



Performance evaluation of machine learning for fault selection in power transmission lines

Daniel Gutierrez-Rojas¹ · Ioannis T. Christou^{2,3} · Daniel Dantas⁴ · Arun Narayanan¹ · Pedro H. J. Nardelli¹ · Yongheng Yang⁵

Received: 10 February 2021 / Revised: 10 January 2022 / Accepted: 15 January 2022
© The Author(s) 2022

Abstract

Learning methods have been increasingly used in power engineering to perform various tasks. In this paper, a fault selection procedure in double-circuit transmission lines employing different learning methods is accordingly proposed. In the proposed procedure, the discrete Fourier transform (DFT) is used to pre-process raw data from the transmission line before it is fed into the learning algorithm, which will detect and classify any fault based on a training period. The performance of different machine learning algorithms is then numerically compared through simulations. The comparison indicates that an artificial neural network (ANN) achieves remarkable accuracy of 98.47%. As a drawback, the ANN method cannot provide explainable results and is also not robust against noisy measurements. Subsequently, it is demonstrated that explainable results can be obtained with high accuracy by using rule-based learners such as the recently developed quantitative association rule mining algorithm (QARMA). The QARMA algorithm outperforms other explainable schemes, while attaining an accuracy of 98%. Besides, it was shown that QARMA leads to a very high accuracy of 97% for highly noisy data. The proposed method was also validated using data from an actual transmission line fault. In summary, the proposed two-step procedure using the DFT combined with either deep learning or rule-based algorithms can accurately and successfully perform fault selection tasks but indicating remarkable advantages of the QARMA due to its explainability and robustness against noise. Those aspects are extremely important if machine learning and other data-driven methods are to be employed in critical engineering applications.

Keywords Fault selection · Deep learning · QARMA · Power system protection

✉ Daniel Gutierrez-Rojas
Daniel.Gutierrez.Rojas@lut.fi

¹ LUT University, Yliopistonkatu 34, 53850 Lappeenranta, Finland

² The American College of Greece, Athens, Greece

³ Research and Innovation Development, NetCompany-Intrasoft, Luxembourg City, Luxembourg

⁴ The University of Manchester, Oxford Rd, Manchester M13 9PL, UK

⁵ Zhejiang University, Zheda Rd. 38, Hangzhou 310058, China

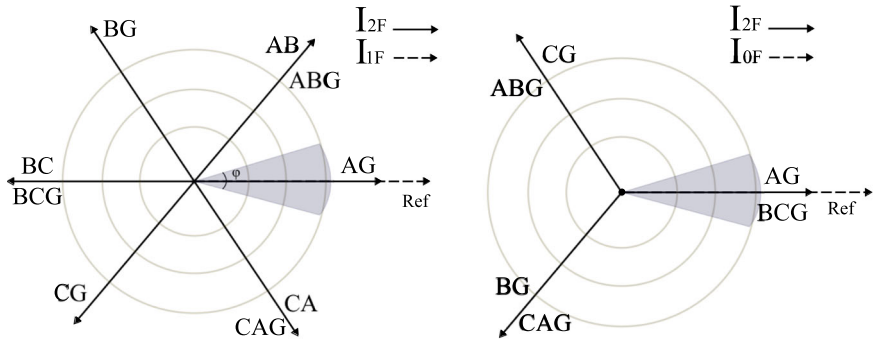
1 Introduction

Transmission lines are a fundamental part of today's power systems, as they ensure power supply to end consumers by connecting them to far-off large generation plants. Hence, it is crucial to have an adequate protective system that is capable of isolating faults quickly and reliably to prevent any possible damage to other electrical components [29]. The most commonly used device for protection of transmission lines is the distance relay, whose operation relies on the impedance between the fault location and relay installation point. Depending on the network conditions, looped segments, and double circuit lines that share towers [4], short lines, and in-feed from the other end of the line, the measured fault impedance can suffer from certain transitory variations that can cause incorrect operation of the distance relay.

Many transmission line protection schemes are used, but they do not provide intrinsic phase selection (e.g., negative-sequence and zero-sequence line differential, neutral over-current protections). However, information of the faulty phase is required to enable single-pole tripping. As any action performed by the protective system during real-time operations will directly affect the grid dynamics, correct tripping is critical to maintain the system stability and reliability [30]. Distance relaying depends on a fault selector to calculate the impedance in the loop that would lead to line triggering when the protective zone requirements are met. Therefore, a reliable distance relaying protection system for transmission networks must have a high-accuracy fault selector for correct operations in any protective zone for fast trip decision-making. In particular, faults in double-circuit and high-impedance faults rarely pose technical challenges in terms of fault selection and proper relay operation [14, 25, 32]. In addition, mutual impedance from double-circuit transmission lines may affect relay performance. When a fault to ground occurs, the zero-sequence currents from one line induce a voltage in the coupled adjacent line, thereby causing a current to flow in the opposite direction, which may add or subtract to the existing zero-sequence current [20].

Both researchers and relay manufacturers have made great efforts to improve fault classification algorithms to perform fault selection and thereby increase the system robustness. The main difficulty for selecting the correct fault is related to the effects of high resistance on the fault parameters at any given point. This leads to a situation where the fault currents are similar to each other in magnitude, and thus, their classification becomes a difficult computational task. Fault selection methods using one-end recordings can be classified according to the algorithm [27]. Following this approach, they can be divided into two broad classes: classical and emerging methods, primarily differing with respect to the balance between speed and accuracy. Some algorithms can perform the fault selection faster than the time taken by a cycle of the system frequency, but at lower accuracy. On the other hand, others perform the fault selection with high accuracy, but they lack speed and even post-protective actions, and therefore are not suitable for real-time protection and trip decision making based on the faulted loop (distance relays) [28]. These algorithms consider all measurements available, if not, approaches like shown in [21] can deal with missing values.

One remarkable example of a classical method is the symmetrical component angle comparison which checks whether the magnitude of sequence currents is sufficient to reliably perform the task by comparing with a threshold. Depending on the currents that are above the threshold, the fault is selected by comparing the angles, as illustrated in Fig. 1. In particular, either negative- and positive-sequence currents (I_{2F} and I_{1F} , respectively; see Fig. 1a), or negative- and zero-sequence (I_{2F} and I_{0F} , respectively; see Fig. 1b) currents are compared. Both cases must be consistent to perform fault selection.



(a) Angle comparison between I_{2F}, I_{1F} (b) Angle comparison between I_{2F}, I_{0F}

Fig. 1 Fault selection method based on symmetrical currents

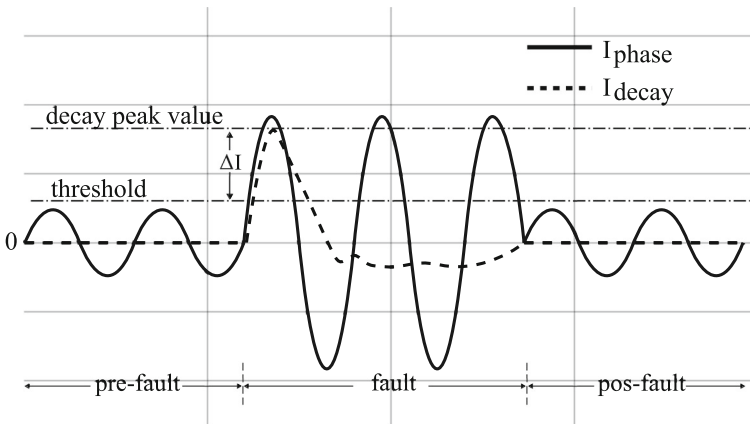


Fig. 2 Fault current from any phase and its associated decaying current

Another classical method is the so-called delta method that uses transient components from faulty continuous currents or voltage signals are used as pre-fault components. The output components employed in this method are, for example, decaying memory function (as illustrated in Fig. 2), superimposed signals, or Fourier transforms.

