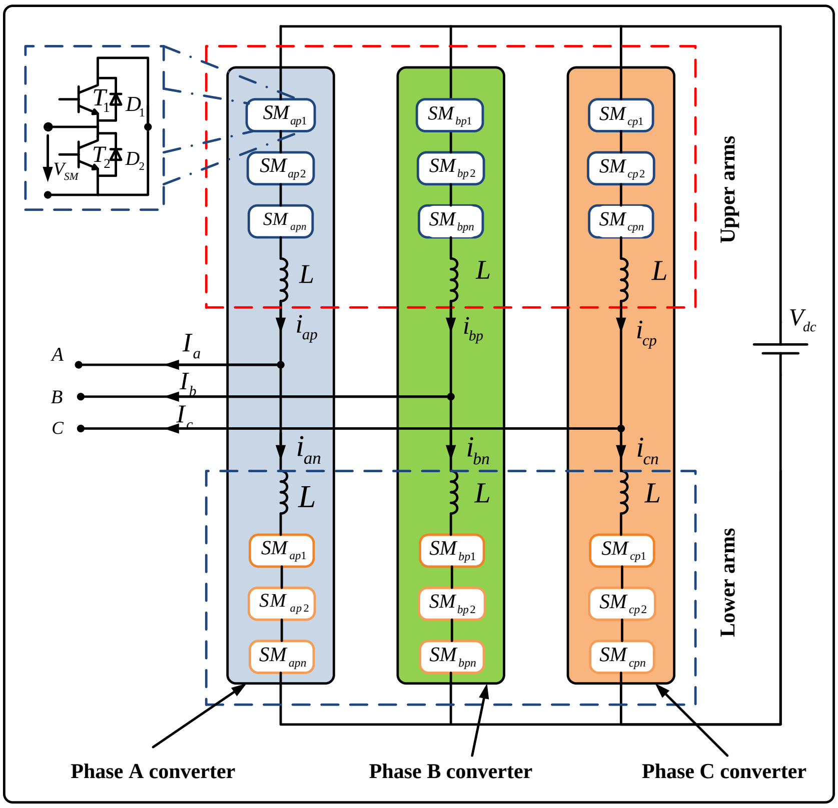


Figure 1. Structure of a three-phase MMC with half-bridge submodules

In our simulation (Table 1), we recorded 9 channels of data for normal and 6 different locations of IGBT break-circuit fault manually for each bridge (namely A-phase lower SMs, A-phase upper SMs, B-phase lower SMs, B-phase upper SMs, C-phase lower SMs, and C-phase upper SMs). There are 100 cases of IGBT break-circuit fault that happened at different IGBTs of the six bridges at different times. The power system is depicted in Figure 2. The type of SMs is half-bridge and the direction of the flow is shown as the arrow above. Ba-A1 and Ba-A2 are two AC bus bars. Bb-A1 and Bb-A2 are two DC bus bars. E1 is an equivalent voltage source for an AC network. E2 is a wind farm.

Table 1. Parameters of MMC.

Parameters	Value
number of SMs per arm	9
SM capacitor	1000uF
arm inductance	50mH
AC frequency	50Hz

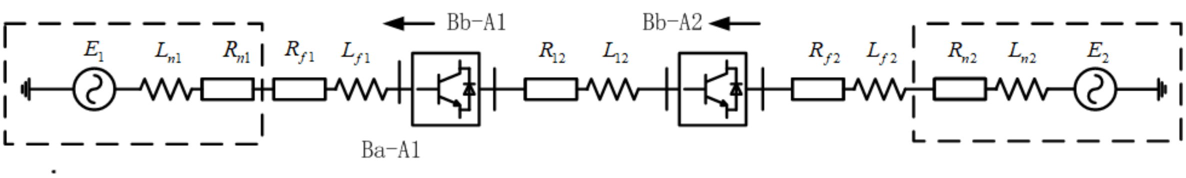


Figure 2. Structure of the HVDC

The whole time period used is 0.1s while the time for the IGBT open circuit fault duration is varied from 0.03s to 0.07s. The simulation time step is 2μs and the sampling frequency is 20μs. The acquired data channels for fault diagnosis are AC-side three-phase current (I_a, I_b, I_c) and three-phase circulation current (I_{diffa}, I_{diffb}, I_{diffc}).

3. The framework of fault classification and design of deep learning methods

3.1. The framework for fault detection and classification

This paper proposes three methods to complete both the fault detection and classification task for MMC, as shown in Figure 3, which are CNN, AE-based DNN, and a stand-alone Softmax classifier. CNN processes the raw sensors data which are nine current signals (I_a, I_b, I_c, i_{ap}, i_{bp}, i_{cp}, i_{an}, i_{bn}, and i_{cn}) and obtains the fault diagnosis results. AE-based DNN and Softmax process the combined information which is concatenated the measurements of these nine parameters to form a vector of samples that represent the current health condition of the MMCs, then obtain the fault diagnosis results.

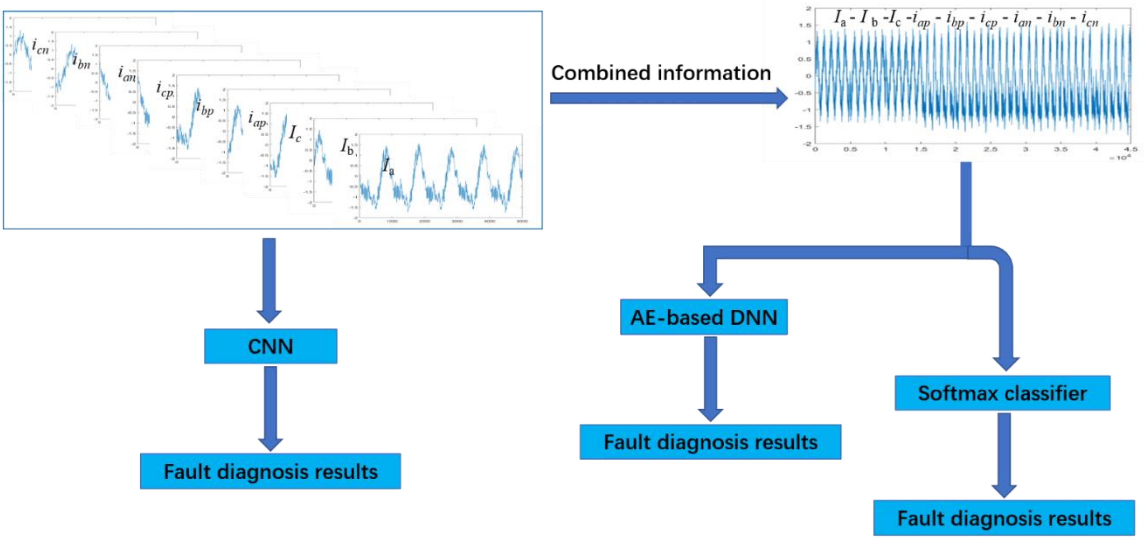


Figure 3. Framework for fault detection and classification for MMC

3.2. Design of CNN

Convolutional neural networks (CNNs) are widely used tools for deep learning which is different from the traditional feed-forward ANN because of its three architectural properties of the visual cortex cell: local receptive regions, shared weights, and subsampling. The crucial advantage of CNNs is that both feature extraction and classification operations are fused into a single Machine learning body to be jointly optimized to maximize the classification performances [26].

