Intelligent fault detection and location scheme for modular multi-level converter multi-terminal high-voltage direct current

Yang, Qingqing; Li, Jianwei; Santos, Ricardo; Huang, Kaijia; Icgi, Petar

Published in:
High Voltage

Link to article, DOI:
10.1049/hve2.12033

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Intelligent fault detection and location scheme for modular multi-level converter multi-terminal high-voltage direct current

Qingqing Yang1 | Jianwei Li2 | Ricardo Santos3 | Kaijia Huang4 | Petar Igel1

1Coventry University, Coventry, UK
2Department of Engineering Science, University of Oxford, Oxford, UK
3Centre of Engineering, Federal University of ABC, Santo Andre, Brazil
4DTU Energy, Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

In order to overcome the drawbacks of the conventional protection methods in high-voltage direct current transmission lines, a deep learning approach is proposed that directly learn the fault conditions based on unsupervised feature extraction to the detection and location decision by leveraging the hidden layer activations of recurrent neural network. The deep-recurrent neural network boosting with the gated recurrent unit compared with the long short-term memory unit is used by analysing both the signal presented in time domain and frequency domain. The proposed method is tested based on a modular multilevel converter based four-terminal high-voltage direct current system. Various faults under different conditions were simulated against fault resistance, external faults and small disturbance immunity with the validity, and the simulation verified a high accuracy, robustness and fast results because of the utilization of characteristic feature extraction.

1 | INTRODUCTION

High-voltage direct current (HVDC)/Muti-terminal direct current (MTDC) is proposed as a promising technology for super-grid or collecting bulk renewable energy sources, that the amount of projects has been considerably increased since the last decade [1–3]. However, the insensitivity of high impedance, unsoundness of close to terminal fault detection and distraction of noise result in immaturity of the HVDC protection system. With the strict target of greenhouse-gas reduction and large penetration of the renewable energy, the new technologies are imperatively needed in today’s grid system [4–7]. With the vigorous development of smart grid, the monitoring data of power systems are increasingly rich [8,9]. Hence, the intelligent methods’ outstanding performance in analysing data is becoming a hot topic in the research of fault detection.

The modular multi-level converters (MMCs) have made a breakthrough of HVDC technology. Hence, the protection of MMC based MTDC systems is one of the critical challenges [10,11]. Protection system is essential to detect, isolate and identify faults and to aid in troubleshooting as fault currents can reach very high values that may bring severe influence to the whole system. Thus, the DC transmission line protection including fault detection and fault location is overriding importance to ensure the security of the entire MTDC transmission system [12,13].

Travelling wave principle is generally used in HVDC protection scheme to capture the transient travelling surges on transmission lines. Travelling wave based protections have obvious advantage of using local measurements and fast operation speed in single-ended method [14] and using only the initial surge caused by the fault in the double-end method with high reliability as well as high location accuracy [15]. Current differential, voltage differential and overcurrent are all other conventional protection methods [16]. However, conventional protection schemes based on travelling wave principle for HVDC transmission line exhibit inherent flaws that they have relatively low sensitivity for high resistance faults, need Global Positioning System to keep measurement
time synchronized [17] and have weak tolerance of uncertainties by measuring wave-head of travelling wave.

The research on transmission line protection by applying artificial intelligence (AI) technique has been proposed for many years, but few of them have been put into practice. However, with the fast and mature development of deep learning (DL) methods in recent years, it is potential to use AI techniques in the real power systems. For power system fault studies, various AI methods including expert systems [18], Fuzzy system [19], Bayesian networks [20], rough set [21], Petri nets [22], neural networks (NNs) [23,24], support vector machine [25] and so on are applied in fault detection. It shows great success and robustness. Papers [26,27] have dealt with the possibility of applying artificial neural network (ANN) in high voltage AC transmission line. The performance ANN in distribution lines also shows effectiveness as presented in paper [28,29]. DL methods can also apply in industrial processes [30]. The DL method auto-encoder effectively applied in fault diagnosis system [31]. In HVDC system, the application of a radial basis function NN is presented in [32] for fault diagnosis. Series DC arc faults are effectively diagnosed in paper [33]. DL methods can effectively solve the problems of uncertain correspondence such as the influence of line parameters, transition resistance, noise and so on. Different types of AI techniques have their own advantages and disadvantages. The comparison of the advantages and disadvantages of different types of Intelligent algorithms is listed in Table 1.

Table 1. Comparison of the advantages and disadvantages of different types of intelligent algorithms [34–36]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian network</td>
<td>Easier computation process, better speed and</td>
<td>Large error with dependency among variables</td>
</tr>
<tr>
<td></td>
<td>accuracy for huge datasets</td>
<td></td>
</tr>
<tr>
<td>Support vector machine</td>
<td>Easily identify complex non-linear data</td>
<td>Expensive and slow computational process</td>
</tr>
<tr>
<td>Expert systems</td>
<td>Easy to adapt to new conditions</td>
<td>High cost, not error-free</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>Linguistic variables</td>
<td>Cannot deal with incomplete information in explicit form</td>
</tr>
<tr>
<td>Artificial neural network</td>
<td>Direct complex data processing can handle noisy data</td>
<td>Large number of samples required, over-fitting problem</td>
</tr>
<tr>
<td>Fuzzy neural network</td>
<td>Robustness in relation of the disturbances</td>
<td>Intensive training time</td>
</tr>
</tbody>
</table>

In previous study by using ANN in the MTDC system fault detection and fault location, a short time step is applied in the fault detection stage [23]. A fix-size input is set in conventional NN; however, recurrent neural network (RNN) will be able to learn with a certain size input, which is a big advantage to model the entire time series better. In a simple word, RNN is able to model sequence of data so that each sample is assumed to be dependent on previous ones. The speed of the training process is overriding importance with no matter how much training data it has that will make the difference in performance with neural nets. Theoretically, RNNs can process varying-length series straightforwardly, while ANNs or CNNs need additional process to work on.