One of the most common classical method is the impedance-based algorithm. Its main advantage is to achieve a speed below one cycle of the system frequency, making it very popular in distance relays, being it implemented by different commercial products used for single pole tripping actions. In this method, current and voltage measurements from the fault condition are used to determine the respective zone of operation for each phase (in the case of a single phase-earth fault) or multiple phases in the loop $R - X$ diagram. These measurements are extensively used in numerical relays. Single phase-earth impedance loop characteristics for relays, such as plain impedance, quadrilateral, self-polarised mho [15], offset mho/lenticular, fully cross-polarized mho, or partially cross-polarized mho, can be defined depending on the manufacturer and system conditions.

The other class of fault classification methods and electric power applications are based on emerging computational approaches such as machine learning (ML) or deep learning (DL) [5, 7, 17, 18]. For example, in [6], the authors introduced a non-intrusive fault identification

method in power transmission lines using PS-HST to extract high-frequency fault components. A feed-forward artificial neural network (ANN) was used to select the fault classes. The authors calculated the HST coefficients and obtained a power spectrum based on the Parseval's theorem. In [1], a semi-supervised ML approach based on co-training of two classifiers is presented. The fault selection was performed in both transmission and distribution systems. Feature extraction was performed using a wavelet transform of the current and voltage signals, and a nature-inspired meta-heuristic, harmony search, was used for determining the optimal parameters of the wavelets.

Another emerging method is pattern recognition, having shown promising results compared with conventional methods. For instance in [8], a summation-Gaussian extreme learning machine (SG-ELM) was used for transmission line diagnosis, which includes fault classification and fault location, by means of an iterative back-propagation learning algorithm. In [27], an intrinsic time decomposition (ITD) algorithm was employed to analyze the frequency and time of non-stationary signals, and subsequently, a probabilistic neural network (PNN) to implement fault classification was developed. The advantage of this approach lies in its training speed that enables the entire process to be performed in real time. A power-spectrum-based hyperbolic S-transform (PS-HST) and back-propagation artificial neural network (ANN) were used in [6] to extract high-frequency components of the electric signal generated by an electric fault to improve fault selection coefficients, and fault classes in power transmission networks were then identified with one-end recordings. Three ML models—naive Bayes classifier, support vector (SV), and extreme learning machine—were compared in [26] for fault classification based on the Hilbert–Huang Transform.

Hybrid techniques can also play an important role. Control strategies that involve two or more of the methods described above can be used to increase the reliability and accuracy of fault selection. New numerical relays (with higher processing capabilities) are often employed to effectively select the proper fault and avoid undesirable tripping. Strategies based on ML use classic methods for pre-processing to improve their models [6].

Here, we will give particular attention to the quantitative association rule mining algorithm (QARMA), which has not yet been employed for the fault selection in transmission lines. QARMA has already been tested in several application scenarios and use-cases in the health domain and in predictive maintenance applications in particular (see [10, 11] for results relating to predicting tool Remaining Useful Life in the auto-motive manufacturing industry from the recently concluded PROPHECY project). Within the context of the EU-funded QU4LITY¹ project, QARMA results have been tested against real-world data-sets ranging from tool wear-and-tear to body measurements to compute morphotype fit scores in the fashion industry.

The main reason therefore for choosing QARMA as a tool to study its applicability in the given domain is the success that QARMA-based classifiers and regressors obtained in such varied domains, and given the natural appeal that the output of QARMA offers, in the form of easy-to-understand rules, that we consider are more directly explainable than higher-order approaches to explainability/interpretability such as Shapley values for explaining otherwise black-box models. Still, we compare our main two approaches, namely deep neural networks and QARMA to several other well-known classification algorithms, see Sect. 4.2. The main criteria for the choice of these other algorithms were their prior use in this domain as established in the literature, their overall popularity in the ML field in general, as established by the number of results returned in Google Search for the respective terms, and finally, their explainability/interpretability.

¹ <https://qu4lity-project.eu/>.

This paper extends the above contributions by proposing a two-stage method. The first stage is the delta method discrete Fourier transform (DM-DFT) that is used to pre-process the raw data from the transmission. The second stage performs a machine learning algorithm for fault selection. We studied different techniques in terms of accuracy and explainability. Our main contributions, also presented in Sect. 4, are as follows:

- We propose a general hybrid methodology based on DM-DFT algorithm that works independent of network topology.
- We test and compare the performance of well-known ML algorithms techniques such as decision trees, neural networks, and support vector machines (SVM).
- We develop a fully explainable method that employs the quantitative association rule mining algorithm (QARMA) [12, 13] and compare its performance with the state-of-the-art ML algorithms (mostly not-explainable).
- We demonstrate with several numerical examples, including real-world data that fault classification task can be solved by QARMA with very high accuracy even when only one-end currents are available, or when the measurements are subject to high levels of noise.

The rest of this paper is divided as follows. Section 2 introduces proposed methodology. Section 3 details the machine learning algorithms employed here, including a detailed description of QARMA. Section 4 presents the numerical results, while Sect. 5 concludes this paper.

2 Proposed method

2.1 Step 1: delta method discrete Fourier transform

To extract fault features (currents and voltages), a combined DM-DFT is employed to identify the fault instance. The DFT maps a given point of the input signal (i.e., current or voltage) into two points in the output signal. For N samples, considering the pair x_n (input signal) and X_k (its DFT)

$$X_k = \sum_{n=0}^{T-1} x_n e^{-2\pi i kn/T}, \quad (1)$$

where $0 \leq k \leq T - 1$, and T is the number of samples per cycle and n represents current phase (a, b or c). The DM-DFT uses a moving window of length T instead of the complete signal, thus allowing faster fault recognition. The fault point requires to be within a fault time, which is considered here to be approximately 3.5 cycles. The sampling rate is 4 kHz, i.e., about 80 samples per cycle (which is usually used in commercial relays). To obtain a highly accurate signal point 1.5 cycles or about 120 samples after fault occurrence are needed. When a transmission line is in a faulty state, magnitudes of currents and voltages (which are the features used in the fault classification task) can suddenly change depending on the type of fault and its characteristics. Figure 3 illustrates a cycle in the periodic sinusoidal signal where the DFT calculations are performed. Once the DFT is calculated, variations in the frequency domain can be detected as follows:

$$\text{threshold} = 1.5I_{n(j)}, \quad (2)$$

$$\Delta I_n = I_{n(j+T)} - \text{threshold}, \quad (3)$$

$$F_i = j + T \leftrightarrow \Delta I \geq 0 \quad (4)$$

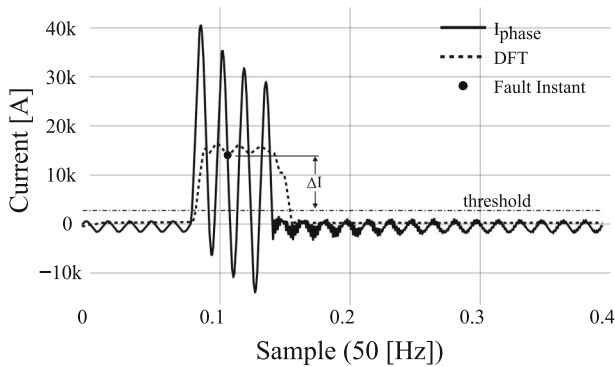


Fig. 3 DM-DFT cycle calculation

where “threshold” refers to the current signal; ΔI_n are the changes in the current signal; $I_n(j)$ is the current Fourier value of the j th sample with $0 \leq j \leq S - T$; S is the total number of samples in the signal; and F_i indicates the fault instance. Threshold for the ongoing signal calculation is given by 1.5 times the Fourier pre-fault signal value. A different threshold value is selected based on the experimental results for different fault conditions. The DM-DFT is applied to the three-phase currents without considering if ΔI_n have similar values; it only considers values above the threshold. If multiple ΔI_n are positive, the fault instant is chosen from the phase n that has the highest value. If $\Delta I_n \leq 0$ for all three phases, then it is assumed that there is no fault and the feature extraction is taken randomly from one of the samples of each signal. Delta methods can be seen as detection of high fluctuation of any quantity (like temperature, current, or even monetary value) and is only used to identify faulted point and then method continues with feature extraction. At this stage, it is possible to miss fault points (miscalculation of a given phasor due to wrong time series point) because of threshold values. However, for the data-set obtained in this process, all the faults were detected successfully.

