Multi-Label Classification for Power Quality Disturbances by Integrated Deep Learning

XIANGUI XIAO AND KAICHENG LI
School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
Corresponding author: Kaicheng Li (likaicheng@hust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52077089.

ABSTRACT
The traditional power quality disturbances classification methods include three stages, i.e., feature extraction, feature selection, classifier training. These methods suffer from low accuracy and a limited improvement margin. Since deep learning can greatly improve the accuracy of classification, a new classification method was designed in this paper by combining three types of deep learning frameworks, including CNN-GRU, ResNet-GRU, and Inception-GRU. The proposed method omits the two steps of feature extraction and feature selection, achieving “end-to-end” PQDs identification. To improve the performance on real signals, “pre-training and re-training” is applied. Then, a voting method was employed to vote the prediction labels by different algorithms, which further improves the accuracy of classification. Simulation experiments show that for the classification of compound PQDs, the proposed method performs better than the triple-stage methods and single deep learning classification method. Finally, real signals from Power source are test by the twice-trained model, and the five metrics are better than the old methods.

INDEX TERMS
Power quality disturbances, deep learning, CNN, ResNet, multi-label learning.

I. INTRODUCTION
With the gradual decline in manufacturing costs of wind turbines and the solar panels, the integration of new energy into the power grid has become inevitable, and the proportion of thermal power plants in power generation is gradually decreasing. All the advances in power technology have successfully reduced greenhouse gases to a certain extent. Meanwhile, the ratio of power electronic equipment (e.g., the charging pile for electric cars) in the power consumption side and other nonlinear loads connected with the power grid are also increasing. Despite these satisfactory technological advances, the occurrence of power quality disturbances (PQDs) is more frequent than ever before, which makes power quality improvement a challenging issue [1].

The PQDs classification has been always a significant problem. The traditional PQDs classification methods consist of three stages. The first stage is feature generation by time-frequency analysis technique; the second stage is feature selection by artificial intelligence, and the last stage is classifier training by machine learning [2]. The first stage mainly involves various mathematical analysis approaches.

The STFT [3] superimposed the signals in a fixed -length window and then obtained the time-frequency characteristics of non-stationary signals through FFT. However, its discrete form has no orthogonal expansion, so it is difficult to implement high efficient algorithms. The wavelet has an excellent analysis ability [4], [5], but it is sensitive to signal noise, which is harmful to the classification accuracy. Besides, the selection of wavelet basis function also depends on experience. Usually, the Daubechies series wavelets, such as wavelet db4, are selected. Empirical Mode Decomposition (EMD) [6] algorithm decomposes the PQDs signals into several IMF components to extract features from the signals. However, EMD lacks a rigorous mathematical basis, and the end-point effect and frequency aliasing cannot be ignored. S-transform (ST) [7]–[9] is the most commonly used for obtaining features from PQDs signals because of its superior time-frequency analysis capability and its insensitivity to noise. However, the effect of ST is affected by the parameter of the Gaussian window. To overcome this drawback, a small Gaussian window is selected in the low-frequency band, and a large Gaussian window is selected in the high-frequency band to achieve a higher resolution. Besides, Variation Mode Decomposition (VMD) is used to perform feature generation [10]. The advantage of VMD is its solid mathematical basis, but the decomposition level needs to be...
set artificially. Usually, the higher the decomposition level, the larger the calculation amount.

In the second stage of feature selection, a large number of artificial intelligence algorithms have been used to reduce the redundant features and thus the training time. The representative algorithms include particle swarm optimization algorithm [11], genetic algorithm [12], etc. However, these algorithms may converge to a local optimum, leading to relatively low accuracy.

The final stage is to train the classifier with the selected features. Classifiers mainly include artificial neural networks (ANN) [13], probabilistic neural networks (PNN) [14], expert systems [15], [16], support vector machines (SVM) [17], naive Bayes networks (NB) [18], [19], decision trees (DT) [20], [21], Random Forest (RF) [22], [23], etc. Zhou [24] pointed that the classification of PQDs can be treated as a multi-label classification task, which provides a novel approach for PQDs classification. In [25], K nearest Neighborhood (ML-KNN) was utilized as a multi-label classifier. The distance between the samples needs to be calculated for the original training set so that the labels of the test set can be predicted. The ML-KNN requires additional storage space to save the training set data, reducing the practicality of the algorithm. Liu [26] proposed a ranking support vector machine based on wavelet kernel function (Rank-WSVM) and used it for multi-label classification of over 40 kinds of compound PQDs. Although the wavelet kernel improved the classification accuracy, the training process is time-consuming.

In recent years, deep learning (DL) has been successful in computer vision and some other fields, a few studies have begun to apply DL to the classification of PQDs. The literature [27] applied 1D-CNN to classify PQDs, and the Dropout technique and early stopping were also adopted in 1D-CNN to prevent overfitting. In [27], the 1D-CNN contains as many as six layers. As a result, a large number of parameters need training, slowing down the training speed accordingly. He, in [28], designed residual network for image recognition. In [29], Gong proposed an Inception-ResNet architecture for PQDs classification, and further improved accuracy. Moreover, Sahani [30] combined deep CNN and on-line sequential random vector functional link networks (OSRVFLN) to improve the accuracy, but the architecture is rather complicated, which is not suitable for the real-time device.