In this study, the fault detection and fault location method by using the RNN-gated recurrent unit (GRU) for MMC-MTDC is proposed. The key constraint for applying the DL method in a fault detection and fault location method in HVDC system is that there is no sufficient data for the system in the training process and lack of computational tools. However, with the development of smart grid, the constraints are tackled by the digitalization of equipment and the development of high-performance computing. Unlike the AC network protection, a more robust method is expected for the protection system against uncertainty. Due to the use of DC circuit breaker (DCCB), a much faster relaying is required [37]. The proposed method is capable of meeting the requirement of fast acting of DCCB with local data processing. In addition, most existing fault detection method is not sensitive for high impedance fault and may be interrupted by some external factors. The frequency-domain signal of fault current is analysed for fault detection because specific frequency components especially high-frequency components which are caused by high fault current will be generated by the rapidly rising of fault current. The analysis in frequency domain can efficiently detect and locate the high impedance faults. Hence, a novel fault detection method with new multi-source information and multi-frequency components relaying technique is proposed. A fault detection and location method is studied, in which both the features generated from local data in time domain and frequency domain are utilized effectively. The fault features are deeply excavated by DL algorithm to gain a better awareness of the nature of the fault classification or location problems [38].

Keeping in view of above perspective and issues, the contributions of this work are as follows: (i) A novel MMC-MTDC system fault detection method based on the deep-RNN structure with the GRU algorithm from the data-driven viewpoint is presented, in which the fault signal can be detected and identified in a short time in milliseconds; (ii) An integrated high precision HVDC fault locator is designed, in which both the features in time domain and frequency domain can be utilized effectively; (iii) The accuracy, stability and robustness of the proposed methodologies are verified through a four-terminal HVDC built in PSCAD to provide fault types and location information; (iv) A comprehensive comparison with the GRU method and with the Long Short-term Memory (LSTM) algorithm is performed.

This study is composed of five sections. Section 2 briefly introduces the structure of proposed fault detection and location method for DC transmission line in MTDC system.
2 | THE PROPOSED METHOD FOR DC TRANSMISSION LINE FAULT DETECTION IN MTDC

Fault detection in MTDC systems is overriding importance for fast system recovery. When a fault occurs, the fault detection scheme should fast detect the existing of a fault, determine the fault types and locate the fault point as quickly as possible to avoid severe damage to the converter station and the rest of the network. Meanwhile, a restoration of power delivery to the affected lines beyond the faulted zone is achieved by power reversal to maintain the network security. The fault detection and fault classification enable subsequent fault isolation operations, while the fault location can benefit service restoration.

The features of fault signals can be exacted in time domain and frequency domain by using various transforms or techniques. However, in the conditions of high impedance fault, line end fault or fault caused by lightning strike, the reasonable features selections are difficult to be distinguished and classified from the steady-state condition and other types of fault by conventional threshold or criterion. The transient disturbance cannot be adequately described in time domain and frequency domain through feature captures which will result in low accuracy. Hence, the data-driven based DL method is proposed to solve the above problem compared with traditional method.

2.1 | Deep-RNN algorithm based HVDC system fault detection

DL is one of the branches and techniques of machine learning, which originated from the Perceptron [39]. DL is realized to be computing systems by establishing ANNs with hierarchical structure. It is an effective tool to solve the real-time related problems through learning and training and has been successfully used in fault diagnose to treat it as a discriminative problem.

With the rapidly development of DL, the recurrent NN, which is a kind of NN for tasks that involve sequential inputs (processing sequence data) [39], is applied in this study to perform the fault detection issue. One of the characteristics of RNN’s network architecture is that the state of the network depends not only on the input but also on the state of the network in previous moment. RNN addresses the issue to classify a status at every point in a time line because RNN has loops inside allowing information to persist. Unlike the feed forward NN which is known as Directed Acyclic Graph, there is at least one loop in the structure of RNN that the state transition of \( b \) occurs in the time dimension.

The training data set is reshaped as feature-time series matrix. The row variables represent time series and the column variables represent features in the training data set. The signals collected by 16 detection points (L1-1p, L1-1n, L1-2p, L1-2n, L2-1p, L2-2p, L2-2n, L2-3p, L2-3n, L3-1p, L3-4p, L3-4n, L4-1p and L4-1n) are used as the features in deep-RNN model.

The feature-time series matrix is transformed into feature vector by the input layer of deep-RNN network. The time series information is presented by the self-connection weight \( \mathcal{W}_{1,1} \). The process of transformation is presented by the following equation:

\[
H_t^i = b_m + \mathcal{W}_{1,1}b_{t-1} + \mathcal{W}_{1,\text{in}}F_{TSM,t}^i.
\]

\[
F_{TSM,t}^i = f_{\text{activation}}(h_t^i).
\]

The fixed feature-time series matrix is presented as:

\[
F_{TSM,t,1} = \begin{bmatrix}
F_{1}^{1,1} & F_{2}^{1,1} & \cdots & F_{t}^{1,1} \\
F_{1}^{1,2} & F_{2}^{1,2} & \cdots & F_{t}^{1,2} \\
\vdots & \vdots & \ddots & \vdots \\
F_{1}^{1,n} & F_{2}^{1,n} & \cdots & F_{t}^{1,n}
\end{bmatrix}
\]

where: \( H_t^i \) represents the state of the input layer nervous at time \( t \), which implicitly contains information about the history of all the past elements of the sequence, \( b_m \) is the input value which evaluated by the activation function, \( F_{TSM}^i \) represents the input vector at time \( t \), \( F_{1}^i \) represents the output vector at time \( t \) and \( f_{\text{activation}} \) is the activation function of the nervous layer in deep-RNN network.