Note that the DFT phasor estimation is less sensitive to noise than the individual measurements, and it is robust to the presence of harmonics [23]. Also the threshold selection can perform even if parallel lines are out of service or if it is applied to transmission lines with different parameters or ratings [16]. That means the threshold is independent of the topology and geometry of the structure. Traditional protection relays use the DFT for protection calculations [23], therefore using the same pre-processing technique to minimize hardware requirements, while providing sufficient information to the neural network. However, the DFT is dependent on the sample frequency, which might be problematic for real-time applications because of the computational time limitations. It means, the DFT calculation might take longer than 20 ms to calculate, which is the time where the trip decision is made in a real-time transmission line scenario.

The process is applied to data-sets (like the one obtained from simulations to be explained in Sect. 4.1) that contain currents and voltages, either with or without a faulted state. The selection is automatically done after DFT procedure is completed. For data-sets with faults, the voltages and current feature extraction are selected at the fault instant point; for the ones without faults, a random point within the signal is selected. The output data-set, listed in Table 1, is the input for the ML methods to be described next. Table 1 contains the absolute values of currents and voltages of local and remote ends of the transmission line. The neutral currents were estimated as a phasorial sum of the abc currents.

Table 1 Output from DM-DFT including all datasets with and without faults

Feature	File 1	File 2	...	Last file
$L_{IA}, L_{IB}, L_{IC} \dots R_{VC}$	Value	Value	...	Value
Target variable				
Fault type	Value	Value	...	Value

2.2 Step 2: machine learning methods

A number of different established algorithms are then considered for the supervised learning: decision trees, artificial neural networks (both shallow and deep), support vector machines, rule-extraction systems (Ripper-k and QARMA), naive Bayes, logistic regression, and finally ensemble methods (AdaBoost). As already mentioned, the main criteria for selecting the above methods were their prior use in the domain, as established in the current literature, their popularity in the machine learning field, and their explainability/interpretability properties.

These algorithms have different accuracy levels and time to train each model. Moreover, the “explainability” of their models also varies. For example, explainable methods, such as decision trees, usually have poorer accuracy than ANNs. On the other hand, when the training data-set is large enough, an ANN often gives very high accuracy, but it is time consuming, and the resulting model offers little in terms of explainability to humans. The algorithm should then be selected depending on the requirements set between accuracy, time, as well as explainability of the outcome.

In this paper, the focus is on representatives from the class of DL methods and the “explainable artificial intelligence (AI)” families—for an exposition to the latter class, see [22]. We built and tested models with multiple hidden layers of feed-forward nodes trained by mini-batch-based optimization methods (including classical stochastic gradient descent with momentum as well as the Adam [24] optimizer); we also built rule sets extracted using the QARMA algorithm for quantitative association rule mining [13]. These choices were made because of the proven capability of DL methods to obtain very high accuracy given enough data, and also because QARMA is an algorithm that has already been successfully tested for Predictive Maintenance (PdM) related tasks in industrial settings. The output data-set from the DM-DFT is used in all cases, and it contains all the produced features. The target variable, i.e., the fault type (see Table 1), is encoded into eleven different classes (ten fault types plus one no_fault mode) as it has string values (one-hot encoding). Figure 4 presents the flowchart of the proposed two-step method. The DM-DFT is employed for all cases as the preprocessing stage, while the second stage is the different ML algorithms presented here.

3 Selected learning algorithms

This section starts with the algorithm that is expected to have the highest accuracy: the ANN as proposed in [9]. Then, the QARMA algorithm [12] is presented in brief, as it is expected to provide reasonably high accuracy but with the added benefit of explainable outcomes.

3.1 Artificial neural networks

Artificial neural networks (ANN) and in particular feed-forward ANNs, also known as multi-layer perceptrons, are a powerful ML tool and have been used extensively for fault diagnosis