Though the 1D-CNN has achieved some great results in PQDs classification, some problems still need to be resolved. For instance, CNN processes the 1D input signals by convolution and pooling operation, but it ignores the inner temporal relationship. Hence, CNN is not suitable for dealing with time series signals, such as the speech voice. To further improve the classification accuracy, Mohan [31] designed a novel algorithm that combines CNN and LSTM. In this algorithm, LSTM is added after the last layer of CNN, which do achieve improvement in the classification accuracy. However, the huge number of trainable parameters in LSTM results in a relatively long training time.

Another approach to achieve higher accuracy is to obtain more characteristic in PQDs signals. In [32], a new composite CNN architecture with two sub-modules was proposed. The input of the first sub-module is signal, i.e., 640 sampling points, and the input of the second one is the FFT results containing the frequency domain characteristics. Since the architecture utilized the time and frequency domain information, it achieved an accuracy of 99.86%, better than that of traditional CNN.

To overcome the defects of the aforementioned algorithms, the contributions of this paper are summarized as follows:
1) A multi-fusion CNN combing the time information and frequency domain information from FFT is designed. It consists of two sub-modules that take signal and FFT as input. After two convolution and pooling layers, the two tensors are concatenated into one layer. Thus, the parameters that need to be trained are much less than those of the classical deep CNN with six convolution layers.
2) The calculation amount of LSTM is huge due to the large number of parameters, to speed up the training process, it is replaced by Gate Recurrent Unit (GRU) to reduce the parameters amount and training time.
3) A new multi-label classifier is proposed. The classifier is an integrated DL method consisting of CNN-GRU, ResNet-GRU, and Inception-GRU networks, and these networks are trained on the training dataset firstly, and trained again on the real signals dataset, so as to improve the performance on real signals.
4) Finally, voting method is adopted to determine the final labels. For instance, if the predicted labels by CNN-GRU are not correct, while the ResNet-GRU and Inception-GRU predicted correctly, the voting method can ensure the labels are correct. Comparing with a single method, the simulation results indicate the ensemble architecture achieves better performance. Moreover, the method does not require feature generation by mathematical time-frequency analysis methods and feature selection by artificial intelligence methods, achieving an “end-to-end” PQDs identification.

The remaining parts are organized as follows. Section II explains the principle and architecture of the units in CNN-GRU, as well as the ResNet-GRU and Inception-GRU framework. In Section III, five evaluation metrics are given. In Section IV, the simulation results and comparisons are presented. In Section V, real signal test are performed by well-trained model, pre-training and re-training are used to improve the accuracy. Finally, the discussions and conclusions are provided in last section.

II. MODEL OF ALGORITHMS
A classic CNN architecture includes several units, and each unit consists of two types of layers that execute convolution and down-sampling operations, respectively. For computer
vision applications, the 2D convolutional layer is usually used. Although 1D signals are studied in this paper, the 1D convolution operation is similar to that of 2D signals, so the architecture is similar to 2D-CNN.

A. CONVOLUTION LAYER

The convolution layer distinguishes CNN from the traditional artificial neural networks. The convolution layer uses several convolution kernels (or filters) to extract the characteristics of the input 1D signals. The calculation formula of convolution is as follows:

\[
(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]
\]

where \(f[n]\) and \(g[m]\) are discrete signals.

Without loss of generality, suppose that the convolution kernel \(W\) is a filter with a size of \(3 \times 1\), i.e., \(W = [1,0,1]^T\), and the object signal \(f\) is a column vector with a size of \(7 \times 1\), i.e., \(f = [1,2,-1,-2,3,-3,0]^T\). First of all, the convolution kernel \(W\) is performed on signal \(f\), i.e., \(1 \times 1 + 0 \times 2 + 1 \times (-1) = 2\). Then, the convolution kernel \(W\) slides one step to the right to perform next convolution, i.e., \(1 \times 2 + 0 \times (-1) + (-1) \times (-2) = 4\). The convoluted results are obtained by repeating the above steps, as shown in Figure 1.

![FIGURE 1. Convolution layer.](image)

B. RELU LAYER

The activation function adds nonlinearity to CNN. The representative activation functions include sigmoid, tanh, and Rectified linear unit (ReLU). Among them, the ReLU function has a simple derivation, and it can effectively avoid the defect of gradient vanishing or gradient explosion in deep CNN. Therefore, ReLU has become a popular activation function in CNN, and its expression is:

\[
y = \max(x, 0) = \begin{cases} 
x, & x > 0 \\
0, & x \leq 0
\end{cases}
\]

the ReLU function keeps the input if its value is greater than or equal to 0, and sets it to 0 if the input is less than 0. Since it is easy to calculate, the calculation efficiency can be greatly improved. However, if filter \(W\) is negative and the input is positive, the corresponding positive output feature value will be deactivated. Similarly, the negative input feature value will be activated if filter \(W\) is positive. To overcome this weakness, several improved functions are proposed on the basis of ReLU, such as Leaky ReLU, Elus, and Paramatic ReLU (PReLU). Since the CNN with Leaky ReLU has limited performance, the CNN with ReLU is adopted in this study.