After the feature-time series matrix is transformed into the feature vector form, the fault information \( F_{TSM} \) in time domain is represented as \( F_{1}^i \) in feature domain. To further improve the performance of fault detection efficiency and accuracy, hidden layer is set in deep-RNN network. The computation principle is presented by the following equation:

\[
F_{1}^i = f_{\text{activation}}(b_i + \mathcal{W}_{ii}H_{t}^{i-1} + \mathcal{W}_{i,i-1}H_{t-1}^{i-1})
\]

To predict class labels in fault detection, the network ends with a softmax layer and a classification output layer. The softmax function \( \sigma(z)_j \) maps multiple scalars into a probability distribution with each value of the output at (0,1). The formula is shown as follows:

\[
\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, 2, \ldots, K
\]
The derivative of the softmax layer is divided into two cases:

i. The node is the output category \( (i = j) \)

\[
\frac{\partial a_j}{\partial z_i} = \frac{\partial}{\partial z_i} \left( e^i \sum_{k,j} e^j \right) = \frac{(e^j) \sum_k e^k - e^j e^i}{(\sum_k e^k)^2} = a_j(1 - a_j) \tag{6}
\]

ii. The node is not the output category \( (i \neq j) \)

\[
\frac{\partial a_j}{\partial z_i} = \frac{\partial}{\partial z_i} \left( e^i \sum_{k,j} e^j \right) = \frac{(e^j) \sum_k e^k - e^j e^i}{(\sum_k e^k)^2} = -a_j a_i \tag{7}
\]

where: \( a_j \) represents the value of the aggregate calculation, \( z_i \) represents the control of each LSTM input, \( e \) represents multiple scalars.

The basic principle of the RNN training is essentially back propagation (BP). The BP algorithm will be used to train the fault detection model; the core principle of the BP algorithm is described by the following equations:

\[
\delta^{k-1} = \frac{\partial C}{\partial \theta} \bigg/ \frac{\partial h^{k-1}}{\partial \theta} = (W^f_i \delta^k) \odot \sigma(h^{k-1}) \tag{8}
\]

\[
\frac{\partial C}{\partial w_{ij}} = \sum_{k=1}^{t} \frac{\partial h_k^i}{\partial w_{ij}} \frac{\partial C}{\partial h_k^i} = \sum_{k=1}^{t} \frac{\partial h_k^i}{\partial w_{ij}} \delta^k \tag{9}
\]

However, its BP method is called BPTT (BP through time). The process is the same with BP algorithm that forward calculating the output value of each neuron in the first step. Then, the error term value of each neuron, which is the partial derivative of the weighted input of the error function \( E \) to the neuron \( j \), is inversely calculated. The final step is to calculate the gradient of each weight. However, there is a long-term dependency problem in RNN training. It turns out that two problems exist during the training of RNN: vanishing gradients and exploding gradients. The so-called gradient disappearance and gradient explosion refer to the gradient in each calculation and BP during training. Time is always inclined to increase or decrease. After a period of time, when the gradient converges to zero (gradient disappears) or diverges to infinity (gradient explosion), the parameters will not be updated at this time, which means the later training will not have an impact on the update of the parameters. In other word, the long-term dependence problem is that the hidden layer at the moment will lose the ability to connect to distant information when the time interval increases. Minimizing a loss function is involved in training a NN. In the iterative process, the gradient descent algorithm is used to minimize the loss function. The gradient of the loss function is recalculated, and the weights are updated in each iteration.

### 2.2 Boosting with long short-term memory unit

The LSTM is a special RNN, which is mainly used to solve the gradient disappearance and gradient explosion problems in long sequence training. In brief, the LSTM has a better performance in longer sequences than the normal RNN. The LSTM is a good recurrent NN that captures patterns from sequence data in power systems like current, voltage and frequency spectrum. It has a secure and powerful modelling capability in temporal and sequential structure with specific internal states in different time series data or sequence.

Compared with RNN, the LSTM has an implicit state variable formula called cell state, which is used to record information. The LSTM based RNN has three more controllers: input control, output control and forgetting control.

An interchangeably block called a memory cell is the core of the LSTM, which maintains its state over time. Each LSTM unit uses three gates including input gate \( i_t \), forget gate \( f_t \) and output gate \( o_t \) to determine the information retained.

The forgotten gate \( f_t \) selects the last cell state \( c_{t-1} \), which determines how much the cell state of the previous time \( c_{t-1} \) remains to the current time \( c_t \). The forget gate selectively forgets the last unit state \( c_{t-1} \), and the obtained output is the memory of historical information as shown in equation.

\[
f_t = \sigma(\omega_i [h_{t-1}, x_t] + b_f) \tag{10}
\]

where \( x_t \) is the input vector at the time \( t \), \( h_{t-1} \) is the output of previous block. The signal processing at each moment will directly spliced the two matrices \( x_t \) and \( h_{t-1} \) together, using \([h_{t-1}, x_t]\) as input with appropriate weight \( \omega \) and biasing \( b \). The input gate determines how much the input \( x_t \) of the network is saved to the unit state at the current time \( c_t \).

\[
i_t = \sigma(\omega_i [h_{t-1}, x_t] + b_i) \tag{11}
\]

The output gate determines the current output value \( h_t \), which is obtained from the state of the control unit \( c_t \) of the LSTM. The output gate selectively exports the state of the current unit, and the final output \( h_t \) is obtained.

\[
o_t = \sigma(\omega_o [h_{t-1}, x_t] + b_o) \tag{12}
\]

The sum of the memory of the historical information and the information entered this time is the current unit state \( c_t \), which is the element-wise summation. The current unit status \( c_t \) is illustrated in the following equations in which \( C_t \) is the input data.

\[
c_t = f_t c_{t-1} + i_t C_t \tag{13}
\]

\[
C_t = \tan b(\omega_i [h_{t-1}, x_t] + b_c) \tag{14}
\]
The output control decides the new memory to the next LSTM block that the gate is controlled by the previous output, the current input and the new memory. The final output is:

\[ b_t = a_t \tan b(C_t) \quad (15) \]

### 2.3 Boosting with gated recurrent unit

The GRU is a variant of LSTM without an output gate, which makes the net easier to build and thus faster to run, because the GRU uses fewer connections that write the contents from its memory cell to the larger net at each time step.