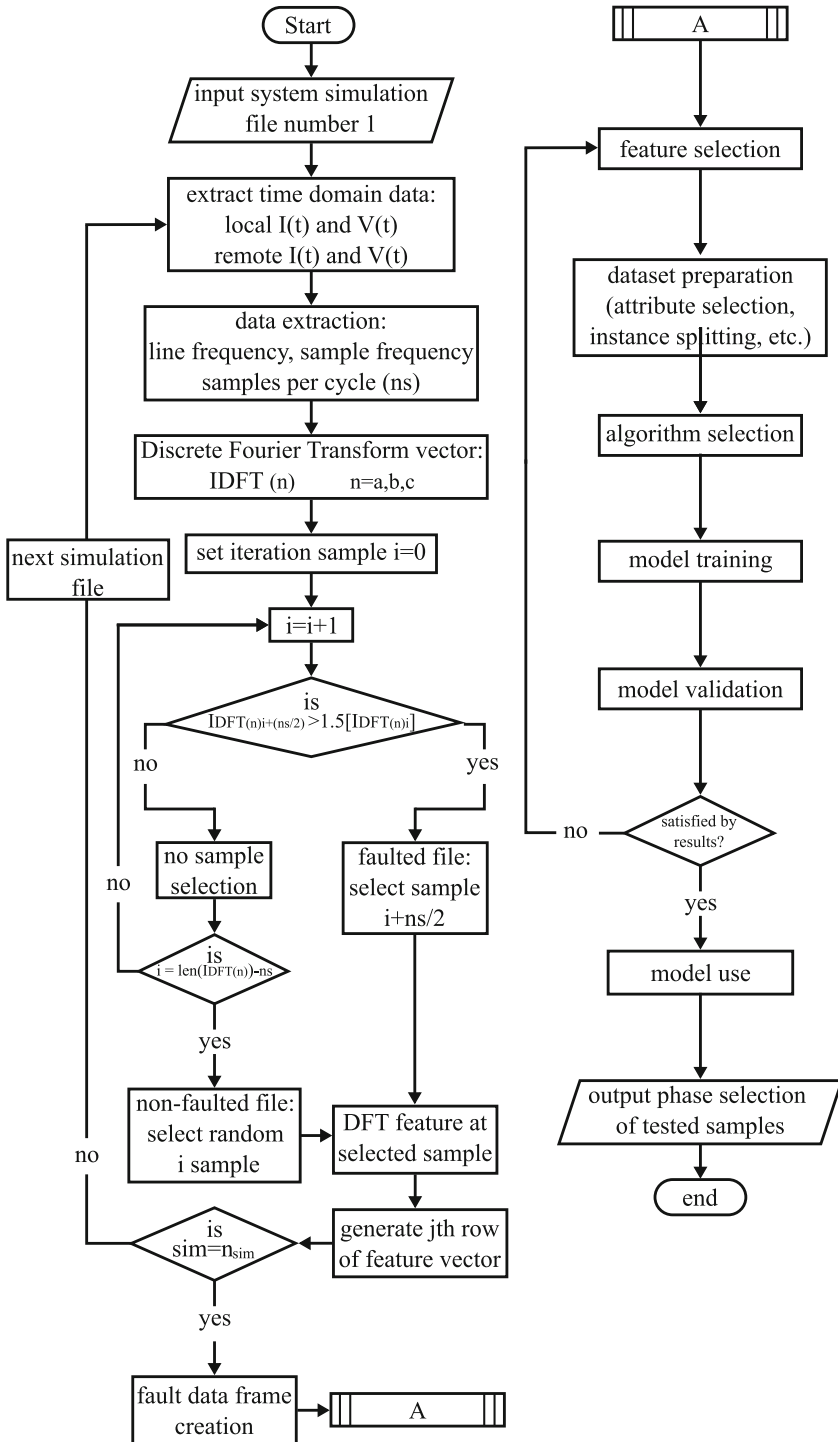


Fig. 4 Flowchart of the fault classification process using ML

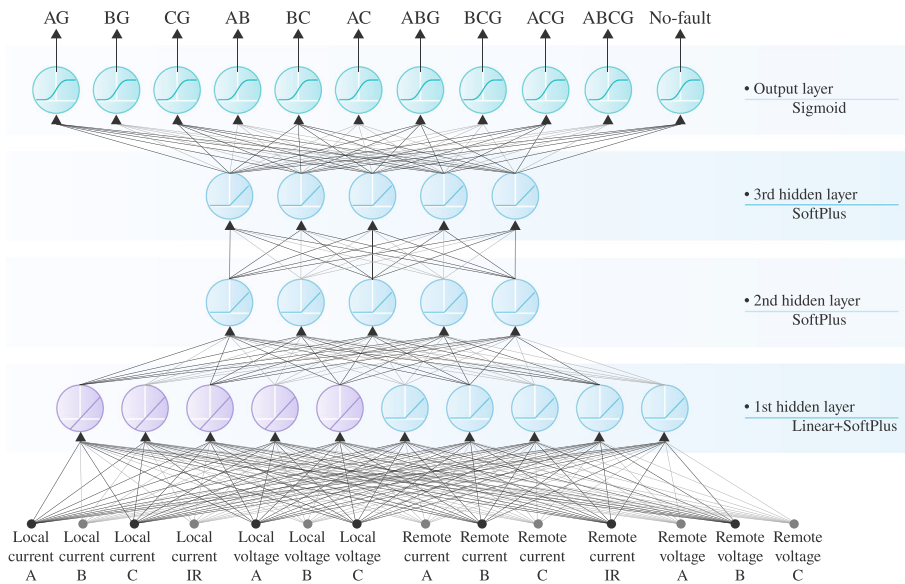


Fig. 5 Structure of a three-stages feed-forward neural network

problems such as the ones mentioned before. The ANN is a feed-forward neural network consisting of three stages. The first stage is the input layer containing the voltages and currents from both ends at the time of fault occurrence given by the DM-DFT along with a fault tag coded into binary form. The second stage is the set of hidden layers, where every node in a particular layer receives inputs from all the nodes in the layer immediately below that layer and sends its output to all nodes in the layer immediately above it. We have experimented with various architectures, shallow and deep, using the Open-Source library `popt4jlib` (<https://github.com/ioannischristou/popt4jlib>) that allows for parallel and distributed evaluation of training instance pairs of both the network output as well as the gradient of the network computed via the classical back-propagation algorithm. The third and final stage is the output layer that returns fault type (or no_fault) signals that are encoded back into the phase selection tag. Figure 5 illustrates the procedure, where “Local current A” refers to the current signal of phase A measured at the left end of the transmission line (see Fig. 6), while “Remote Current A” refers to the current signal of phase A measured at the right end of the transmission line and subsequently with voltages and phase B and C. Together, they form the features of the data-set. The multi-layer ANN in [9] has the following parameters: two fully connected layers with a rectifier linear unit (ReLU) activation, one output layer with *softmax* activation, categorical cross-entropy loss function, and *adam* optimizer.

In our experiments, the best topology was achieved with a deeper network consisting of 4 layers in total: the first hidden layer consisting of a mixture of 5 linear activation units and 5 SoftPlus activation units (smoother version of ReLU), and the other two hidden layers consisting of 5 SoftPlus activation units each. The output layer, using one-hot encoding, comprised of 11 sigmoid (logistic) activation units, each one corresponding to one of the possible classification results for the problem (10 different fault types and one no_fault type.)

3.2 Quantitative association rule mining for fault diagnosis

Association rule mining (ARM) is a major and still very active research area; implementations of the algorithms developed over the years are found in most popular software packages for data mining, such as WEKA, MOA, KEEL, and Orange. ARM works on datasets that contain subsets of “items.” A typical dataset applicable for ARM is a database containing super-market basket data, i.e., the items in customers’ shopping carts during check-out. Its major objective is to discover statistical rules that relate the presence of a set of such items to the presence of other items, and a typical association rule for such market basket data would be $Buys(“Milk”) \implies Buys(“Bread”)$ where the implication is understood to hold in a statistical sense, so that the rule means that the percentage of baskets that contain both milk and bread is above a minimum threshold (support of the rule) as well as that the ratio of all baskets that contain both milk and bread over the number of baskets that contain at least milk is above another threshold (confidence of the rule.) The a priori [3] algorithm is a famous early algorithm for discovering all such rules satisfying minimum support and confidence in a given dataset. In the following years, many different authors improved upon this first algorithm (see [19] for a notable example).

However, the above notion of association rules is a “qualitative” one: any possible quantitative attribute belonging to the items is not taken into account. Quantitative association rule mining (QARM) is an extension of the standard ARM that allows for items to quantify any attributes they may have in the rule antecedents and/or consequences, for more precise rules.

An illustrative example of a quantitative association rule would then be $Buys(Milk).price \leq 0.9 \wedge Buys(Bread).price \leq 0.25 \implies Buys(Sugar).price \leq 0.1$ which says that (for a percentage of customers above the specified support) customers who buy milk at a price less than or equal to $USD\$0.9$ and bread at a price less than or equal to $USD\$0.25$ will also purchase sugar at a price less than or equal to $USD\$0.1$. This is significantly more information than simply knowing that when a customer buys bread and milk they are also likely to buy sugar.

QARMA [12, 13] is a family of efficient novel cluster-parallel algorithms for mining quantitative association rules with a single consequent item, and many antecedent items with different attributes in large multidimensional datasets. Using the standard support-confidence framework of qualitative association rule mining [2], it extends the notions of support, confidence, and many other “interestingness” metrics so that they apply to quantitative rules.

QARMA is configured to produce rules of the form $I_1.attr_1 \in [l_{1,1}, h_{1,1}] \wedge \dots \wedge I_n.attr_m \in [l_{n,m}, h_{n,m}] \implies J_0.p \in [l_0, h_0]$ or alternatively to produce rules of the form: $I_1.attr_1 \in [l_{1,1}, h_{1,1}] \wedge \dots \wedge I_n.attr_m \in [l_{n,m}, h_{n,m}] \implies J_0.p = v$. The latter form is very useful in supervised classification problems where the value of the target item attribute is essentially the class variable that is being learned.