C. POOLING LAYER

The purpose of the pooling layer is to reduce dimensionality. It reduces the length of the signal array without discarding the original information of the original signal so that the number of neurons and the parameters is reduced. In general, the pooling operation includes maximum pooling and average pooling, which calculate the maximum value and the average value in the filter mapping area, respectively. In terms of PQDs identification, the maximum pooling is adopted in this study, and the calculation formula is as follows:

\[
y_l^{t+1} = \max_{(i-1)K+1 \leq j \leq K} (X_l^t)\]

where \(y_l^{t+1}\) is the \(t\)-th value of the \((l+1)\)-th layer’s output; \(K\) is the length of the pooling window; \(X_l^t\) is the \(t\)-th value of the input from the \(l\)-th layer. The computation process is shown in Figure 2.

![FIGURE 2. Pooling layer.](image)

In 2D-CNN, it is necessary to flatten the 2D tensor into a vector for the output layer. In this paper, “Dense” is used as the last layer.

D. BATCH NORMALIZATION LAYER

The Batch Normalization (BN) layer was proposed by Google in 2015. Specifically, this layer helps to make the outputs from the upper layer conform to Gaussian distribution. The formula of BN is given as follows:

\[
\hat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}
\]

\[
y_i = \gamma \hat{x_i} + \beta
\]
where, $\mu$ is the mean value of data $x_i$, $\sigma$ is the standard deviation, $\varepsilon$ is used to avoid a divisor of 0.

After the BN layers are integrated into CNN, the data from the upper layer is normalized. Compared to the CNN without BN layers, the CNN with BN layers can achieve much higher training efficiency. Also, its convergence speed is much faster than that of Stochastic Gradient Descent (SGD). Therefore, the BN layer has become a standard configuration in DL.

E. GATE RECURRENT UNIT

The literature [30] combines LSTM and CNN to achieve better classification results. Inspired by this work, this paper combines GRU with CNN, ResNet, and Inception to obtain three improved architectures for PQDs classification. The forward propagation of the GRU unit can be divided into four steps:

Step 1: the Update Gate calculation is performed, and the formula is as follows:

$$
    z_t = \sigma(W_c \cdot [h^{t-1}, X^t] + b_c)
$$

where $\sigma(\cdot)$ represents the Sigmoid function; $[h^{t-1}, X^t]$ represents the concatenation operation of $h^{t-1}$ and $X^t$; $h^{t-1}$ is the state passed from the previous cell to the current cell; $W_c$ and $b_c$ are the weight and the bias, respectively. Finally, the value is processed by the Sigmoid function, and a result between 0 and 1 is obtained.

Step 2: The Reset Gate calculation is performed, and the formula is as follows:

$$
    r_t = \sigma(W_r \cdot [h^{t-1}, X^t] + b_r)
$$

The calculation process of the Reset Gate is the same as that of Update Gate.

Step 3: the current state $\Tilde{h}$ is calculated, and the calculation formula is:

$$
    \Tilde{h} = \tanh(W_{\Tilde{h}} \cdot [r_t \odot h^{t-1}, X^t] + b_{\Tilde{h}})
$$

where “$\odot$” represents Hadamard product. Here, $r_t$ indicates a filtering operation on $h^{t-1}$. The value close to 0 in $r_t$ is multiplied by the element at the corresponding position of $h^{t-1}$. If the result is close to 0, the element is blocked and will not enter the next GRU unit. If $r_t$ is close to 1, the element in $h^{t-1}$ at the corresponding position will be retained and enter the next GRU unit. Finally, a value between −1 and 1 is obtained by the tanh function.

Step 4: the final output state $h^t$ of the current unit is calculated, and the calculation formula is:

$$
    h^t = (1 - z_t) \ast h^{t-1} + z_t \ast \Tilde{h}
$$
where $h^t$ represents the information to be passed to the next unit, and it is controlled by the update gate $z_t$. If the update gate is close to 1, there is more information from $h^t$; otherwise, there is more information from $h^{t-1}$. It can be seen from the calculation formula that the GRU can still retain the previous information after the information passes through a sufficiently long unit, so the gradient vanishing problem can be resolved.

After passing through several units consisting of several layers, it is generally regarded that the useful features of the signal have been fully extracted, and the function of the fully connected layer is to gather these characteristics and be prepared for output. If the task is a multi-classification problem, the output layer is Softmax. However, if the task is a multi-label classification, there will be several numbers close to “1” and close to “0” in the output vector at the same time, the output layer uses several “sigmoid” functions or “tanh” functions. In this article, each sample has 8 labels totally, so the output layer has a total of 8 sigmoid functions.

F. LOSS FUNCTION AND LEARNING RATE

The objective function is employed to compute the error between the true values and the predicted values, according to which the direction of parameters adjustment is decided. For multi-label problem, the objective function formula is expressed as follows:

$$cost = -\frac{1}{n} \sum_{i=1}^{n} \left[ L_i \ln v_i + (1 - L_i) \ln(1 - v_i) \right]$$  \hspace{1cm} (10)

where $v_i$ are the predicted vector by the trained model, $L_i$ are the true labels, $n$ denotes the instance numbers.

Learning rate (LR) is a key parameter in DL. If LR is too large, the convergence speed is fast during the early training process, but it may fluctuate during the late training process, and the training may fail to converge. If LR is too small, the training is bound to converge, at the cost of excessive training epochs. Therefore, an adaptive attenuated LR is used in this study, and the calculation formula is as follow:

$$lr = lr_{base} \times \gamma \left[ \frac{epochs}{epochsize} \right] + 1$$  \hspace{1cm} (11)

where $lr_{base}$ is the initial LR, which is set to 0.0001; $\gamma$ is 0.1, and epochsize is 10, indicating that the LR becomes one-tenth of the original one after 10 epochs. epochs stand for the total number of epochs, and it is 20 here.