CNN consists of multiple layers such as figure 4 which are the input layer, convolutional layer, activation layer, pooling layer, full connect layer, softmax layer, and a classification layer. Among these layers, there are two basic layers in CNN which are the convolutional layer and the pooling layer. Convolution operation implements the first two properties that are local receptive regions and shared weights. The pooling operation implements the subsampling property [35].

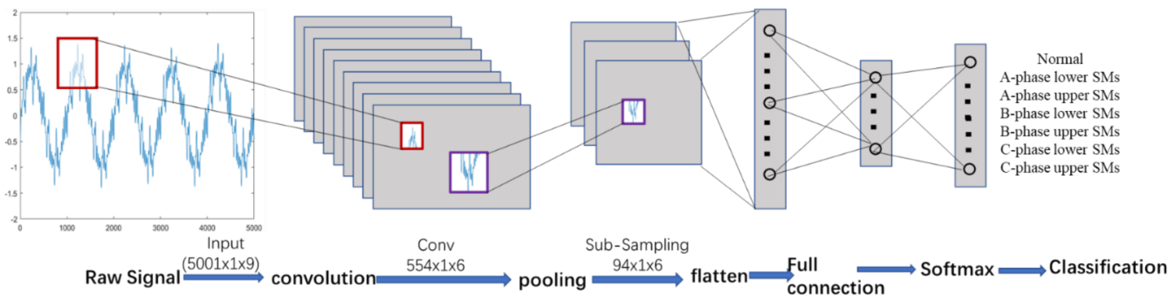


Figure 4. Architecture of the signal-level CNN classifier

A convolutional layer consists of neurons that connect to small regions of the input and operate the convolution computation. The output feature map of the convolutional layer can be written as:

$$F_j = \varphi(\sum_{i=1}^{N} W_{i,j} \otimes I_i + b_j), \tag{1}$$

For the jth filter, the output is a new feature map F_j, Where $W_{i,j}$ and b_j denote the jth filter kernel and bias, respectively. I_i is the input matrix of the ith channel, $\otimes$. represents the convolutional operation, and I_i is convoluted with a corresponding filter kernel $W_{i,j}$. The sum of all convoluted matrices is then obtained and a bias term b_j is added to each element of the resulting matrix. There are many several choices we could make activation function φ be a non-linear. But in this paper, we simply use a named leaky rectified linear unit (leaky ReLU). The function of leaky ReLU is given by:

$$\varphi(x) = \begin{cases} x, & x \geq 0 \\ scale * x, & x < 0 \end{cases} \tag{2}$$

It is a simple threshold that makes the negative value be zero. Then we can obtain the output feature map F_j.

Pooling layers perform down-sampling operations. Pooling methods usually include max-pooling and average-pooling. In this paper, the average-pooling function is applied which outputs the average values of rectangular regions of its input. In a fully connected layer, neurons between two adjacent layers are fully pairwise connected but neurons within the same layer share no connections. Then the Softmax function is commonly adopted for classification tasks. The introduction of Softmax will be presented in the following subsection 3.4.

3.3. Design of AE-based DNN

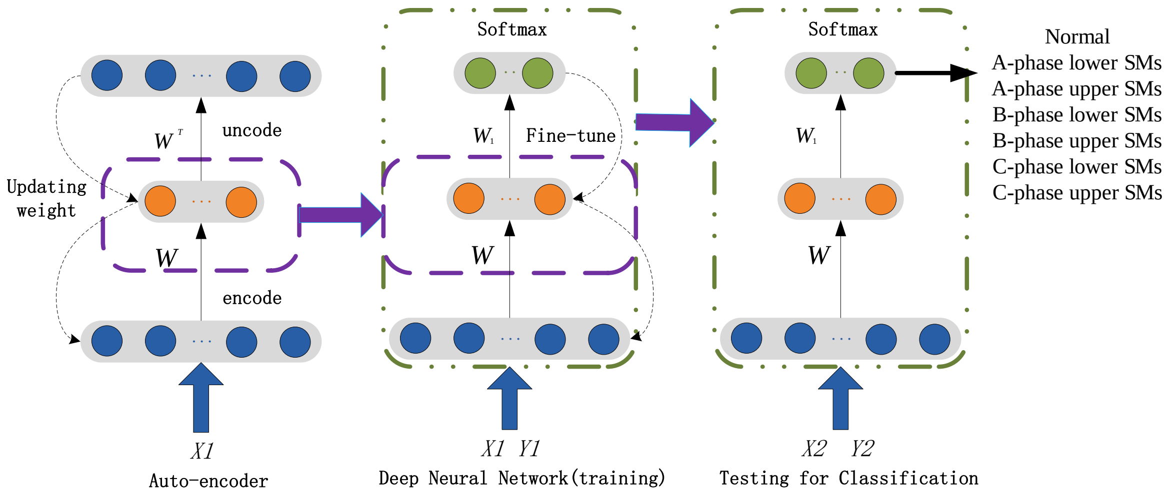


Figure 5. Architecture of the AE-based DNN

An AE-based DNN (Deep Neural Network) is constructed by several autoencoders (AEs) stacked with each other and a Softmax classifier on the output layer. In this paper, we stacked one AE with a Softmax classifier as can be seen in figure 5. The AE needs to be pretrained by Greedy layer-wise training algorithm. The simplest form of an AE includes three layers: the input layer, hidden layer, and output layer. An AE network consists of an encoder and a decoder. The encoder maps the input to a hidden representation and the decoder attempts to map this representation back to the original input. given an unlabeled vector sample x, The encoder network can be explicitly defined as:

$$h = f(w_1 x + b_1), \tag{3}$$

Similarly, the decoder network can be defined as:

$$\hat{x} = g(w_2 x + b_2), \tag{4}$$

where $\hat{\mathbf{x}}$ are the approximate reconstruction of the inputs, and $\theta = \{\mathrm{w}, b\}$ are the reconstructing parameters, and f and g are the activation function of the encoder and decoder, respectively. The reconstruction error E between the inputs x and output $\hat{\mathbf{x}}$ are defined as:

$$E = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}_{mean\ squared\ error} + \lambda * \underbrace{\underbrace{\Omega_{weights}}_{L_2}}_{regularization} \tag{5}$$

Where the first part is the mean square variance used to measure the average discrepancy and N is the number of neurons in the output layer, and the second part is the regularization term used to prevent overfitting. λ is the coefficient for the L_2 regularization term.

$$\Omega_{weights} = \frac{1}{2}\sum_{l}^{L}\sum_{j}^{N}(w_j^{(l)})^2 \tag{6}$$

Where L is the number of hidden layers. The following subsection introduces the softmax classifier.