Compared with the LSTM, the GRU model maintains the LSTM effect with a simpler structure, fewer parameters and better convergence which only consists of update door and reset door. The GRU combines the input gate and the forgetting gate into one which is called update gate. \( z_t \) controls the amount of data that can be retained to the current moment by the front memory information, and decides how much information of the previous time step and the current time step should be transferred to the future. Another GRU gate is called the reset gate \( r_t \) which controls how much information from the past should be forgotten.

Reset gate \( r_t \) is used to control the influence of the hidden layer element \( h_{t-1} \) at the previous moment on the current status \( x_t \). If \( h_{t-1} \) is not essential to \( x_t \), it means that the current status \( x_t \) has no fault detected. Then the switch \( r_t \) will be turned on so that \( h_{t-1} \) has no effect on \( x_t \).

\[ r_t = \sigma(W_z[h_{t-1}, x_t]) = \sigma(W_zx_t + U_zr_{t-1} + b_r) \quad (16) \]

Update gate \( z_t \) is used to determine whether to ignore the current state \( x_t \). Similar to the input gate \( i_t \) in the LSTM, \( z_t \) can judge whether the current state \( x_t \) is essential to the following detection. When \( z_t \) switch is connected to the following branch, the current state \( x_t \) will be ignored, forming a short-circuit connection from \( h_{t-1} \) to \( h_t \), which allows the gradient to be effectively back-propagated. Like the LSTM, this short-circuit mechanism effectively alleviates the phenomenon of gradient disappearance.

\[ z_t = \sigma(W_z[h_{t-1}, x_t]) = \sigma(W_zx_t + U_zb_{t-1} + b_z) \quad (17) \]

Instead, linear self-updating is carried out directly in the hidden unit by means of gate control

\[ \tilde{h}_t = \tan h(W[r,h_{t-1}, x_t]) = \tan h(Wx_t + r_t \tilde{h}_{t-1} + b) \quad (18) \]

\[ h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (19) \]

### 2.4 Hybrid fault detection and location method

The proposed RNN-GRU method which is shown in Figure 1 consists of three steps: (i) Signal processing of current and voltage feature extraction in frequency domain through continuous wavelet transform (CWT) analysis based on fast Fourier transform based algorithm, (ii) Real time fault detection based on classification type recurrent NN and (iii) Precise fault locating based on regression type recurrent NN. From Figure 1, the proposed method is divided into two stages, one for fault detection and classification (P1) which is the deep-RNN based fault detection and one for fault location (P2) which is the hybrid method for precious fault positioning.

The hybrid fault detection and location method is based on the deep-RNN method. When a fault occurs in the system, the increasing current will result in changes in the system. Both the current and voltage signals in time domain and frequency domain will be used in unsupervised feature extraction for NNs. Through softmax layers and classification layers, the system will tell whether there is a fault. The fault location will be determined through regression layer of the network.

To analyse the fault location of HVDC system, the further information in the fault signal should be extracted and utilized. In high accuracy fault locating, the frequency domain information should be extracted and utilized. The fast Fourier transform (FFT) is the main approach for spectral analysis to characterize the magnitude and phase of a signal. In addition, the acquired data is enhanced by using digital filtering. In order to prove the frequency domain transformation quality, spectral windowing is applied using Hanning. The window pre-multiplies input data is provided to an FFT with a value that is smoothly reduced to zero at each end of the data. The purpose is to reduce ‘leakage’ aberrations in the output that are introduced by sudden changes in the data at the start and end of data. Much research has been applied to the selection of the proper ‘windowing function’, so the system signal has been analysed in several frequency bands.

The high-frequency travelling wave signal contains a large amount of frequency domain and time domain information. The analysis of the cause of the fault provides sufficient data support, and the identification of the fault cause is more obvious to be realized. However, most of the actual live data is recorded data with a sampling frequency of less than 10 kHz. The frequency and time domain information are relatively small, which makes the identification of the cause of the fault difficult. Therefore, the time domain information in fault time series are also used in fault locating issue to further improve the accuracy and performance of HVDC fault locating system in this study. To simplify the model training process, the trained deep-RNN based fault detection model will be used as the time domain feature attractors in fault locating issue. The multi-frequency domain and multi-channel time domain information are combined to perform real time fault location. The corresponding process is presented by the following equations:
i. The fault signals in time domain are transformed into frequency domain based on FFT algorithm:

\[
F(\omega)_{n_2} = \mathcal{F}[FTSM_{t,2}(t)] = \sum_{n=-\infty}^{\infty} FTSM_{t,2}(t)e^{-jwn}
\]  

(20)

in which \( n \) represents the sample frequency in FFT algorithm.

ii. After signal processing, the features in frequency domain will go to the full connected layer together with the original system data in time domain followed by a regression layer to obtain the fault location results. The formation of training data matrix is:

\[
F_{TRAIN} = \begin{bmatrix}
F_{1}^{L_1-1} & F_{1}^{L_1-1}(\omega) & F_{2}^{L_1-1} & F_{2}^{L_1-1}(\omega) & \cdots & \cdots & \cdots & \cdots \\
F_{1}^{L_2-1} & F_{1}^{L_2-1}(\omega) & F_{2}^{L_2-1} & F_{2}^{L_2-1}(\omega) & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots \\
F_{1}^{L_n-1} & F_{1}^{L_n-1}(\omega) & F_{2}^{L_n-1} & F_{2}^{L_n-1}(\omega) & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
\]  

(21)

iii. The labels in the training process are the fault locations, and the fault locators are trained based on BPTT algorithm.