The optimizers such as Momentum, AdaGrad, RMSprop, and Adam [26] are commonly used for optimization problems. Generally, all these optimizers perform well. However, the performance of AdaGrad and RMSprop may fluctuate during the optimization process, whereas Adam does not have this problem. Thus, the Adam optimizer is used in this study. Based on the above description, a novel 1D-CNN-GRU architecture for multi-label PQDs classification is proposed, as shown in Figure 3.

In Figure 3, the CNN-GRU architecture used for PQDs identification includes dual input channels. The first input is a tensor with shape of 1280 × 1, the first convolutional layer contains 16 convolution kernels, each convolution kernel size is 2, and the sliding step is 2, hence we get a tensor with shape of 640 × 16; then through the maximum pooling layer, and the tensor with shape of 320 × 16 is obtained. The second input is FFT results, we obtain a tensor with shape of 320 × 16. After the tensor are concatenated, it will pass through 16 GRU units, and finally we can obtain the predicted labels.

G. ResNet-GRU

The deep CNN suffers from the problem of gradient vanishing or gradient explosion in the training process. Generally, the more layers of CNN, the higher the accuracy. But as the number of layers continues to increase, the training effect tends to be worse. To address this issue, Kaiming He, in 2015, proposed a residual network (ResNet), which won the championship of ISLVRC that year [28].

In CNN, there is only one path for the training data to pass from the input to the output; whereas there are two paths in ResNet. The first path is from input to output through two or three convolutional layers; the second path, also called Shortcut, is directly connected from the input to the output through only one BN layer. According to whether there is a convolution module in the shortcut path, it can be divided into Convolution Block and Identity Block. The specific structure is shown in Figure 4.

A classic ResNet-GRU architecture generally consists of 50 layers, namely ResNet50. In total, there are 101 layers, even up to 152 layers. Generally, the more layers, the better the training effect. But in this case, the training time becomes quite long, and the improvement of the prediction accuracy on the test set is limited. Considering the training time, a ResNet-GRU architecture with 11-layers is proposed, which include 5 convolution layers, 5 BN layers, 16 GRU units, and the last output layer. The architecture is shown in figure 4. Since the convolution and BN module are similar to the ones in figure 4, the details are no longer explained.

H. INCEPTION-GRU

In 2015, Google proposed GoogLeNet, in which the Inception module was used. It is the application of this module that enables GoogLeNet to improve the accuracy of image classification to a level close to that of humans. The Inception module is relatively simple, and it is essentially a concatenation of several shallow convolution modules. In this paper, a Inception-GRU architecture was designed, as shown in Figure 5.

As can be seen from Figure 5, different from CNN-GRU and ResNet-GRU, the Inception structure can be regarded as the concatenation of several convolution modules. Because the shape of the convolution kernels of each branch is different, to ensure that the concatenation can be carried out, the number of convolution needs to be set to “SAME”. The optimizer and LR are still the same as that of 1D-CNN-GRU and ResNet-GRU. After the training process, the predicted
III. METRIC

Assuming $X \in \mathbb{R}^d$ is a $d$-dimensional sample space, $L = \{L_1, L_2, \ldots, L_Q\}$ is a label set composed of $Q$ labels, i.e., each label vector contains $Q$ labels. Assuming that there are now $n$ samples and label sets $D = \{(X_1, L_1), (X_2, L_2), \ldots, (X_n, L_n)\}$. The objective of multi-label is to generate classifier $h: X \rightarrow 2^L$ by using a ranking function $f: X \times L \rightarrow \mathbb{R}$ during the training process to meet the following conditions: for any $x_i \in X$, $L_1 \in L_i$, $L_2 \in L_i$, $L_1 \neq L_2$, the inequality $f(x_i, L_1) > f(x_i, L_2)$ should hold. In other words, the classifier should output a bigger number for the correct labels than the incorrect ones. For multi-label classification tasks, five evaluation metrics, which are listed from Equation (12) to (17), are employed to evaluate the classifier performance. The meaning and calculation formula of these five evaluation metrics are as follows:

Hamming loss. The purpose of this metric is to evaluate the proportion of the total amount of wrong and missing labels to the whole ones. The formula is listed as follows:

$$HL(h) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{Q} |h(x_i) \Delta L_i|$$  \hspace{1cm} (12)

where $n$ stand for the samples amount, and $Q$ stand for the whole labels amount. For example, the true labels of a sample are (0, 1, 1, 0, 0), and the labels predicted by the classifier are (1, 1, 1, 0, 0), there are 3 error labels in total, then Hamming loss is $3/5 = 0.6$.