3.4. Introduction of Softmax classifier

The Softmax function, also known as softargmax or normalized exponential function, is a function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. It is calculated as:

$$y_r(x) = P(c_r|x,\theta) = \frac{exp(a_r(x))}{\sum_{j=1}^{k} exp(a_j(x))}, \tag{7}$$

The loss function can use mean squared error function and the cross-entropy function. In this paper, we used the cross-entropy function which is given by:

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{k} t_{ij}\, ln y_{ij}, \tag{8}$$

where t_{ij} is the indicator that the ith example belongs to the jth class, y_{ij} is the output for example i, which here is the value from the softmax function.

4. Experimental study

Seven conditions of MMCs status have been recorded which include normal, A-phase lower SMs, A-phase upper SMs, B-phase lower SMs, B-phase upper SMs, C-phase lower SMs, and C-phase upper SMs faults. 100 examples were collected from each condition. So there are a total of 700 (100 x 7) raw data files to process with. All the nine parameters, i.e., I_a, I_b, I_c, i_{ap}, i_{bp}, i_{cp}, i_{an}, i_{bn}, and i_{cn}, were recorded to obtain 5001-time samples.

Experiments were conducted for testing data rates from 0.1 to 0.9 and 20 run times for each testing data rate. We need to point out that the detection and classification results in the following paper are the average of 20 run results. In order not to be influenced by the difference in data used, it is important to ensure that these methods work with the same data at each run. The following code is pseudo-code which can explain this scenario.

```
For TestingDataRate=0.1:0.1:0.9
    For i=1:20
        [trainData testData]=split(RawData,TestingDataRate);
        CNN=trainCNN (trainData);
        ResultsCNN=CNN(testData);
        [trainDataCI testDataCI]=combined Information( trainData, testData);
        AE-basedDNN=trainAE-basedDNN(trainDataCI);
         ResultsAE= AE-basedDNN(testDataCI);
        Softmax=trainSoftmax(trainDataCI);
         ResultsSoftmax =Softmax(testDataCI)
    End
```

End

4.1. Implementation details and results of CNN

4.1.1. Implementation details of CNN

Figure 4 illustrates the architecture of CNN for fault detection and classification. The input data is the raw sensor signals. Each channel denotes one sensor which records 5001-time samples. So the size of input current signals is [5001x1x9], where the length is 5001 and the height is 1 as the signals are one dimensional, and the depth is 9 as the signals come from 9 channels. The input is convolved with 6 filters of size [30 1] with stride 9 and padding 3, then applied a leaky ReLU function, in which the scalar multiplier for negative inputs is set as 0.01, resulting in a new feature map of size 554x1 and 6 channels. The sequence is pooling operation which is applied to each feature map separately. Our pooling size is set 6x1 and stride is 6. Therefore, a convolution feature map is divided into several disjoint patches and then the average value in each patch is selected to represent the patch and transmit to the pooling layer, then the feature map is reduced to 94x1 by the pooling operation.

As stochastic gradient descent with momentum (SGDM) algorithm may reduce the oscillations along the path of the steepest descent towards the optimum that is sometimes caused by stochastic gradient descent algorithm [36], we use the SGDM algorithm to update the parameters of the deep NN. The momentum is set at 0.95, the learning rate is 0.01 and the maximum number of epochs to use for training is set at 30.

4.1.2. Results of CNN

The accuracy of the CNN fault detection is shown in Table 2. For fault detection, the output network is divided into two types: fault and normal. We can see from Table 2 when the testing rate is 0.1~0.5 and 0.7, the detection accuracy is 100%. The min of the detection accuracy is 99.7% at the testing rate of 0.9. In which, there are 0.3% fault cases are misclassified as normal cases.

Table 2 Fault detection accuracy of CNN

Testing data rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Detection accuracy(%)	100	100	100	100	100	99.9	100	99.8	99. 7

The classification results of training and testing data using convolutional NNs are shown in figure 6. From the viewpoint of trending, we can see that with the testing data rate increases, both classification accuracy for training data and testing data decline. For the training dataset, the standard deviation of classification accuracy increases with the increase of the testing data rate. For testing data set, the max of mean accuracy is 98.6% with testing data rate 0.1 and the min of the average accuracy is 93.0% with testing data rate 0.9. The standard deviation of classification accuracy in the middle of the testing data rate is smaller than both ends of the testing data rate. Moreover, for each testing data rate, the standard deviation of classification accuracy for the training data set is less than the standard deviation of classification accuracy for the testing data set.