The fault features are presented in both time domain and frequency domain. The signals in time domain will be divided into segments so that they can directly be used in the method to reduce the detecting period and also used to generate frequency features through signal processing. For a more accuracy and robust fault detection method, the presentation in frequency domain will be obtained through CWT based on FFT algorithm which convert the time-series data set into mutually orthogonal [40]. The time–frequency domain characteristics can be extracted through the preprocessing using signals generated from the system. The frequency of different systems has different responses to the frequency in groups, so that the frequency will be divided into different frequency bands, which will have more efficient feature extraction in both low frequency and high frequency range. As a consequence, a fault detection with real-time action and fast response, and fault location with high precision is expected.

In addition, the fault detection method is trained on the basis of the training data. Once the fault is detected, the trained network will be updated by incorporating the new data into the original training data. Also, the data in steady-state condition will be added and trained for a more robust protection system.

### 3 | EXPERIMENTAL SETUP AND SYSTEM MODELLING

As the most advanced DC transmission technology, the MMC based HVDC/MTDC has the advantages of supplying power to the passive grid (or island mode), quickly and independently controlling active and reactive power, fast load flow reversal and flexible operation mode [41]. It has been successfully applied in the fields of wind power grid connection, grid interconnection, island mode operation and weak grid power supply and urban power supply [42]. A four-terminal voltage sourced converter
(VSC) based MMC-MTDC system with meshed topology was modelled using the PSCAD in the case study.

3.1 | Experimental setup and system modelling

The basic configuration of the whole test system is presented in Figure 2. The generated power of offshore wind power plant is collected by wind farm MMCs and injected into the system on grid 2 and grid 4. The pumped-storage hydroelectricity station is located on grid 3. Overhead transmission lines are used to transmit the power to the point of common coupling (PCC). Then the power is allocated to each grid side VSC according to the control strategy.

The DC rated voltage level is ±500 kV. The DC transmission line uses the bipolar topology with metal circuit wiring, and the inverter adopts a half-bridge modular multi-level topology. Each DC line is equipped with a DC circuit breaker and a set of current limiting reactors. Capacity in T1, T2, T3 and T4 is 3000, 1500, 1500 and 3000 MW, respectively. The AC frequency is set to be 50 Hz. The transmission line from T1 to T2 to T3 to T4 and back to T1 is 400, 800, 800 and 1000 km, respectively, in length. The studied network integrated with remote wind and hydro energy in a transmission with ring construction that ensures optimization of power flow. This system is built on the PSCAD software platform for detecting and reflecting transient behaviour. This study has been developed with the parameters-line-model which accurately captures the high frequency transient response. Also, the distributed parameter line model is used, which represents wave propagation phenomena and line end reflections with much better accuracy and are frequency dependent [43].

The training process is important for the robustness of the method. As the simulation work must prove the concept first on one system, the training data is limited to one system in this study. In order to reduce the uncertainties, more training data will be used to produce a more generic system in later work.

3.2 | DC fault transient analysis for four-terminal MMC HVDC

Power transmission line faults are the main concern among all kinds of faults in MTDC. Many fault detection and fault location methods for DC transmission lines already exist. The environmental conditions such as rough terrain and bad weather conditions inevitably cause frequent faults on the MTDC transmission lines. The complexity of faults in a large power system lies in the significant volatility uncertainty. The patterns may reflect faults like noise, distance, fault impedance and external faults. Electricity interruptions may cause several economic impacts.

3.2.1 | Internal faults

This study mainly focuses on the DC transmission line protection. In the event of a DC line fault, the DC breaker command is issued immediately after the protection action. The DC circuit breaker will be closed after the setting of arc-extinguishing time. If the reclosure is unsuccessful or the fault is identified as a permanent fault, the DC circuit breaker will be tripped off again meanwhile the DC breaker will be locked. If one line has been tripped, the upper line overload control coordinates the power return of other converter stations to ensure the safety of the equipment and realize the coordination of the converter station in the DC grid. The underlying line overload control realizes the power drop of the station and ensures the equipment safety.
Pole-to-pole fault and pole-to-ground fault, which are analysed in the simulation network, are the two most common overhead line short circuit faults. Among two types of fault, pole-to-pole faults are not common fault in DC transmission line but have severe effect for the system, which is caused by direct contact or insulation breakdown between two poles of conductors of DC lines.

### 3.2.2 External fault and disturbances

The faults that occur on AC side will be distinguished from internal faults. In a DC system, smoothing reactors and the capacitors installed at both ends of the DC transmission line and the inherent bus bar capacitance represent a natural boundary that does not transmit high frequency signals, and thus can be used to rapidly distinguish between internal and external faults [44]. Surge disturbances, for example, lightning strokes, will not affect DC side fault detection. The transient waveforms of surge interferences look similar to the ones of ground faults in some cases. The detection function only acts upon DC transmission line faults within a designated zone and will be stable to other types of disturbances or external faults.

### 3.3 Signal processing

This algorithm uses a full-cycle Moving Data Window to obtain current samples at the terminals. The Scalogram, which shows the absolute value of the CWT coefficients, is used to explain the features in frequency domain by analysing real-world signals especially for the signal punctuated by abrupt transients. First, the signals are divided into equal-length segments. The segments must be short enough (0.5 ms) in order to satisfy the time requirement by DC circuit breakers. The overlapped segments are used in this study. Then, each segment is windowed and computes its spectrum to get the short-time Fourier transform. The sampling frequency is set to 10 kHz. The length of the time series is 0.005 s with 500 sample points. Each window is 0.5 ms. Scalogram in three dimensions (frequency, time and power) for current signal (using positive-to-ground [PG] fault on L1 50% from T1 0.01 Ω as an example) clearly shows the change before and after internal fault occurs from the spectrum power in Figure 3.