Ranking loss refers to the average error of the sample label order.

$$RL(f) = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i}{|L_i||\bar{L}_i|}$$  \hspace{1cm} (13)

where

$$D_i = \left| \{(L_1, L_2)|f(x_i, L_1) \leq f(x_i, L_2), (Y_1, Y_2 \in L_i \times \bar{L}_i)\} \right|$$  \hspace{1cm} (14)

where $\bar{L}_i$ is the complement of set $L_i$. 

labels are decide by the voting method, or the three vote two wins principle.
Coverage refers to the average number of labels that need to be moved down to cover all labels corresponding to the sample.

\[
CR(f) = \frac{1}{n} \sum_{i=1}^{n} \max_{L \in L_i} \text{rank}_{f}(x_i, L) - 1
\]  

One error represents the probability that the top label in the ranking sequence of multi-label prediction results is not the correct label.

\[
OR(f) = \frac{1}{n} \sum_{i=1}^{n} \left\{ f_{\text{arg} \max}(x_i, L_j \notin L_i), L_j \in L \right\}
\]  

where if “·” is true, then |·| = 1, otherwise, there is |·| = 0. Here, \( n \) is the samples amount, \( f \) is the ranking function, \( x_i \) is the \( i \)-th sample, \( L_j \) and \( L \) has the same meaning as before.

Average precision this metric can be employed to compute the average proportion of the predicted value ranked over a certain label in the label set.

\[
AP(f) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|L_i|} \sum_{L \in L_i} \left| \frac{\sum_{L' \in L} \text{rank}_{f}(x_i, L') \leq \text{rank}_{f}(x_i, L), L' \notin L_i}{\text{rank}_{f}(x_i, L)} \right|
\]

Among the above five metrics, the larger the Average precision, the better the performance. Conversely, the smaller the other four metrics, the better the classifier.

**IV. SIMULATION**

In order to verify the effectiveness of the algorithm, 24 types of PQDs samples are generated in MATLAB, including 9 types of single PQDs, 8 types of dual PQDs, and 7 types of triple PQDs. The quadruple compound disturbance signals are not considered in this paper because of the low probability of occurrence in the real situation. The signal model and parameters conforms to IEEE 1159 standard [33].

The disturbance parameters are randomly generated, the sampling frequency is set to be 6.4KHz, i.e., 128 sampling points per cycle. The number of samples in each type is 1000, 800 of which are used as training set, 100 of which as validation set, and the rest of 100 samples are used to test the trained model. Since there are 24 types of samples totally, so the total number of samples is 24000.

After the signals are generated, ten-Cross-validation technique is utilized to get the mean value of the metrics. Moreover, to validate the robustness of the proposed CNN-ResNet-Inception framework, the 20dB, 30dB, and 40dB disturbances are considered.
40dB Gaussian white noise are superimposed to the pure PQDs, and then five metrics, as depicted from Equation (12) to (17), are calculated under various signal noise ratio (SNR) level to check the robustness of the framework.

### A. COMPARISON WITH SINGLE DL METHOD

In this paper, the three sorts of network, i.e., CNN-GRU, ResNet-GRU, and Inception-GRU are integrated, and finally a voting method is used to determine the output of the classifier. Therefore, the three ensemble methods are theoretically better than a single deep learning method in terms of accuracy. The statistics of five evaluation metrics of the algorithm in this paper and that of a single DL algorithm are given in Table 1.

Another worthy to be mentioned is the symbol ‘↓’ in Table 1 indicates that the smaller the metrics, the better the performance of the algorithm. In contrast, the meaning of symbol ‘↑’ is the larger the metrics, the better the performance of the algorithm.