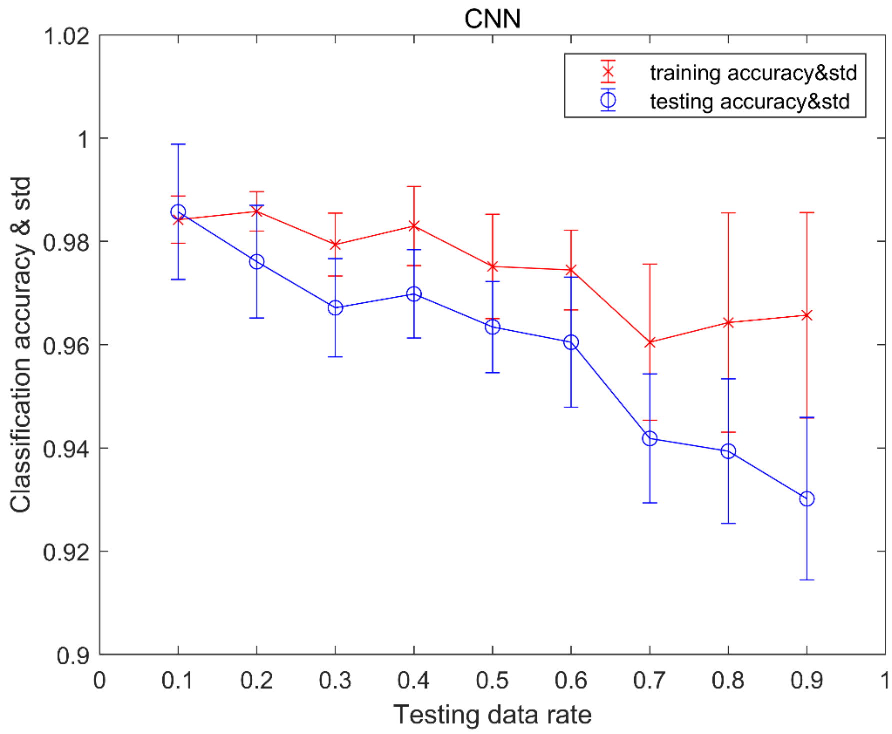


Figure 6. The classification accuracy and the standard deviation of CNN

Table.3 provides a confusion matrix of the classification results for each condition with testing data rates of 0.2, 0.5 and 0.8. As can be seen from Table.3 that the recognition of the normal condition of the MMCs is 100% with 0.2, 0.5, and 0.8 testing data rates. With 0.2 testing data rate, our method misclassified 3.2% of testing examples of condition 4 as condition 2 and 2% of testing examples of condition 4 as condition 6; With 0.5 testing data rate, our method misclassified 1.6% of testing examples of condition 4 as condition 2 and 3.4% of testing examples of condition 4 as condition 6; Furthermore, with 0.8 testing data rate, our method misclassified 0.8% of testing examples of condition 4 as condition 2 and 6.4% of testing examples of condition 4 as condition 6.

Table.3 Sample confusion matrix of the classification results of CNN

Testing data rate=0.2							Testing data rate=0.5							Testing data rate=0.8						
100	0	0	0	0	0	0	100	0	0	0	0	0	0	100	0.2	0	0	0	0.5	1
0	97.8	0	3.2	0	0	0	0	95	0	1.6	0.2	0.7	1.3	0	91.6	0	0.8	0	0.9	2.3
0	0	97.3	0	0	0	0.8	0	0	97.2	0	0.9	0	1.1	0	0	94.4	0	2.5	0	2.2
0	0.7	0	94.8	0	2.2	0	0	1	0	95	0	1.9	0	0	3.8	0.4	92.8	0	0.6	0.2
0	0	2.2	0	99.8	0	3.2	0	0	1.9	0	96.1	0	3	0	0	3.2	0	90.8	0	2.5
0	0.7	0	2	0	97.8	0	0	3.6	0.2	3.4	0.3	96.9	0.4	0	4	0.3	6.4	0.6	97.1	0.9
0	0.2	0.5	0	0.2	0	96	0	0.4	0.7	0	2.5	0.5	94.2	0	0.4	1.7	0	6.1	0.9	90.9

4.2. Implementation details and results of AE-based DNN

4.2.1. Implementation details of AE-based DNN

First, the measurements of nine current signals were concatenated to form a vector of samples that represent the current health condition of the MMCs. This gave a total of 45009 (5001 x 9) samples dimension for each vector of health condition. Second, we used the AE with three layers: the input layer, hidden layer, and output layer. In which, the number of neurons in the hidden layer is set as 250 which means the sample dimension will be reduced from 45009 to 250. An AE network consists of an encoder and a decoder. The transfer function for the encoder and the decoder is the Satlin function and the logistic sigmoid function, respectively. Satlin function is a positive saturating linear transfer function given as:

$$f(z) = \begin{cases} 0, & if\ z \leq 0 \\ z, & if\ 0 < z < 1, \\ 1, & if\ z \geq 1 \end{cases} \quad (9)$$

The algorithm to use for training the autoencoder applied scaled conjugate gradient descent (SCGD). The maximum number of training epochs for this autoencoder is set as 10. Third, the 250 features achieved by trained AE are used as the input of the Softmax classifier. The maximum number of training epochs for the Softmax classifier is set as 20. Next, we stacked the trained AE and Softmax classifier into a deep NN. Finally, we trained this deep NN using the training data.

4.2.2. Results of AE-based DNN

The fault detection results of the AE -based DNN are shown in table 4. When the testing rate varies from 0.1 to 0.7, the detection accuracy is 100%. The lowest detection accuracy is 99.7% at the testing rate of 0.9. In which, there are 0.3% fault cases are misclassified as normal cases. Compared with Table 2 of CNN, AE-based DNN has better detection accuracy.

Table 4. Detection accuracy of AE -based DNN

Testing data rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Detection accuracy	100	100	100	100	100	100	100	99.9	99.7

Figure 7 shows the classification results of training and testing data using AE-based DNN. From the viewpoint of trending analysis, we can see that with the testing data rate increase, the classification mean accuracy for testing data declines but the classification accuracy for training data increases. For the training data set, the highest average accuracy is 99.5% with testing data rate 0.8 and the lowest is 98.6% with testing data rate 0.1. The standard deviation of classification accuracy increases with the increase of the testing data rate. For the testing data set, the max of mean accuracy is 97.6% with testing data rate 0.1 and the min of mean accuracy is 92.1% with testing data rate 0.9. The standard deviation of classification accuracy in the middle of the testing data rate is smaller than both ends of the testing data rate. We also can see that for each testing data rate the standard deviation of classification accuracy for the training data set is less than the standard deviation of classification accuracy for the testing data set.