Figure 4 shows the Scalogram for four different states including the steady-state condition, external fault condition, fault with low fault resistance and fault with high resistance occurring near the stations. The highest spectrum power and average spectrum power is collected from the Scalogram for both steady-state (Table 2), external faults (Table 2), internal faults including PG, negative-to-ground (NG) and positive-to-negative (PN) detected on healthy line (Table 3) and internal faults detected on faulted line (Table 3) in various conditions. It can be noticed from both Figure 4 and Tables 2 and 3 that the spectrum power is higher in steady-state and external faults conditions and also in healthy lines on normalized frequency 0.312 rad/ sample. The scale power is the accumulation based on sample time. During a fault, there is an attenuation on low frequency bands and the appearance of high frequency bands. For specified frequency bands, a large change will be noticed.

### 3.4 Program implementation

The DL program is designed with multiple stages including signal pre-processing, data pooling, data sampling, network training and evaluation. The specific algorithm pseudo code is shown in Program for fault detection (P1) and fault location (P2) respectively.

**P1:** RNN-GRU program for fault detecting in MTDC system

- Load the real time dataset from local terminals
- Set input values: Learning rate \( \eta_i \), parameters weight matrix \( W_{ij} \), max epoch M1 of GRU algorithm, total number of batches B
- Build deep-GRU with network size \((h,k)\) in MATLAB
- Generating the training dataset matrix \( FTSM_{1,1} \)
- Transferring time domain \( FTSM_{1,1} \) to feature domain \( F_1 \)
- while epoch \( k < M \) do
  - Train hidden layer of GRU network with \( FTSM_{1,1} \)
- end if
- Fault detection based on softmax functions and extracted features \( F_1 \)
  - At \( k_{th} \) epochs do:
    - Train the softmax layer \( D \), maps multiple scalars into a probability distribution
  - end for

**P2:** Hybrid fault locating method in MTDC system

- Set input values: Learning rate \( \eta_j \), parameters weight matrix \( W_{ij}^{2} \), max epoch M2 of regression algorithm
Load fault information matrix $\mathcal{FTSM}_t$ and $\mathcal{F}_t^1$ form program 1

Extracting the frequency domain information $F(t)_{x,2}$ from $\mathcal{FTSM}_t$ based on Fourier transform

Merging information of time-frequency domain to generate the training data set $\mathcal{F}_{\text{TRAIN}}$

Build regression neural network in MATLAB.

Train neural network for fault locating

while epoch $k < M2$ do

Train the network (L) with $\mathcal{F}_{\text{TRAIN}}$

end for

**TABLE 2** Spectrum power for steady-state and external faults condition on nomialized frequency 0.312 RAD/sample detection on L1-1P

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Time</th>
<th>Highest spectrum power</th>
<th>Average spectrum power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steady-state</td>
<td>3–3.005</td>
<td>0.0589</td>
<td>0.0563</td>
</tr>
<tr>
<td></td>
<td>3.495–3.5</td>
<td>0.0694</td>
<td>0.0677</td>
</tr>
<tr>
<td></td>
<td>4–4.005</td>
<td>0.0769</td>
<td>0.0754</td>
</tr>
<tr>
<td></td>
<td>5–5.005</td>
<td>0.0635</td>
<td>0.0627</td>
</tr>
<tr>
<td>External fault on terminal A</td>
<td>6–6.005</td>
<td>0.0568</td>
<td>0.0542</td>
</tr>
<tr>
<td></td>
<td>6.005–6.01</td>
<td>0.0668</td>
<td>0.0651</td>
</tr>
<tr>
<td></td>
<td>6.01–6.015</td>
<td>0.0652</td>
<td>0.0630</td>
</tr>
<tr>
<td></td>
<td>6.03–6.035</td>
<td>0.0637</td>
<td>0.0618</td>
</tr>
</tbody>
</table>

**FIGURE 4** Scalogram for current signal analysis detected on L1 T1

**4 | SIMULATION RESULT**

In the MTDC system, faulty branch needs to be discriminated from healthy lines so that the corresponding DCCB will be correctly operated to isolate the faulted part of the system. Based on the test model, internal faults including pole-to-ground faults and pole-to-pole faults and external faults including extreme AC system faults, three-phase faults at PCC are set respectively with different location, fault resistance and noises.

After obtaining the sampling data, the fault detection is simultaneously performed. To evaluate the proposed method
in a more objective way, the evaluation method used in this study is clarified in the mean absolute error (MAE) content, which is the average of the absolute errors. The actual situation of the predicted value error is better reflected. The MAE is used to evaluate and compare the results of different methods.

$$\mathcal{MAE}(X, h) = \frac{1}{m} \sum_{i=1}^{m} |h(x_i) - y_i| \quad (22)$$

In order to evaluate the optimal performance of the proposed method, the RNN-GRU and the RNN-LSTM method have been studied and compared with the RNN and BP algorithm method, which are shown in Figure 5. Subfigure (a) describes the accuracy of the proposed fault detection method during a training period, and subfigure (b) presents the loss during training BP algorithm, RNN, the RNN-LSTM and the RNN-GRU method are compared in Figure 5.

The accuracy (Figure 5a) of BP algorithm based fault detection is not satisfactory because the BP algorithm is not able to extract and utilize the time-series information in the data. The time-dependence information in fault time-series can be utilized in RNN algorithm, so the accuracy of RNN model is better than that of BP algorithm. Compared to RNN and RNN based on LSTM method, the proposed RNN-GRU based fault detection method is able to converge at a satisfactory speed. As shown in the figure, the accuracy of RNN method keeps fluctuating in the whole training period. But the situation is absolutely different in deep-GRU algorithm based method that the model becomes convergent after 70th iteration. The GRU unit can capture the long-term dependence and excave the features in time-series deeply. The fault features generated from signals can be captured and utilized more efficiently in GRU base fault detection model. Hence, compared with RNN model, both the convergence speed and accuracy is improved significantly.