QARMA (fully specified in [13], and then extended in [12]) within the particular context of grid fault diagnosis, works as follows:

First, all subsets of variables including the target variable (fault indicator) of length 2, then 3, then 4, up to a user-specified length are constructed, and called “itemsets.” Then, the algorithm proceeds sequentially to produce all valid quantitative association rules from each itemset of length 2, then 3, then 4... Within each phase of producing all valid rules of length $l = 2, 3, \dots$, the algorithm considers in parallel all frequent itemsets of length l . For a given itemset, it produces all possible rules (with each attribute in the rule being un-quantified in the beginning); for each such initially unquantified rule, a possibly different CPU core runs a procedure called $QUANTIFY_RULE()$ maintaining a local rule set R (initially empty) and runs a modified breadth-first Search procedure that first assigns the consequent attribute

to the highest possible value, and, as long as the resultant partially quantified rule has support above the threshold required, adds it to a queue data structure T .

While this queue is not empty, the first rule inserted in the queue is retrieved and removed from the queue. For each attribute that has not been quantified in it yet, the algorithm creates as many new rules as there are different values in the dataset for the attribute being examined in an ascending attribute order value and enters the queue T in this order, but only if the newly quantified rule exceeds the minimum support requirement. If the partially quantified rule also meets minimum confidence (or any other metric), then it is checked against the current set of local rules R to see if it is dominated by another rule in R . If no other rule in R dominates the current rule, the current rule is added to the set R . After having run this BFS process in parallel for all frequent itemsets of length l , the various CPUs participating in the run synchronize to obtain all rules from all the other ones before moving to process the frequent itemsets of length $l + 1$.

The resulting rule set has the theoretical property that it maximally covers the dataset it has worked on: there is no other rule outside the produced dataset in the form described above that can cover even a single extra instance in the dataset while having the required minimum support and confidence (or other specified interestingness metrics) required. Once the set of all non-dominated rules has been computed, a classifier based on their ensemble works as follows:

1. Select all the rules whose antecedent conditions are satisfied by this instance and add them to the set F ;
2. Sort out the rule-set F in decreasing order of confidence and decreasing order of support on the training set;
3. Remove all but the top-100 rules of the sorted set F ;
4. Each rule in F carries a weight equal to its confidence on the training set;
5. The weighted majority vote of the rules in F decides the class of the instance.

4 Simulation results

4.1 Test system description

A 400-kV, 50-Hz power system (Fig. 6) was simulated to extract features and then generate the dataset of currents and voltages based on the DFT at a fault point (when there is a fault). Under this setting, 10 different faults can occur involving the electrical phases A , B or C and ground G of the transmission line: three-phase faults (ABCG), bi-phase faults (ABG, BCG, CAG, AB, BC and CA) and mono-phase faults (AG, BG and CG). They differ from each other due to the phases involved and their parameters. The electrical system under study is composed of a double-circuit transmission line typical, for example, in Finland and other European countries. It has two lines connected to a local end marked as L and remote end R. At each end, a source is connected representing a transmission network. These types of lines represent a challenge for correct fault identification and selection owing to the strong impact of mutual impedance on the fault resistance. As for the communication channel, data were gathered by intelligent electronic devices (IED) from both ends and sent via a wireless link (e.g., 4G or 5G) to the fault selector, as shown in Fig. 6. It also presents the data flow blocks how the fault selection is performed and retrieved back to the smart devices for protective actions. The training and testing data-sets were collected in the preprocessing phase.

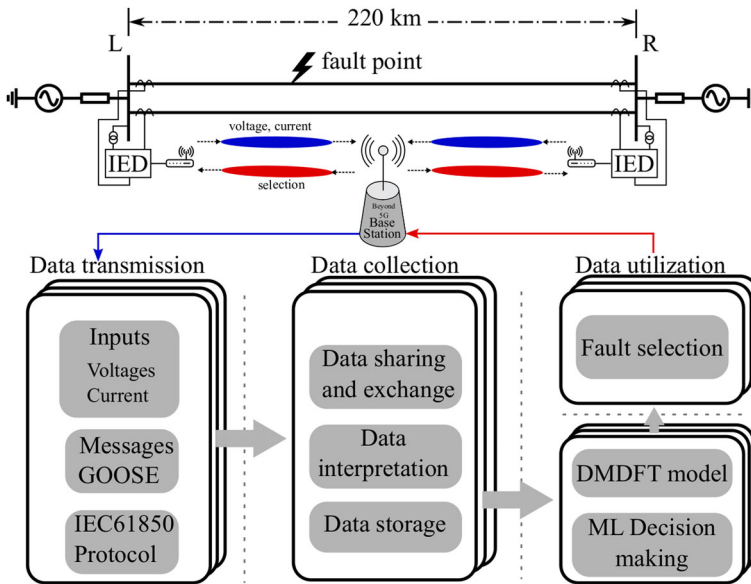


Fig. 6 A 400-kV double-circuit transmission line and data flow

Table 2 Simulation input parameters

Parameter	Training data set	Testing data set
Fault type	None, AG, BG, CG, ABG, BCG, CAG, AB, BC, CA, ABC	None, AG, BG, CG, ABG, BCG, CAG, AB, BC, CA, ABC
Fault resistance (Ω)	0.01, 0.1, 1, 5 ^a	Random
Fault distance (%)	10–90 (steps of 10)	Random
Fault inception angles	2 (45° and 90°)	Random
Power flow variation	2	Random
Line parameter error	5	Random
Total size	15,120	5040

^aFault resistances from 10–200 Ω (steps of 10 Ω) are only applied for single-phase faults

All the simulations were carried out in MATLAB/Simulink. The simulations were prepared with the specifications shown in Table 2, the transmission line parameters in Table 3. Both normal operations and different fault types (10 in total) were simulated along with different fault resistances (24), fault inception angles (2), line parameter errors (5), high and low power flow (2), and fault locations along the line (9). The simulation comprised 20160 rounds to collect data of both fault and non-faulted systems whose details are presented in Table 2. The resulting data-set is already publicly available.²

² https://github.com/ioannischristou/popt4jlib/tree/master/testdata/grid_fault.

Table 3 Transmission line parameters

Parameter	Transmission line L–R
Voltage (kV)	400
Length (km)	220
Positive-sequence resistance (Ω/km)	0.0033564
Positive-sequence inductance (H/km)	0.00057347
Positive-sequence capacitance (F/km)	$2.0423e^{-8}$
Zero-sequence resistance (Ω/km)	0.27073
Zero-sequence inductance (H/km)	0.0039052
Zero-sequence capacitance (F/km)	$7.9939e^{-9}$

4.2 Results

Two simulation scenarios and a real fault from a transmission line were used to test the proposed methodology. Note that, for these experiments, all machine learning algorithms ran on the same machine. However, the proposed implementations are fully parallel and do take advantage of all CPU cores available in the computer running the codes. This makes it more computation-efficient. Specifically, QARMA does not require any hyper-parameters to run.

This is not the case for the ANN, though, for which, the architecture (number of layers, number of nodes in each layer, type of each node and so on) must be specified in advance, and forms the set of hyper-parameters that need to be fine-tuned through experimentation and best-practice guidance.

Nevertheless, it is worth mentioning that we do not claim that the parameters of our ANN model are optimal, as they were found by manual search in repeated experiments; they only provided excellent accuracy, and it is only this best set of results for the ANN that we report in this paper. Further, regarding the hyper-parameters required for the other classification algorithms that we experimented with, Naive Bayes requires no hyper-parameters, Ripper-k requires only the number of FOLD iterations which is by default set to 2, Decision Trees, Logistic Regression and Support Vector Machines are famous for requiring very few hyper-parameters (gain criterion function, and penalty factors “w” and “C,” respectively); finally, for the AdaBoost.M1 method, that does require the base weak learners to be fully specified, we left the default settings specified in the WEKA package.