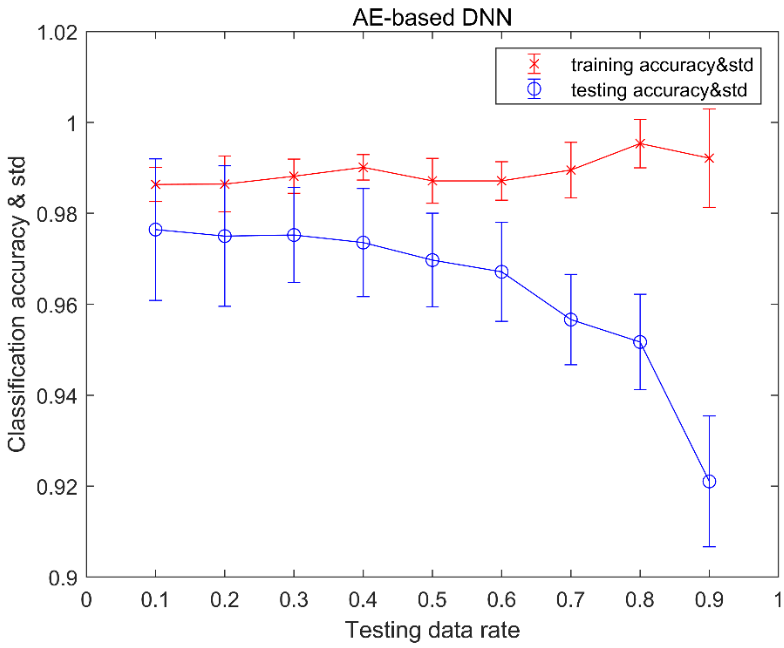


Figure 7. The classification accuracy and the standard deviation of AE-based DNN

Table 5 provides a confusion matrix of the classification results for each condition with testing data rates of 0.2, 0.5and 0.8. As can be seen from Table 5 that the recognition of the normal condition of the MMCs is 100% with 0.2, 0.5, and 0.8 testing data rates. With 0.2 testing data rate, our method misclassified 1.5% of testing examples of condition 3 as condition 5; With 0.5 testing data rate, our method misclassified 1.8% of testing examples of condition 3 as condition 5 and 0.2% of testing examples of condition 3 as condition 7. With 0.8 testing data rate, our method misclassified 0.7% of testing examples of condition 3 as condition 4, 1.6% of testing examples of condition 3 as condition 5, 1% of testing examples of condition 3 as condition 6 and 1.9% of testing examples of condition 3 as condition 7.

Table 5. Sample confusion matrix of the classification results of AE-based DNN

Testing data rate=0.2							Testing data rate=0.5							Testing data rate=0.8						
100	0	0	0	0	0	0	100	0	0	0	0	0	0	100	0.1	0	0	0	0.2	0.4
0	97	0	3.2	0	1.5	0.3	0	96.3	0	2	0.3	0.8	1.3	0	96.1	0	1.4	0.7	0.8	2.4
0	0	98.5	0	0	0	0	0	0	98	0	0.4	0	0.5	0	0	94.8	0	2.1	0.1	1.8
0	0.7	0	95.5	0	1	0.2	0	1.5	0	97	0	1.8	0.1	0	2.5	0.7	96.1	0	2	1.4
0	0	1.5	0	97	0	2.5	0	0	1.8	0	97.2	0	3.8	0	0.1	1.6	0	92.6	0	2.7
0	2.3	0	1.3	0.5	97.5	0	0	1.5	0	1	1.2	96	0	0	0.6	1	2.5	1.2	96.4	1
0	0	0	0	2.5	0	97	0	0.7	0.2	0	0.9	1.4	94.3	0	0.6	1.9	0	3.4	0.5	90.3

4.3. Results of Softmax classifier

The accuracy of Softmax Classifier fault detection is shown in Table 6. The detection accuracy is 100% at all testing rates.

Table 6. Detection accuracy of AE -based DNN

Testing data rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Detection accuracy	100	100	100	100	100	100	100	100	100

Figure 8 shows the classification results of training and testing data using the Softmax classifier. From the trending view, we can see that with the testing data rate increases, the classification average accuracy for testing data declines but the classification average accuracy for training data keeps steady which is 100%. The standard deviation of classification accuracy in the middle of the testing data rate is smaller than both end of testing data rate for testing data set but the standard deviation of classification accuracy keeps steady which is 0. For testing data set, the highest average accuracy is 99.46% with testing data rate of 0.2 and the lowest average accuracy is 93.52% with testing data rate of 0.9. It is obvious to see that for each testing data rate the standard deviation of classification accuracy for training data set is less than the standard deviation of classification accuracy for testing data set.

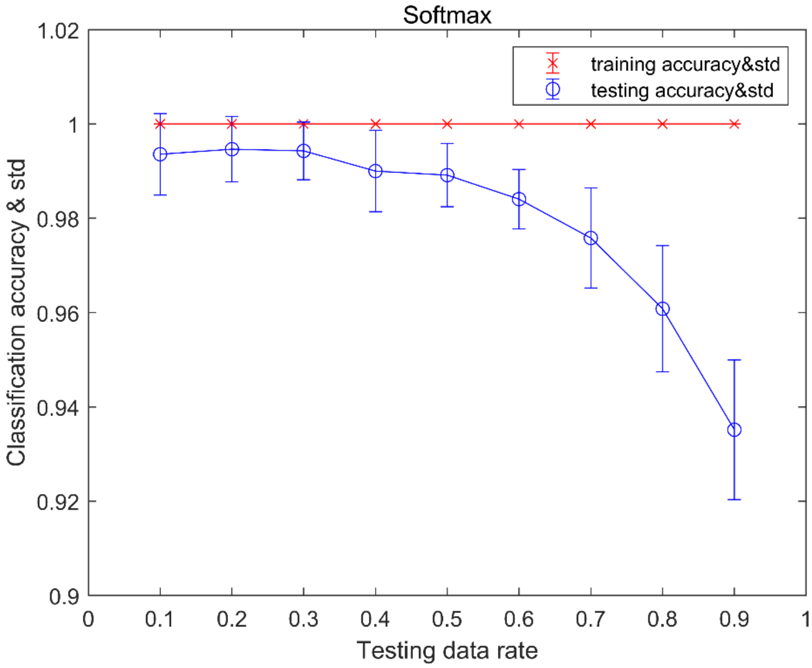


Figure 8. The classification accuracy and the standard deviation of the stand-alone Softmax classifier

Table 7 provides a confusion matrix of the classification results for each condition with testing data rates of 0.2, 0.5 and 0.8. As can be seen from Table 7 that the recognition of the normal condition of the MMCs is 100% with 0.2, 0.5, and 0.8 testing data rate. With 0.2 testing data rate, our method misclassified none of the testing examples of condition 4; With 0.5 testing data rate, our method misclassified 0.4% of testing examples of condition 4 as condition 2. With 0.8 testing data rate, our method misclassified 1.5% of testing examples of condition 4 as condition 2 and 1.88% of testing examples of condition 4 as condition 6.