| SNR   | Proposed  | CNN-GRU  | ResNet-GRU | Inception-GRU | CNN   | ST+MLRBF  | ST+Rank-WSVM |
|-------|-----------|----------|------------|---------------|-------|-----------|--------------|
| 40dB  | 5.942E-4 ± | 7.425E-4 ± | 7.425E-4 ± | 7.500E-4 ± | 0.0017 ± | 0.1966 ±  | 0.2056 ±     |
|       | 1.8593E-4 ± | 1.9867E-4 ± | 2.2551E-4 ± | 2.9125E-4 ± | 5.1562E-4 | 0.0051 ±  | 0.0047 ±     |
|       | 0.0009 ±   | 0.0013 ±   | 0.0021 ±   | 0.0046 ±   | 0.0043 ±   | 0.1990 ±   | 0.2050 ±     |
| 30dB  | 4.7896E-4 ± | 6.6786E-5 ± | 4.9707E-4 ± | 9.0666 ±   | 7.1997E-4 | 0.0037 ±   | 0.0013 ±     |
|       | 0.0047 ±   | 0.0052 ±   | 0.0083 ±   | 0.0075 ±   | 0.0120 ±   | 0.2042 ±   | 0.2067 ±     |
| 20dB  | 5.4872E-5 ± | 1.7352E-5 ± | 5.9758E-4 ± | 0.0011 ±   | 0.0024 ±   | 0.0055 ±   | 0.0037 ±     |
|       | 3.966E-4 ±  | 5.0156E-4 ± | 4.0298E-4 ± | 0.0012 ±   | 0.0018 ±   | 0.1189 ±   | 0.1133 ±     |
|       | 0.0009      | 0.0011      | 0.0012      | 0.0015      | 0.0021      | 0.0045      | 0.0021      |
|       | 2.1725E-5 ± | 3.1845E-5 ± | 0.0042 ±   | 0.0035 ±   | 0.0043 ±   | 0.1211 ±   | 0.1139 ±     |
|       | 6.2892E-5 ± | 7.1208E-5 ± | 0.0027 ±   | 0.0030 ±   | 0.0034 ±   | 0.0047 ±   | 0.0047 ±     |
|       | 0.0044 ±   | 0.0057 ±   | 0.0083 ±   | 0.0112 ±   | 0.0290 ±   | 0.1224 ±   | 0.1157 ±     |
|       | 0.0011      | 0.0014      | 0.0028      | 0.0026      | 0.0215      | 0.0050      | 0.0055 ±     |
| 40dB  | 3.514E-5 ±  | 0.0000 ±   | 1.2500E-4 ± | 3.1250E-4 ± | 3.7500E-4 ± | 0.1497 ±   | 0.1316 ±     |
|       | 2.4936E-4 ± | 1.7116E-4 ± | 2.2097E-4 ± | 2.6146E-4 ± | 0.0109      | 0.0144      |             |
|       | 0.0003 ±   | 6.2500E-4 ± | 0.0011 ±   | 0.0074 ±   | 0.0022 ±   | 0.1410 ±   | 0.1270 ±     |
| 30dB  | 1.0823E-4  | 1.3975E-4 ± | 6.0917E-4 ± | 0.0142 ±   | 7.9672E-4 ± | 0.0061      | 0.0075 ±     |
|       | 0.0062 ±   | 0.0067 ±   | 0.0101 ±   | 0.0084 ±   | 0.0168 ±   | 0.1428 ±   | 0.1507 ±     |
|       | 0.0010      | 0.0013      | 0.0011      | 0.0019      | 0.0037      | 0.0112      | 0.0118 ±     |
| 20dB  | 0.4624 ±   | 0.4708 ±   | 0.4711 ±   | 0.4709 ±   | 0.4715 ±   | 2.3530 ±   | 2.3050 ±     |
|       | 1.9460E-4 ± | 3.9460E-4 ± | 4.3625E-4 ± | 2.4608E-4 ± | 8.1614E-4 ± | 0.0380      | 0.0280 ±     |
|       | 0.4683 ±   | 0.4706 ±   | 0.4728 ±   | 0.4895 ±   | 0.4746 ±   | 2.3442 ±   | 2.3238 ±     |
| 30dB  | 6.4851E-4 ± | 1.3153E-4 ± | 0.0014 ±   | 0.0388 ±   | 0.0014      | 0.0306      | 0.0274 ±     |
|       | 0.4861 ±   | 0.4789 ±   | 0.4855 ±   | 0.4822 ±   | 0.4988 ±   | 2.3539 ±   | 2.3420 ±     |
|       | 0.0011      | 0.0016      | 0.0025      | 0.0022      | 0.0111      | 0.0342      | 0.0321 ±     |
| 40dB  | 0.9999 ±   | 1.0000 ±   | 0.9999 ±   | 0.9997 ±   | 0.9997 ±   | 0.8655 ±   | 0.8721 ±     |
|       | 5.5713E-5  | 6.9877E-5  | 9.7439E-5  | 1.1411E-4  | 2.2881E-4  | 0.0059      | 0.0031 ±     |
|       | 0.9997 ±   | 1.0000 ±   | 0.9993 ±   | 0.9947 ±   | 0.9985 ±   | 0.8640 ±   | 0.8689 ±     |
| 30dB  | 4.3712E-4 ± | 6.9877E-5 ± | 3.9713E-4 ± | 0.0105 ±   | 4.4234E-4 ± | 0.0026      | 0.0058 ±     |
|       | 0.9970 ±   | 0.9963 ±   | 0.9941 ±   | 0.9953 ±   | 0.9897 ±   | 0.8585 ±   | 0.8635 ±     |
| 20dB  | 0.0009      | 0.0014      | 6.7225E-4 ± | 0.0010      | 0.0028      | 0.0062      | 0.0065 ±     |
Table 1 tell us that the proposed algorithm performs better on the five evaluation metrics. In terms of Average precision, when the SNR is 20dB, this metric reaches 0.9970, which is higher than that of single DL method. This is because the voting method is used to vote on the label predictions of the three classifiers. When there are 2 or 3 predicted values as 1, the label will be 1. On the contrary, if 2 or 3 predicted labels are 0, the label will be 0, thus further improved the accuracy of label prediction.

Besides, the results are better than the traditional CNN with 3 Convolution and BN layers, as shown in 6th column in Table 1. Although the results of CNN are satisfying enough, the metrics of the algorithm is much higher than that of CNN, showing excellent performance.

B. COMPARISON WITH SINGLE TRADITIONAL METHOD

To illustrate the superiority of the proposed algorithm to the triple stages algorithm, the results are also compared with that of ST+MLRBF and that of ST+Rank-WSVM, with the kernel function Morlet wavelet. The results are shown in rightmost two columns in Table 1.

The statistic results in Table 1 show the proposed algorithm is significantly better than the traditional multi-label classification algorithm. With different SNR levels, the mean value of Average Precision of Rank-WSVM varies from 0.8721 to 0.8635. In contrast, the Average Precision of the proposed algorithm in this paper are higher than 0.9970 in despite of the SNR of signals, suggesting the robustness of the algorithm. The remaining four metrics are also less than the results of traditional classifier. From the results it is concluded that the algorithm is significantly better than the traditional triple stages methods.

V. REAL SIGNALS TEST

In order to verify the effectiveness of the algorithm in this article, the Power source FLUKE6105A provided by the Wuhan Branch of China Electric Power Research Institute, is used to generate the PQDs signals. These PQDs signals are sampled by oscilloscope LeCroy WaveRunner-604ZI, and finally sent to the PC via series port for training and test. As shown in Figure 6, the signal source FLUKE6105A generated a PQDs signal with a duration of 10 cycles and a sag duration of 3 cycles.