Besides, all simulation scenarios were based on typical topologies and parameters used in the specialized literature, which represent real transmission lines and their operation.

4.2.1 Test system 1

In the first test system, the generated data-set was split into two subsets: 75% of a random shuffle of the data-set was kept for training and the remaining 25% was used to validate the accuracy of the trained models. The exact same split was used for all simulations with all different algorithms. Experiments with fivefold cross-validation gave essentially identical results. Table 4 shows the results of running the above-mentioned algorithms for supervised learning on the produced data-set; Fig. 7 shows the results for the classification task. The accuracy achieved with the DL model setup was remarkably high, 98.33%. It was achieved by a 4-layer deep network, with 10 nodes in the first hidden layer (5 linear and 5 SoftPlus units), 5 SoftPlus units in the second layer, and 5 SoftPlus units in the third layer; the output layer had 11 sigmoid units corresponding to each of the 11 fault class types (including the

Table 4 ML results on the fault-grid dataset

Classifier	Accuracy (%)
Decision tree	94.62
ANN (1 hidden layer)	95.18
ANN (3 hidden layer)	98.33
SVM	89.05
Ripper-k	86.17
Naïve Bayes	59.42
Logistic regression	78.47
AdaBoost.M1	17.81
QARMA	98

Bold indicates the two best accuracy methods

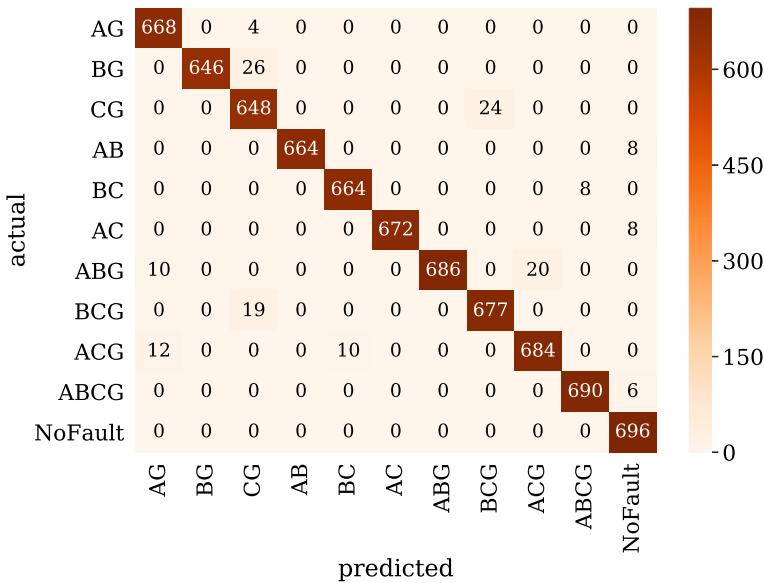


Fig. 7 Fault classification task confusion matrix for test system 1

“no-fault” type). This particular architecture was determined via trial-and-error, as the best observed among 50 different architectures. The total cost function of the network was the sum of square errors of each output node over all training instances. The entire network was trained via stochastic gradient descent (SGD) as the weights optimization algorithm. The backpropagation algorithm was used to compute the overall function gradient (derivatives corresponding to each data instance within a batch computed in parallel and then summed together to form the total gradient). The open-source library pop4jlib (<https://github.com/ioannischristou/pop4jlib>) was used to train this network, and it also contains the simulation datasets used in this paper. Note that simpler methods such as Naive Bayes or logistic regression did not perform well on this dataset. This happens because the relatively deep neural network employed here has enough layers to produce an intermediate representation that makes it easy for the final layer to classify correctly the 11 different classes; and that

Table 5 Feature selection on original data-set

Round	Feature
1	All features (local and remote current and voltages including I_R)
2	$L_{I_A}, L_{I_B}, L_{I_C}, L_{I_R}$
3	$R_{I_A}, R_{I_B}, R_{I_C}, R_{I_R}$
4	$L_{I_A}, L_{I_B}, L_{I_C}, L_{V_A}, L_{V_B}, L_{V_C}$
5	$R_{I_A}, R_{I_B}, R_{I_C}, R_{V_A}, R_{V_B}, R_{V_C}$
6	$L_{I_A}, L_{I_B}, L_{I_C}$
7	$R_{I_A}, R_{I_B}, R_{I_C}$
8	$L_{V_A}, L_{V_B}, L_{V_C}$
9	$R_{V_A}, R_{V_B}, R_{V_C}$

the large number of produced high-confidence rules leads to majority votes that are usually correctly predicting the fault.

The SGD method was used with a mini-batch size of 50 instances. In addition, normalization of the gradient vector $g(w) = \nabla E(w)$ to 1 before the steepest gradient descent rule $w \leftarrow w - \alpha g(w)$ was important for quick convergence; the learning rate α decayed as the epochs progressed according to the formula $\alpha \leftarrow \alpha 500 / (500 + \text{epoch})$. The remarkable validation accuracy was achieved after only 10 epochs in less than 8.6 s of wall-clock training time on an Intel i-9 10920X processor, using all its 24 logical cores. This high accuracy is due to the large size of the simulated fault dataset and equally importantly because of the *balance between the sample sizes of the various classes*. The strong success of the DL model is also because all the voltages and currents from both lines were available, including neutral currents. The faults that were not selected properly were all single-phase-to-ground faults. This can be explained as follows: in those fault cases where the fault resistance took the larger value, only one of the phases changed slightly compared to the other two phases making the feature variation difficult for the model to detect. Further, perfect communication and without any problems related to latency, availability, or synchronization was considered.

With this setup, the importance of availability of all features was tested. Table 5 lists the number of features tested, and Fig. 8 shows the results with an ANN.

With fewer features, the ANN does not perform as well, emphasizing the importance of neutral current estimation. However, when only one-end currents are available, the validation error of the algorithm is still adequate for the task.

We also ran an experiment to test the sensitivity of the neural network to measurement noise; we progressively added more Gaussian white noise (with zero mean and increasing sigma values) to each of the features in our training and/or test data except for the class attribute (fault type.) The results are tabulated in Table 6 and show that for small σ values less than 10, the trained model is still able to classify test data with nearly the same accuracy as when there is no noise in the measurements; however, for large $\sigma = 100$, the neural network accuracy drops significantly, to around 89% which indicates that the trained model is no longer able to accurately identify fault types when measurement noise reaches such high levels. The situation is the same or worse when the training data-set itself suffers from measurement noise: when the training data-set is “polluted” with white noise with small $\sigma = 10$, even when the test data have no noise at all, the accuracy of the trained model

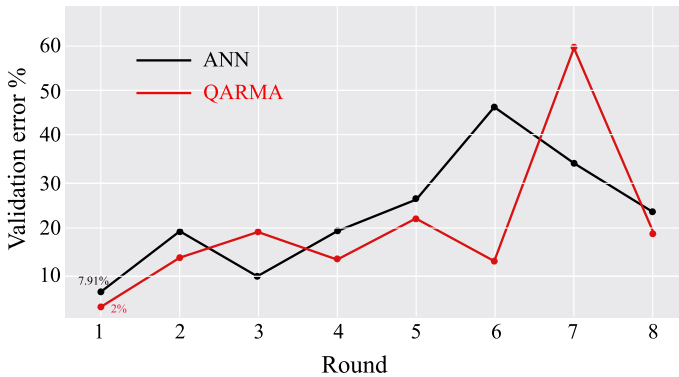


Fig. 8 Validation error obtained with same model parameters using less features

Table 6 Neural network performance under different noise levels in the data

Training σ	Testing σ	NN error%	QARMA error%
0	0.01	2.98	2
0	1	3	2
0	10	3.14	2.01
0	100	11.7	2.23
0.01	0	3.04	2
1	0	3.04	2
10	0	6.1	2.01
100	0	17.84	2.47
100	100	20.67	2.89

drops to less than 94%. When both training and test data are “polluted” with white noise with $\sigma = 100$, the neural network accuracy drops to less than 80%.