A loss function (Figure 5a) is calculated on training to optimize the DL algorithm. It shows the sum of errors for each method in training sets. The losses in aforementioned three methods present absolutely different characters. The loss in the BP and RNN method keeps at a high level in the whole training period, which indicates that the training process is

<table>
<thead>
<tr>
<th>Type</th>
<th>Distance</th>
<th>Detection point</th>
<th>Highest spectrum power</th>
<th>Average spectrum power</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG</td>
<td>70% 0.01 Ω</td>
<td>L2-2p</td>
<td>0.0589</td>
<td>0.0542</td>
</tr>
<tr>
<td>PG</td>
<td>70% 500 Ω</td>
<td>L2-3n</td>
<td>0.0558</td>
<td>0.0537</td>
</tr>
<tr>
<td>PN</td>
<td>5% 0.01 Ω</td>
<td>L3-3p</td>
<td>0.0591</td>
<td>0.0579</td>
</tr>
<tr>
<td>PN</td>
<td>5% 500 Ω</td>
<td>L3-4n</td>
<td>0.0536</td>
<td>0.0525</td>
</tr>
<tr>
<td>NG</td>
<td>95% 0.01 Ω</td>
<td>L4-4p</td>
<td>0.0572</td>
<td>0.0553</td>
</tr>
<tr>
<td>NG</td>
<td>95% 500 Ω</td>
<td>L4-1n</td>
<td>0.0559</td>
<td>0.0528</td>
</tr>
<tr>
<td>PG</td>
<td>70% 0.01 Ω</td>
<td>L1-1p</td>
<td>0.0024</td>
<td>0.0019</td>
</tr>
<tr>
<td>PG</td>
<td>70% 500 Ω</td>
<td>L1-1p</td>
<td>0.0109</td>
<td>0.0087</td>
</tr>
<tr>
<td>PN</td>
<td>5% 0.01 Ω</td>
<td>L1-1p</td>
<td>0.0023</td>
<td>0.0019</td>
</tr>
<tr>
<td>PN</td>
<td>5% 500 Ω</td>
<td>L1-2n</td>
<td>0.0093</td>
<td>0.0082</td>
</tr>
<tr>
<td>NG</td>
<td>95% 0.01 Ω</td>
<td>L1-2n</td>
<td>0.0053</td>
<td>0.0023</td>
</tr>
<tr>
<td>NG</td>
<td>95% 500 Ω</td>
<td>L1-2n</td>
<td>0.0027</td>
<td>0.0048</td>
</tr>
</tbody>
</table>

Abbreviations: NG, negative-to-ground; PG, positive-to-ground; PN, positive-to-negative.

FIGURE 5 The comparison of the (a) accuracy and (b) loss of different algorithm based fault detection method. BP, back propagation; GRU, gated recurrent unit; LSTM, long short-term memory; RNN, recurrent neural network.

| Table 3 | Spectrum power for fault on L1 detection from 6.00 s to 6.01 s on normalized frequency 0.312 RAD/sample |

YANG ET AL.
unfavourable. In the deep-LSTM and deep-GRU based method, the loss both decrease of which the RNN-GRU can meet the satisfactory speed rather than the LSTM based method. In summary, the deep-GRU boosts the model training process significantly as shown in Table 4.

The final training results of the BP algorithm, RNN, RNN-LSTM and RNN-GRU algorithm based fault detection method are compared in Table 5. The performance of the proposed GRU based method is generally better than traditional methods. As can be seen in the table, the method has nearly 97.4% accuracy on average, and the worst accuracy is 97.1% in NG faults. The accuracy of the BP algorithm and RNN based fault detection is not satisfactory compared with feature extraction method, the average accuracy is only 74% and 80.4% respectively. When it refers to calculation speed, even if the LSTM and GRU methods can have accurate

**TABLE 4** Recognition rate and time spent

<table>
<thead>
<tr>
<th>Fault types</th>
<th>BP (MAE) %</th>
<th>Time (ms)</th>
<th>RNN (MAE) %</th>
<th>Time (ms)</th>
<th>LSTM (MAE) %</th>
<th>Time (ms)</th>
<th>GRU (MAE) %</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG</td>
<td>72.6</td>
<td>3.6</td>
<td>77.6</td>
<td>3.6</td>
<td>97.2</td>
<td>4.4</td>
<td>97.1</td>
<td>3.2</td>
</tr>
<tr>
<td>NG</td>
<td>74.2</td>
<td>3.7</td>
<td>84.2</td>
<td>3.4</td>
<td>96.4</td>
<td>4.6</td>
<td>98.0</td>
<td>3.1</td>
</tr>
<tr>
<td>PN</td>
<td>75.4</td>
<td>3.3</td>
<td>79.4</td>
<td>3.9</td>
<td>98.5</td>
<td>4.2</td>
<td>97.2</td>
<td>3.3</td>
</tr>
<tr>
<td>Overall</td>
<td>74.0</td>
<td>3.5</td>
<td>80.4</td>
<td>3.6</td>
<td>97.3</td>
<td>4.4</td>
<td>97.4</td>
<td>3.2</td>
</tr>
</tbody>
</table>