Due to the output limitation of Fluke6105A, it cannot generate the following three types of disturbances, including transient oscillation, voltage notch, and spike.

Therefore, the real PQDs signals set in this article only contains 6 types of single disturbances, 5 types of double disturbances and 3 types of triple disturbances. The relevant parameters of the PQDs signals are randomly generated. One hundred PQDs signals of each type of PQDs are generated, of which 80 are used for training and 20 for test. The specific disturbance types are shown in Table 2.

VOLUME 9, 2021

Accuracy will inevitably decline. In order to further improve the classification accuracy, the technique of “pre-training and re-training” is applied. The algorithm is firstly trained on the simulated signals, and then the trained model is retrained with the real PQDs signals, as a result, it further improved the classification accuracy. The classification accuracy of the proposed algorithm and the competing method are given in Table 3.

In order to visually display the data in Table 3, we draw two bar graphs, which show the “coverage” and “Average Precision”.

Figure 7 and figure 8 show the results are significantly better than that of Rank-SVM and ML-RBF, although the results of Proposed method are close to the results of CNN, the statistics still show better performance. Besides, it can
be seen from Table 3 that transfer learning technique can effectively improve the real signals classification accuracy. The mean value of “Average Precision” of the algorithm on the real PQDs signals set reaches 0.9998, which is higher than the three comparison algorithms, including the CNN, Rank-WSVM and ML-RBF. From the data in Table 3 we concluded that the algorithm with transfer learning has better adaptability and application prospects to the real PQDs signals.

VI. CONCLUSION

An ensemble architecture is designed for multi-label PQD classification. Firstly, the CNN-GRU, ResNet-GRU, and Inception-GRU networks are used to fit the training sets of PQDs signals under different SNR levels. Then, the trained networks are exploited for classification on the test set. Besides, the transfer learning technique is applied to improve the accuracy on real signals. Finally, the result of the label is determined by a voting method. In terms of the five metrics that are commonly employed for evaluating multi-label classification performance, the method proposed in this paper performs better than the traditional three-step method and a single DL method. The proposed method significantly improves the performance of multi-label classification for PQDs, providing more accurate information for further PQD analysis.

REFERENCES

[1] S. He, K. Li, and M. Zhang, “A real-time power quality disturbances classification using hybrid method based on S-transform and dynamics,” IEEE Trans. Instrum. Meas., vol. 62, no. 9, pp. 2465–2475, Sep. 2013.
[2] T. Zheng, S. Zhang, G. Cai, Y. Li, B. Yang, and Y. Chen, “Power quality disturbance recognition based on multiresolution S-transform and decision tree,” IEEE Access, vol. 7, pp. 88380–88392, 2019.
[3] C. M. Orallo, I. Carugati, S. Maestri, P. G. Donato, D. Carrica, and M. Benedetti, “Harmonics measurement with a modulated sliding discrete Fourier transform algorithm,” IEEE Trans. Instrum. Meas., vol. 63, no. 4, pp. 781–793, Apr. 2014.
[4] W. Gao and J. Ning, “Wavelet-based disturbance analysis for power system wide-area monitoring,” IEEE Trans. Smart Grid, vol. 2, no. 1, pp. 121–130, Feb. 2011.
[5] P. K. Ray, N.kishor, and S. R. Mohanty, “Islanding and power quality disturbance detection in grid-connected hybrid power system using wavelet and S-transform,” IEEE Trans. Smart Grid, vol. 3, no. 3, pp. 1082–1094, Sep. 2012.
[6] S. Shukla, S. Mishra, and B. Singh, “Power quality event classification under noisy conditions using EMD-based de-noising techniques,” IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1044–1054, May 2014.
[7] M. Biswal and P. K. Dash, “Measurement and classification of simultaneous power signal patterns with an S-transform variant and fuzzy decision tree,” IEEE Trans. Ind. Informat., vol. 9, no. 4, pp. 1819–1827, Nov. 2013.
[8] J. Li, Z. Teng, Q. Tang, and J. Song, “Detection and classification of power quality disturbance using double resolution S-transform and DAG-SVMs,” IEEE Trans. Instrum. Meas., vol. 65, no. 10, pp. 2302–2312, Oct. 2016.
[9] R. G. Stockwell, L. Mansinha, and R. P. Lowe, “Localization of the complex spectrum: The S transform,” IEEE Trans. Signal Process., vol. 44, no. 4, pp. 998–1001, Apr. 1996.
[10] C. Zhao, K. Li, Y. Li, L. Wang, Y. Luo, X. Xu, X. Ding, and Q. Meng, “A novel method based on variational mode decomposition and a random discriminative projection extreme learning machine for multiple power quality disturbance recognition,” IEEE Trans. Ind. Informat., vol. 15, no. 5, pp. 2915–2926, May 2019.
[11] R. Hooshmand and A. Enshaee, “Detection and classification of single and combined power quality disturbances using fuzzy systems oriented by particle swarm optimization algorithm,” Electr. Power Syst. Res., vol. 80, no. 12, pp. 1552–1561, 2010.
[12] K. Manimala and K. Selvi, “Power disturbances classification using S-transform based GA-PNN,” J. Inst. Eng., vol. 96, no. 3, pp. 283–295, Sep. 2015.
[13] E. R. Kumar, B. Singh, D. T. Shahani, A. Chandra, and K. Al-Haddad, “Recognition of power-quality disturbances using S-transform-based ANN classifier and rule-based decision tree,” IEEE Trans. Ind. Appl., vol. 51, no. 2, pp. 1249–1258, Mar./Apr. 2015.
[14] S. R. Mohanty, P. K. Ray, N. Kishor, and B. Panigrahi, “Classification of disturbances in hybrid DG system using modular PNN and SVM,” Int. J. Elect. Power Energy Syst., vol. 44, no. 1, pp. 764–777, 2013.
[15] E. Styvaktakis, M. H. J. Bollen, and I. Y. H. Gu, “Expert system for classification and analysis of power system events,” IEEE Trans. Power Del., vol. 17, no. 2, pp. 423–428, Apr. 2002.
[16] M. B. I. Reaz, F. Choong, M. S. Sulaiman, F. Mohd-Yasin, and M. Kamada, “Expert system for power quality disturbance classifier,” IEEE Trans. Power Del., vol. 22, no. 3, pp. 1979–1988, Jul. 2007.
[17] P. Janik and T. Lobos, “Automated classification of power-quality disturbances using SVM and RBF networks,” IEEE Trans. Power Del., vol. 21, no. 3, pp. 1665–1669, Jul. 2006.
[18] M.-L. Zhang, J. M. Peña, and V. Robles, “Feature selection for multi-label naïve Bayes classification,” Inf. Sci., vol. 179, no. 19, pp. 3218–3229, Sep. 2009.
[19] Y. Luo, K. Li, Y. Li, D. Cai, C. Zhao, and Q. Meng, “Three-layer Bayesian network for classification of complex power quality disturbances,” IEEE Trans. Ind. Informat., vol. 14, no. 9, pp. 3997–4006, Sep. 2018.
[20] F. A. S. Borges, R. A. S. Fernandes, I. N. Silva, and C. B. S. Silva, “Feature extraction and power quality disturbances classification using smart meters signals,” IEEE Trans. Ind. Informat., vol. 12, no. 2, pp. 824–833, Apr. 2016.
[21] P. R. Babu, P. K. Dash, S. K. Swain, and S. Sivanagaraju, “A new fast discrete S-transform and decision tree for the classification and monitoring of power quality disturbance waveforms,” Int. Trans. Electr. Energy Syst., vol. 24, no. 9, pp. 1279–1300, Sep. 2014.