We ran QARMA on the same training set with the user-defined support threshold of 3.5%, and the confidence threshold of 90% to obtain 5333 rules covering 97.8% of the entire training set. Then, a slight variant of the decision making algorithm described in the previous section, based on weighted voting, was used: for each instance in our test set, as long as the instance is covered by more than 100 rules, the instance’s class is decided upon by the majority vote of the *top 10 firing rules having the highest confidence on the training set; instances that fail the minimum coverage requirement are not classified*. This algorithm resulted in high accuracy comparable with the one obtained by the DL, around 98% but at the following cost: a longer training time (around 15 min of wall clock time on the same i-9 10920X CPU with 24 logical cores.) For a small percentage of testing instances, approximately 4%, QARMA was not able to provide a decision, because of the small number of rules firing on them. However, we expect that QARMA and its decision-making components will compare equally well or even outperform deep learning techniques in training sets that are more highly skewed.

Another advantage of QARMA relates to the sensitivity of the produced rules with respect to noise in the data. We already saw that when the training and testing data suffer from Gaussian white noise with $\sigma = 100$ the performance of the neural network drops just below 80%. On the other hand, when QARMA ran on the same noise-polluted training dataset with $\sigma = 100$, and then the resulting rule-ensemble asked to classify an equally noise-polluted

testing data-set (with $\sigma = 100$), surprisingly, QARMA performance remained very high at 97.11% making QARMA much more robust to noise in the measurements than the neural network. QARMA performance is then very little affected by noise in absolute terms, ranging from 2% of error in the best studied case (first row of Table 6) to 2.9% in the worst (last row).

Even though more research and experiments are needed to fully explain why this might happen, we believe that the answer about the cause of the difference of robustness between the two classifiers is probably lying on the underlying models' complexity: the NN being a deeply composite function of many variables (connection weights and bias thresholds) when optimized on a noise-polluted training data-set is easier to over-fit, and "learn" some of the noise in its weights. On the other hand, QARMA being a rule extractor that learns rules that have only a small number of different features in their antecedent conditions provides an ensemble of simple if-then decision rules that are more likely to hold true in the presence of noise.

Besides, QARMA produces a model with a set of quantitative rules that are much easier to understand and reason about than most of the other models, and DL models in particular; this makes QARMA results much easier to explain to humans than any other model. Every extracted rule is trivially checked against the training data-set for validation purposes, and it is also trivial to understand "what it means" since the preconditions of the rule are nothing more than a conjunction of the restrictions of the attributes that comprise the rule's antecedents to certain intervals. This ease of understanding of rules is what has made them particularly attractive since the beginning of AI and ML research. In fact, already since the 1980s, there have been attempts to extract the knowledge that is embedded in neural network models into sets of rules [31] since such rule sets were recognized from the beginning as the most obvious knowledge representation that can exist. Therefore, QARMA is, in general, a particularly good fit for the newly emerging "eXplainable Artificial Intelligence" (XAI) paradigm, the term "explainable" meaning that the resulting model that the algorithm produces can be easily understood by humans.

4.2.2 Test system 2

A different line configuration was also tested in order to evaluate the generalization capabilities. The test system 2 consists of a single-circuit 400-kV transmission line connected to two Thevenin equivalents. Although this is a simpler system that does not have the same impact of mutual impedance of double-circuit transmission lines, the inclusion of this simulation in the data-set allows the analysis of generalizing the solution to different systems. A data-set containing 990 rows was used for testing the original model; the ANN achieved an accuracy of 98.8% while QARMA achieved 98.1% for all faulty classes and non-faults in this system. The results were slightly better than the test performed on the original data-set, showing the viability of DMFT for fault detection and the ANN/QARMA for classification. The confusion matrix of the mentioned test can be seen in Fig. 9.

We also ran a symmetrical algorithm method using the model data-set for this paper. This method is used as the basis for comparison of our proposed approach because it is employed by one top relay manufacturer. The results can be seen in Fig. 10 (note that a confusion matrix like the one presented in Fig. 9 cannot be used for comparing all faults using the symmetrical method because the datasets have different lengths; only AC, BC and CG faults can be compared in that way). The accuracy of this method for single-phase faults can be represented in Table 7 along with the false positive single-phase detection. False positive in this context is seen as the number of single-phase faults selected by the symmetrical method

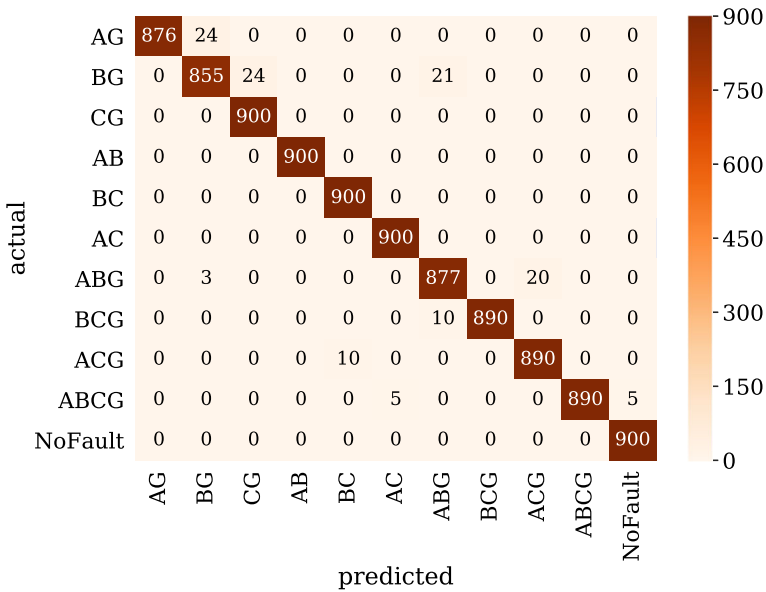


Fig. 9 Fault classification task confusion matrix for test system 2

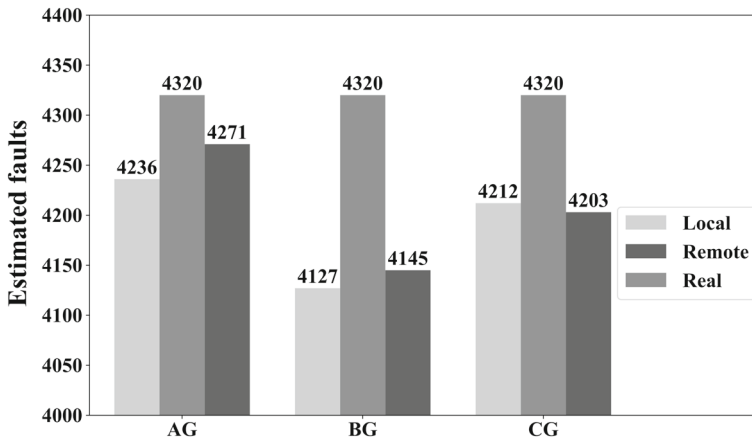


Fig. 10 Fault classification by symmetrical angle method

given that the real fault involved at least 2 phases. Under those conditions, the classification strategy takes the system to a situation less secure than with a tripolar tripping.