Table 7. Sample confusion matrix of the classification results of the stand-alone Softmax classifier

Testing data rate=0.2								**Testing data rate=0.5**								**Testing data rate=0.8**						
100	0	0	0	0	0	0		100	0	0	0	0	0	0		100	0	0	0	0	0	0
0	98.5	0	0	0	0.5	0.5		0	98.1	0	0.4	0.2	0.8	0.4		0	97.1	0	1.5	0.4	1.4	3.7
0	0	100	0	0.2	0	0		0	0	99.4	0	0.7	0	0.2		0	0	95.6	0	1.3	0	0.5
0	0	0	100	0	0.8	0		0	0.7	0	99.6	0	0.6	0		0	2.2	0.3	96.6	0	2.3	0
0	0	0	0	99.8	0	0		0	0	0.6	0	99.1	0	0.6		0	0	1.5	0	94.3	0	2
0	1.5	0	0	0	98.5	0		0	0.6	0	0	0	97.4	0		0	0.5	0.3	1.9	0.8	95.9	0.7
0	0	0	0	0	0.2	99.5		0	0.6	0	0	0	1.2	98.8		0	0.2	2.3	0	3.2	0.4	93.1

Above all, for the training data set, with the increase of testing data rate, the average accuracy of Softmax keeps steady which is 100% and the average accuracy of CNN decreases but the average accuracy of AE-based increases. The standard deviation of accuracy for SoftMax keeps steady which is 0 and the standard deviation of accuracy for other methods increases with the increase of the testing data rate. For the testing data set, the average accuracy of all methods decreases with the increase of the testing data rate. And the standard deviation of accuracy in the middle is less than both ends of the testing data rate for all methods.

5. Comparisons

We compared the three methods on the classification accuracy and the standard deviation of classification accuracy for the testing data with the testing data rate from 0.1 to 0.9 and compared the three methods from the viewpoint of training time spent and testing time spent which are presented in Figure 9~11 respectively.

5.1. Comparison of average accuracy

From figure 9, we can see that the Softmax classifier behaves outstandingly on the testing data rate from 0.1 to 0.9 compared to CNN and AE-based DNN. When the testing data rate is 0.1, 0.2 and 0.9 which locates both ends, the classification accuracy of CNN is better than AE-based DNN.

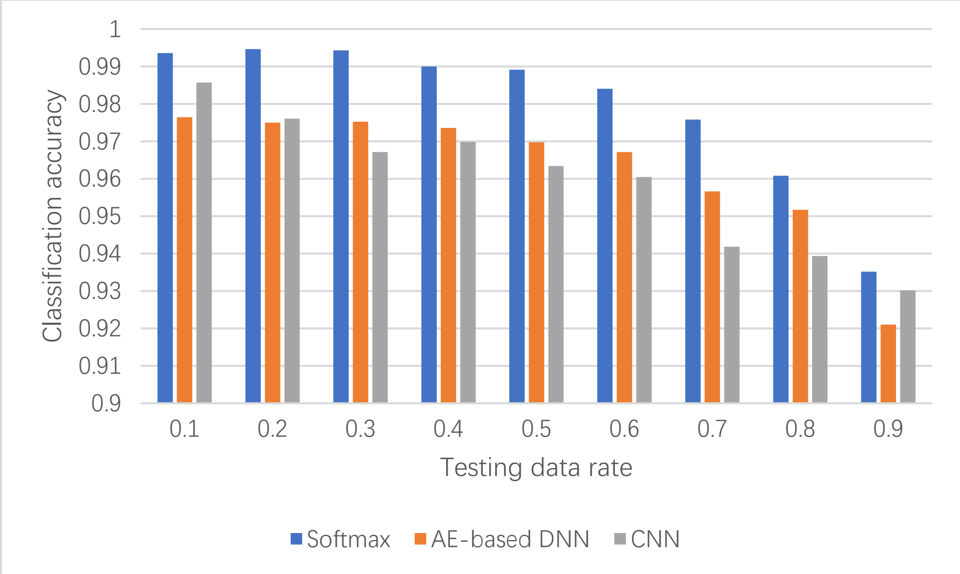


Figure 9. Comparison of classification accuracy for the three methods

5.2. Comparison of Standard deviation

We know that in statistics, the standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the values tend to be close to the expected value of set, while a high standard deviation indicates that the values are spread out over a wider range. From figure 10, it is clear that the standard deviation of accuracy of Softmax is lower than other methods when the testing data rate is 0.1 to 0.6, which means that for every run for different training data set and testing data set, the classification accuracy of Softmax is more stable and other methods are more spread out. When the testing data rate varies from 0.7 to 0.9, the AE-based DNN has the lowest standard deviation. AE-based DNN is the most spread out when the testing data rate is from 0.1 to 0.5 and CNN is the most spread out when the testing data rate is from 0.6 to 0.9.

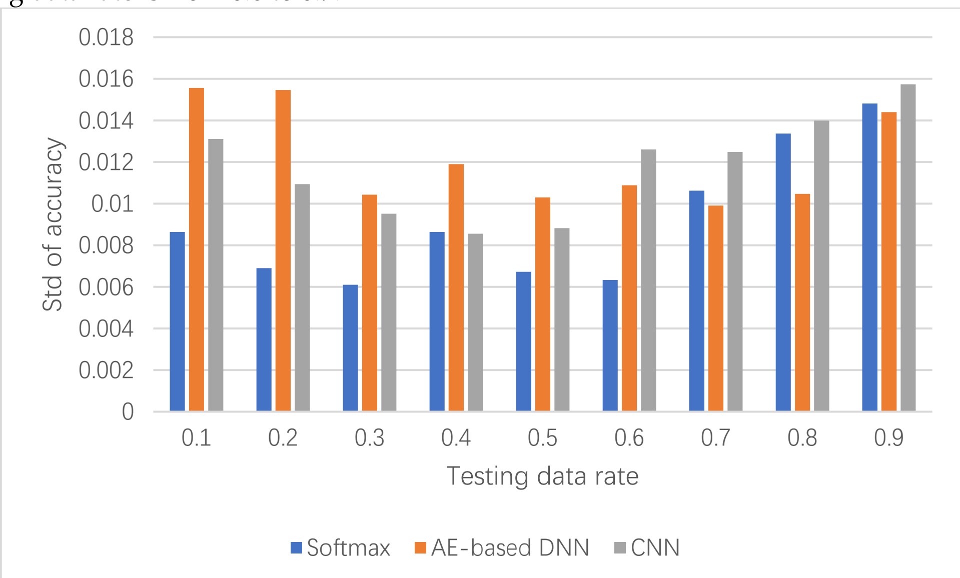
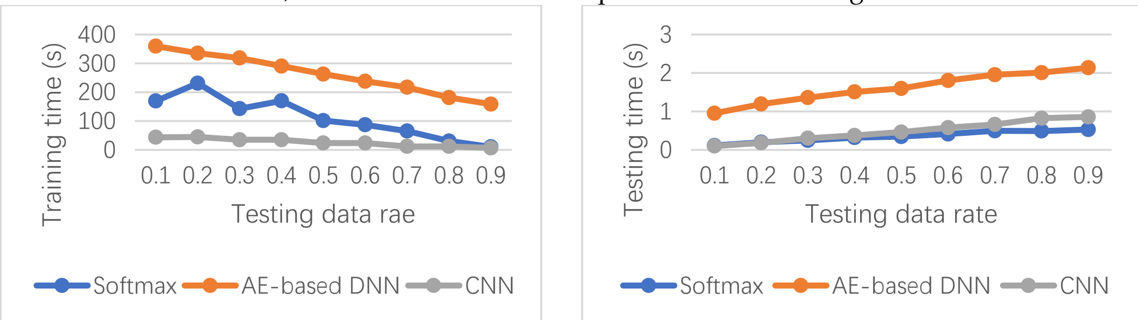