**TABLE 5** Fault location result

<table>
<thead>
<tr>
<th>Fault types</th>
<th>Fault position</th>
<th>Faulted line</th>
<th>Fault resistance (Ω)</th>
<th>Travelling wave (%)</th>
<th>Relative error (%)</th>
<th>BP (%)</th>
<th>Relative error (%)</th>
<th>LSTM (%)</th>
<th>Relative error (%)</th>
<th>GRU (%)</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG fault</td>
<td>7% to T1</td>
<td>Line 1</td>
<td>34</td>
<td>8.68</td>
<td>0.68</td>
<td>8.71</td>
<td>0.71</td>
<td>7.23</td>
<td>0.23</td>
<td>7.32</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td>31% to T2</td>
<td>Line 2</td>
<td>3</td>
<td>31.11</td>
<td>0.11</td>
<td>30.64</td>
<td>0.36</td>
<td>30.85</td>
<td>0.15</td>
<td>31.07</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>71% to T3</td>
<td>Line 3</td>
<td>701</td>
<td>69.12</td>
<td>1.88</td>
<td>70.46</td>
<td>0.54</td>
<td>71.13</td>
<td>0.13</td>
<td>70.23</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>94% to T4</td>
<td>Line 4</td>
<td>91</td>
<td>89.92</td>
<td>0.92</td>
<td>88.24</td>
<td>0.76</td>
<td>88.87</td>
<td>0.13</td>
<td>88.87</td>
<td>0.13</td>
</tr>
<tr>
<td>NG fault</td>
<td>4% to T2</td>
<td>Line 1</td>
<td>17</td>
<td>4.46</td>
<td>0.46</td>
<td>4.76</td>
<td>0.76</td>
<td>4.17</td>
<td>0.17</td>
<td>4.25</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>23% to T3</td>
<td>Line 2</td>
<td>950</td>
<td>25.23</td>
<td>2.23</td>
<td>23.95</td>
<td>0.95</td>
<td>22.96</td>
<td>0.04</td>
<td>22.92</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>67% to T4</td>
<td>Line 3</td>
<td>0.04</td>
<td>67.13</td>
<td>0.13</td>
<td>67.45</td>
<td>0.45</td>
<td>67.25</td>
<td>0.25</td>
<td>67.12</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>94% to T1</td>
<td>Line 4</td>
<td>105</td>
<td>93.19</td>
<td>0.81</td>
<td>93.32</td>
<td>0.68</td>
<td>93.87</td>
<td>0.13</td>
<td>93.89</td>
<td>0.11</td>
</tr>
<tr>
<td>PN fault</td>
<td>7% to T1</td>
<td>Line 1</td>
<td>251</td>
<td>8.99</td>
<td>1.99</td>
<td>7.82</td>
<td>0.82</td>
<td>31.05</td>
<td>0.05</td>
<td>7.12</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>31% to T2</td>
<td>Line 2</td>
<td>82</td>
<td>31.43</td>
<td>0.43</td>
<td>31.35</td>
<td>0.35</td>
<td>31.14</td>
<td>0.14</td>
<td>30.82</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>61% to T3</td>
<td>Line 3</td>
<td>367</td>
<td>62.79</td>
<td>1.79</td>
<td>60.23</td>
<td>0.77</td>
<td>61.73</td>
<td>0.27</td>
<td>61.15</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>93% to T4</td>
<td>Line 4</td>
<td>361</td>
<td>91.89</td>
<td>1.11</td>
<td>93.55</td>
<td>0.55</td>
<td>93.21</td>
<td>0.21</td>
<td>93.12</td>
<td>0.12</td>
</tr>
<tr>
<td>Overall</td>
<td>7% to T1</td>
<td>Line 1</td>
<td>251</td>
<td>8.99</td>
<td>1.99</td>
<td>7.82</td>
<td>0.82</td>
<td>31.05</td>
<td>0.05</td>
<td>7.12</td>
<td>0.12</td>
</tr>
</tbody>
</table>

**FIGURE 6** The comparison of stability of different algorithm (First: BP; Second: BP + feature extraction; Third: LSTM + feature extraction; Fourth: GRU + feature extraction). BP: back propagation; GRU, gated recurrent unit; LSTM, long short-term memory.
To verify the stability of the proposed fault detection algorithm, discrete verified constants are used to make the comparison more remarkable in Figure 6. It can be seen that the stability of BP algorithm based fault locator is improved significantly after the feature extraction method applied in fault locating issue. The fault features play an important role in fault location process. The LSTM and GRU algorithm further optimizes the accuracy of fault locator, as shown in the figure. Compared with BP method, not only the maximum errors, but also the medians and quartiles decreased remarkably. Based on the aforementioned discussion, the conclusion is drawn that the feature-based fault locator with GRU has a great potential to perform well in realistic scenario.

5 | CONCLUSION

This study presents a deep-RNN method with GRU for the fault detection, classification and location in a MTDC system with new multi-source information and multi-frequency components relaying technique. The proposed DL architecture based on GRU can effectively detect the fault, correctly recognize the fault types and then accurately locate the fault position with feature extraction compared with existing method.

A four-terminal MMC-based HVDC grid is developed in PSCAD/EMTDC platform under the real operation environment, and the fault detection and location method use both the features generated from local data in time domain and frequency domain effectively. The proposed method can meet the requirement of fast acting of DCCB with local data processing and is highly sensitive for high impedance fault, which shows high validity.

Compared with the existing solutions, the proposed Deep-RNN method integrated with the GRU has a better performance that the accuracy can reach to overall 99.85% both for the synthetic and real signals with noisy environment compared with the accuracy in traditional travelling wave based method and the RNN method without feature extraction. However, the method with GRU has a faster operation time which is not only satisfactory but also robust.

It should be figured out that the detailed analysis of the influence to the network at different the system parameter is not discussed in this study as we aim to use the network to extract the specific features in the proposed system. That is actually the reason why the DL method, which is insensitive to system parameters, is investigated in this study. Therefore, the proposed method can be potentially used in the other systems, for example, the point-to-point HVDC systems.

REFERENCES


How to cite this article: Yang Q, Li J, Santos R, Huang K, Igie P. Intelligent fault detection and location scheme for modular multi-level converter multi-terminal high-voltage direct current. High Voltage. 2020;1–13. https://doi.org/10.1049/hve2.12033