[22] J. Liu, H. Song, H. Sun, and H. Zhao, “High-precision identification of power quality disturbances under strong noise environment based on FastICA and random forest,” IEEE Trans. Ind. Informat., vol. 17, no. 1, pp. 377–387, Jan. 2021.

[23] L. Zhou, C. Guan, and W. Lu, “Application of multilabel classification method to categorization of multiple power quality disturbances,” Proc. CSEE, vol. 31, no. 4, pp. 45–50, 2011.

[24] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[25] Z. Liu, Y. Cui, and W. Li, “A classification method for complex power quality disturbances using EEMD and rank wavelet SVM,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 1678–1685, Jul. 2015.

[26] S. Wang and H. Chen, “A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network,” Appl. Energy, vol. 235, pp. 1126–1140, Feb. 2019.

[27] J. Wang, Z. Xu, and Y. Che, “Power quality disturbance classification based on compressed sensing and deep convolution neural networks,” IEEE Access, vol. 7, pp. 78336–78346, 2019.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[29] R. Gong and T. Ruan, “A new convolutional network structure for power quality disturbance identification and classification in micro-grids,” IEEE Access, vol. 8, pp. 88801–88814, 2020.

[30] M. Sahani and P. K. Dash, “FPGA-based deep convolutional neural network of process adaptive VMD data with online sequential RVFLN for power quality events recognition,” IEEE Trans. Power Electron., vol. 36, no. 4, pp. 4006–4015, Apr. 2021.

[31] N. Mohan, K. P. Soman, and R. Vinayakumar, “Deep power: Deep learning architectures for power quality disturbances classification,” in Proc. Int. Conf. Technol. Adv. Power Energy (TAP Energy), Dec. 2017, pp. 1–6.

[32] W. Qiu, Q. Tang, J. Liu, and W. Yao, “An automatic identification framework for complex power quality disturbances based on multilayer convolutional neural network,” IEEE Trans. Ind. Informat., vol. 16, no. 5, pp. 3233–3241, May 2020.

[33] IEEE Recommended Practice for Monitoring Electric Power Quality, IEEE Standard 1159-2009, 2009.

XIANGUI XIAO was born in Hubei, China, in 1988. He received the B.E. degree from the Wuhan University of Technology, Wuhan, China, in 2011, and the M.E. degree from the Shanghai University of Electrical Power, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Advanced Electromagnetic Engineering and Technology, Huazhong University of Science and Technology, Wuhan. His current research interests include signal processing and deep learning in power systems.

KAICHENG LI was born in Hubei, China, in 1962. He received the B.E. and M.E. degrees in electrical engineering and the Ph.D. degree in power electronics from the Huazhong University of Science and Technology, Wuhan, China, in 1985, 1988, and 1998, respectively. He is currently a Professor with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology. His research interests include measurement and control technology, power quality analysis, electronic voltage, and current transformer.