Figure 10. Comparison of the standard deviation of classification accuracy for the three methods

5.3. Speed Comparison

Figure 11 describes the training time and testing time spent by three methods. It shows that for each testing data rate, the AE-based DNN spends more training time than other methods, and the CNN spends the least training time, and the Softmax takes the least testing time when the testing data rate is from 0.3 to 0.9, and the AE-based DNN spends the most testing time.



(**a**) (**b**)

Figure 11. Comparison of speed for the three methods. (**a**) Comparison of training time spent by the three methods; (**b**) Comparison of testing time spent by the three method.

Over all, the stand-alone Softmax Classifier provides better functionality, including fault detection accuracy, classification accuracy, least standard deviation and speed, as well as its strong ability to dealing with high dimensional data. The AE-based DNN has the second best classification ability, but it needs more training time and testing time. CNN has enough classification accuracy and it needs the least training time.

6. Conclusions

Fault detection and classification are two of the challenging tasks in MMC-HVDC systems. This paper presented two deep learning methods (CNN and AE-based DNN) and a stand-alone Softmax classifier for fault detection and classification. CNN and AE-based DNN can fuse both feature extraction and classification operations into a single machine learning scheme for joint optimization to maximize the classification performance, which avoided the design of handcrafted features. In this paper, we only use raw current sensor data as input to our proposed approaches to detect and classify faults of MMC-HVDC. The simulation results in PSCAD/EMTDC show that three methods all have high detection accuracy more than 99.7%, in which the stand-alone Softmax classifier has 100% detection accuracy, and AE-based DNN is a little better than of CNN. Three methods also have high classification accuracy, small standard deviation, and high speed. Softmax classifier behaved better than others in classification accuracy and testing speed, while CNN needs the least training time.

Author Contributions: Qinghua Wang, Hosameldin O. A. Ahmed and Asoke K. Nandi conceived and designed this paper. Yuexiao Yu performed the experiments and generated the raw data, Mohamed Darwish gave the best suggestions about these experiments. Qinghua Wang analyzed the data and wrote a draft of the paper. All authors contributed to discussing the results and revising the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number No. 51105291, and the Shaanxi Provincial Science and Technology Agency, No. 2020GY-124.

Acknowledgments: This work is supported by Brunel University London (UK) as well as the National Fund for Study Abroad (China).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, C.; Li, Z.; Murphy, D.L.K., et al. Photovoltaic multilevel inverter with distributed maximum power point tracking and dynamic circuit reconfiguration. 2017 Int. Future Energy Electronics and ECCE Asia, Kaohsiung Taiwan, 2017; Publisher: IEEE; pp.1520-1525. (10.1109/IFEEC.2017.7992271)

2. Ghazanfari, A.; Mohamed, Y.A.-R.I. A resilient framework for fault-tolerant operation of modular multilevel converters. *IEEE Trans. Ind. Electron* 2016, Volume 63, (5), pp.2669-2678. (10.1109/TIE.2016.2516968)
3. Shahbazi M.; Poure P.; Saadate S.; et al. FPGA-based Fast Detection with Reduced sensor Count for a Fault-tolerant Three-Phase Converter. *IEEE Trans on Industrial Informatics* 2013, Volume 9(3), pp.1343-1350. (10.1109/TII.2012.2209665)
4. F Richardeau. T T L Pham. Reliability Calculation of Multilevel Converters: Theory and Applications. *IEEE Trans on Industrial Electron* 2013, Volume 60, (10), pp. 4225-4233. (10.1109/TIE.2012.2211315)
5. Wang C.; Zhou L.; Li Z. Survey of switch fault diagnosis for modular multilevel converter. *IET Circuits Devices & Systems* 2019, Volume 13, (2), pp. 117-124. (10.1049/iet-cds.2018.5136)
6. Nandi, R.; Panigrahi, B. K. Detection of fault in a hybrid power system using wavelet transform. Michael Faraday IET Int. Summit 2015, Kolkata, India, September 2015; Publisher: IET; pp. 203–206. (10.1049/cp.2015.1631)
7. Li, Y.; Shi, X.; Wang, F.; et al. Dc fault protection of multiterminal VSCHVDC system with hybrid dc circuit breaker. 2016 IEEE Energy Conversion Congress and Exposition (ECCE), Milwaukee, WI, September 2016; Publisher: IEEE; pp. 1–8. (10.1109/ECCE.2016.7854990)
8. Liu, L.; Popov, M.;Van Der Meijden, M.; et al. A wavelet transform-based protection scheme of multi-terminal HVDC system. 2016 IEEE Int. Conf. Power System Technology (POWERCON), Wollongong, NSW, Sept 2016; Publisher: IEEE; pp. 1–6. (10.1109/POWERCON.2016.8048434)
9. Costa, F. B. Boundary wavelet coefficients for real-time detection of transients induced by faults and power-quality disturbances. *IEEE Trans. Power Deliv* 2014, Volume 29, (6), pp. 2674–2687. (10.1109/TPWRD.2014.2321178)
10. Khomfoi, S.; Tolbert, L.M. Fault diagnosis and reconfiguration for multilevel inverter drive using AI-based techniques. *IEEE Transactions on Industrial Electronics* 2007, Volume 54, (6), pp.2954-2968. (10.1109/TIE.2007.906994)
11. Wang X.; Saizhao Y.; Jinyu W. ANN-based Robust DC Fault Protection Algorithm for MMC High-voltage Direct Current Grid. *IET Renewable Power Generation* 2020, Volume 14, (2), pp.199-210. (10.1049/iet-rpg.2019.0733)
12. Furqan A.; Muhammad T.; Sung H. K. Neural Network Based Fault Detection and Diagnosis System for Three-Phase Inverter in Variable Speed Drive with Induction Motor. *Journal of Control Science and Engineering* 2016, Volume 2016, pp.1-13.
13. Merlin, V. L.; Santos, R. C. dos; Le Blond, S.; et al. Efficient and robust ANN-based method for an improved protection of VSC-HVDC systems. *IET Renew Power Gener* 2018, Volume 12, (13), pp. 1555-1562. (10.1049/iet-rpg.2018.5097)
14. Liu H.; Loh, P. C.; Blaabjerg, F. Sub-Module Short Circuit Fault Diagnosis in Modular Multilevel Converter based on Wavelet Transform and Adaptive Neuro Fuzzy Inference System. *Electric Power Components and Systems* 2015, Volume 43, (8-10), pp. 1080-1088. (10.1080/15325008.2015.1022668)
15. Parimalasundar, E.; N. Suthanthira, V. Identification of Open-Switch and Short-Switch Failure of Multilevel Inverters through DWT and ANN Approach using LabVIEW. *Journal of Electrical Engineering & Technology* 2015, Volume 10, (6), pp. 2277-2287. (http://dx.doi.org/10.5370/JEET.2015.10.6.2277)
16. Wang, T.; Xu, H.; Han, J.; et al. Cascaded H-bridge multilevel inverter system fault diagnosis using a PCA and multiclass relevance vector machine approach. *IEEE Trans. Power Electron* 2015, Volume 30, (12), pp. 7006–7018. (10.1109/TPEL.2015.2393373)
17. Wang, C.; Lizana F, R.; Li, Z.; et al. Submodule short-circuit fault diagnosis based on wavelet transform and support vector machines for modular multilevel converter with parallel connectivity. 43rd Annual Conf. of the IEEE Industrial Electronics Society, Beijing, China, 2017; Publisher: Technical Committee on Control Theory, Chinese Association of Automation; pp. 3239–3244. (10.1109/IECON.2017.8216547)
18. Jiao W.; Liu Z.; Zhang Y. Fault Diagnosis of Modular Multilevel Converter with FA-SVM Algorithm. Proceedings of the 38th Chinese Control Conference. China, July, 2019; Publisher: IEEE; pp.5093 – 5098. (10.23919/ChiCC.2019.8866021)
19. Zhang M.; Wang H. Fault location for MMC–MTDC transmission lines based on least squares-support vector regression. *The Journal of Engineering* 2019,Volume 2019, (16), pp. 2125-2130. (10.1049/joe.2018.8640)

20. Khomfoi, S.; Tolbert, L.M. Fault diagnostic system for a multilevel inverter using a neural network. *IEEE Trans on Power Electron* 2007, Volume 22, (3), pp.1062–1069. (10.1109/TPEL.2007.897128)
21. Wang, T.; Xu, H.; Han, J.; et al. Cascaded H-bridge multilevel inverter system fault diagnosis using a PCA and multiclass relevance vector machine approach. *IEEE Trans. Power Electron* 2015, Volume 30, (12), pp. 7006–7018. (10.1109/TPEL.2015.2393373)
22. Khomfoi, S.; Tolbert, L. M. A diagnostic technique for multilevel inverters based on a genetic-algorithm to select a principal component neural network. Proc. IEEE Applied Power Electronics Conference and Exposition, Anaheim, CA, Feb,2007; Publisher: IEEE; pp. 1497–1503. (10.1109/APEX.2007.357715)
23. Zhang, Y..; Hu, H.; Liu, Z.; et al. Concurrent fault diagnosis of modular multilevel converter with Kalman filter and optimized support vector machine. *Systems Science & Control Engineering: Optimization and Control in Financial Engineering* 2019, Volume 7, (3), pp.43 – 53. (10.1080/21642583.2019.1650840)
24. Geometry, V.; Selvaperumal, S. Fault detection and classification with optimization techniques for three phase single inverter circuit. *Journal of Power Electronics* 2016, Volume 16, pp. 1–14. (https://doi.org/10.6113/JPE.2016.16.3.1097)
25. Zhu B.; Wang H.; Shi S.; et al. Fault location in AC transmission lines with back-to-back MMC-HVDC using ConvNets. *Journal of Engineering* 2019, pp.2430-2434. (10.1049/joe.2018.8706)
26. Kiranyaz S.; Gastli, A.; Ben-Brahim, L.; et al. Real-time fault detection and identification for MMC using 1-D Convolutional Neural Networks. *IEEE Transactions on Industrial Electronics* 2019, Volume 66, No.11, pp.8760-8771. (10.1109/TIE.2018.2833045)
27. Qu, X.; Duan, B.; Yin, Q.; et al. Deep convolution neural network based fault detection and identification for modular multilevel converters. 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, or USA, August, 2018; Publisher: IEEE; pp.1-5.
28. Wang, J.; Zheng, X.; Tai, N. DC fault detection and classification approach of MMC-HVDC based on convolutional neural network. 2018 2nd IEEE Conf. on Energy Internet and Energy System Integration; Publisher: IEEE; pp.1-6. (10.1109/EI2.2018.8582391)
29. Jack L.B; Nandi A. K. Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. *Mech. Syst. Signal Process* 2002, Volume 16, (2–3), pp. 373–390. (10.1006/mssp.2001.1454)
30. McCormick, A. C; Nandi, A. K. Real-time classification of rotating shaft loading conditions using artificial neural networks. *IEEE Trans. Neural Netw*1997, Volume 8, (3), pp. 748–757. (10.1109/72.572110)
31. Guo, H.; Jack, L.B.; Nandi, A. K. Feature generation using genetic programming with application to fault classification. *IEEE Trans. Syst., Man, Cybern, Part B(Cybernetics)* 2005, Volume 35, (1), pp. 89–99. (10.1109/TSMCB.2004.841426)
32. Ahmed, H. O. A.; Wong, M. L. D.; Nandi, A. K. Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features. *Mech. Syst. Signal Process* 2018, Volume 99, pp. 459–477. (10.1016/j.ymssp.2017.06.027)
33. Ahmed, H.; Nandi, A. K. Compressive Sampling and Feature Ranking Framework for Bearing Fault Classification With Vibration Signals. *IEEE Access* 2018, Volume 6, pp. 44731-44746. (10.1109/ACCESS.2018.2865116)
34. Yang, S.; Tang, Y.; Wang, P. Seamless fault-tolerant operation of a modular multilevel converter with switch open-circuit fault diagnosis in a distributed control architecture. *IEEE Transactions on Power Electronics* 2018, Volume 33, (8), pp.7058-7070. (10.1109/TPEL.2017.2756849)
35. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, Volume 521, (7553), pp. 436–444. (10.1038/nature14539)
36. Murphy, K. P. Machine Learning: A Probabilistic Perspective. The MIT Press, Cambridge, Massachusetts, 2012.