In summary, the results shown in Fig. 7 indicate that the errors in the proposed method occurred in the form of lack of identification of some faults. Since fault selection systems are meant to be associated with protection algorithms, those errors can cause an unnecessary tripolar breaker opening—security error. Considering an interconnected system, security errors are less likely to cause system-wide power outages than protection dependability errors. Therefore, in comparison with the symmetric method, the proposed solution will promote better system stability than the traditional method’s results depicted in Table 7.

Table 7 Results obtained by replicating symmetrical method

	Dependability		Security	
	Local end (%)	Remote end (%)	Local end	Remote end
AG	98.05	98.86	910	910
BG	95.53	95.94	896	909
CG	97.50	97.29	880	880

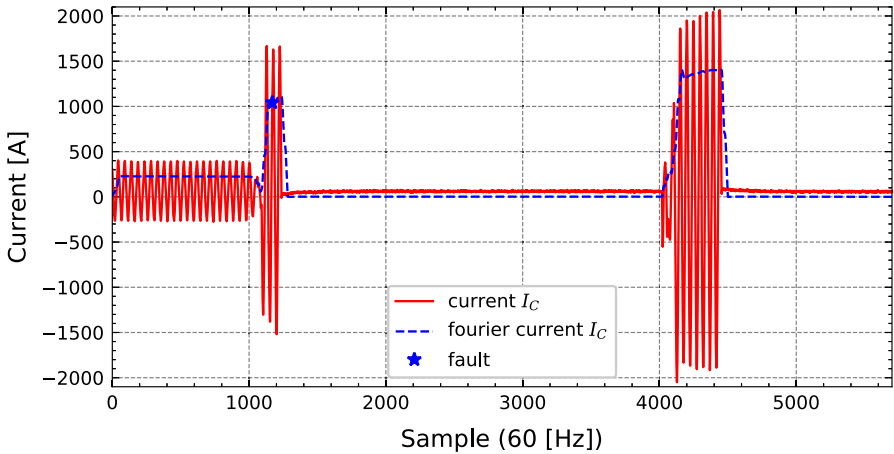


Fig. 11 Phase C and DFT of real transmission system fault

4.2.3 Real fault file

To test the proposed procedure, we used a real fault file from a transmission system located in Brazil, whose exact location cannot be disclosed. Real faults are usually gathered by fault recorders in *.cvg* files, we used an algorithm to convert into matrices (*.mat*) for easier processing. Once the voltages and currents matrices are obtained, they can be injected in DM-DFT algorithm that yields the fault point and extracts the features as seen in Fig. 11.

In real situations, faults can suddenly reappear for reasons as re-closure or reinsertion. This is the case on the CG-type fault we see in Fig. 11. The algorithm detects successfully the first fault occurrence and also locate the exact sample where the phasors are extracted to perform selection. The NN and QARMA techniques were applied in the real fault data with a successful result: both correctly classify the fault as CG. Particularly with QARMA, it yielded 1100 rules that predicted the class of the fault, which resulted in the overall correct classification of the test case. One of the highest confidence rules for this test case was,

$$\begin{aligned}
 &[\text{local_voltage}_A \geq 235730.0266612848] \\
 &\text{AND } [\text{local_voltage}_B \geq 231130.0132737731] \\
 &\text{AND } [\text{remote_ir} \geq 321.3212781340371] \\
 &\Rightarrow [\text{fault_type} = \text{CG}].
 \end{aligned}
 \tag{5}$$

The support of this rule on the training set is 2.72%, and it holds with confidence 100%.

4.3 Discussion

4.3.1 Implications of the results

Current implementations for real-time use cases, such as relay 21 in transmission lines, usually employ a full cycle of phasor estimation and around 4 ms of angle comparison between current/voltage components, as reported by some manufacturers. With either of the studied methods (DL or QARMA), once the model is generated based on historical data of the target system, the time taken to perform phase evaluation given a single-phase fault in the system is as small as 4 ms. Therefore, both methods can reliably select the faulty phase in the relay to make additional trip decisions. The algorithms described in this paper that are commonly implemented in relays in operation today employ a full cycle Fourier phasor estimation for both the protection and the phase estimation. Because the phasor calculation is done in real time, it is based on the Fourier transform (or another filter with a similar output) that provides the root-mean-square (RMS) value required for the proposed phase estimation, implying that only the phase selection itself has to be calculated.

Distance relays (ANSI 21) require 4 ms of angle comparison between current/voltage components, as reported by some manufacturers. The process to utilize the algorithms' outputs only requires multiplications and additional operations, making it more computationally efficient than most phasor operations and suitable for use in real-time applications.

4.3.2 Communication systems

As for the requirements for a communication setup where one can perform phase selection, no communication is needed between the two ends as a signal input, because once the model is generated, the validation is performed at the end. However, there is still a need for communication for data gathering from both ends for phasor estimation. Current mobile communication advances could enable wireless communication between the ends for instantaneous data gathering of currents and voltages, and interface diversity could enable a centralized system that is cheaper to implement in a communication architecture, as shown in Fig. 6.

4.3.3 Explainable results

When comparing the results of the DL method against the ones provided by QARMA, it is clear that the rule set produced by QARMA leads to a slightly lower accuracy than the DL method, while still being highly accurate. However, the resulting QARMA model is by default much more “explainable” than the DL model and has the extra advantage that it can be “reverse-engineered” much more easily than any other model. As an example, consider the following QARMA produced rule:

$$\begin{aligned} \text{local_current_A} &\in [433.89, 589.99] \\ &\wedge \text{local_current_B} \in [433.96, 564.25] \\ &\implies \text{fault_type} = 10 \text{ (no_fault)} \end{aligned} \quad (6)$$

which holds with support 3.66% and confidence 91.84% on the training set. This rule is highly statistically significant as it has conviction 1,115.06, and lift equal to 10.13. It is also obvious to a human what it means. As another example, consider the rule:

$$\text{local_current_A} \geq 438.39$$

$$\begin{aligned} & \wedge \text{local_voltage_B} \geq 235721.9 \wedge \text{local_voltage_C} \geq 225592.7 \\ \implies & \text{fault_type} = 0 \text{ (AG)} \end{aligned} \quad (7)$$

holding with support 4.19% and confidence 95.06%. Again, a human can understand what the rule means instantly.

When the QARMA rule set leads to a false diagnosis, it is trivial to see which set of rules led to the wrong decision. These rules can then be individually checked by human experts to see if their validity still holds in the face of new data and/or operating system conditions. Thus, at least, in principle, the entire model can be monitored and “debugged” in real time by human experts when it is put in production. This contrasts with models that make decisions based on the output of a highly nonlinear equation.

When less features are available in the training set, it has been shown that performance drops as expected. In certain cases, the performance degrades gracefully, but it can also be more serious. This performance degradation could be mitigated to a larger extent if we performed our simulations allowing the design of the deep network to vary with all hyper-parameters, from the number of layers and the number of epochs chosen, to the optimization algorithm used for the learning of the network weights and threshold biases. However, with such an approach, the ML process shown in Fig. 4 would essentially have to be repeated anew. Instead, we show how a network with predefined hyper-parameters, in particular those proposed in Sect. 3, performs when trained on different subsets of the original training dataset containing less features, such as local only information (local currents or local voltages and so on.)

Moreover, we also presented key aspects related to the real dataset from a transmission line and how the proposed method can be used by power engineers on their operational decision-making, including the association rules provided by QARMA that “explain” the fault selection. Results from association rules improve the knowledge on how power systems work in face of stressful events. This is indeed an important step if traditional engineering fields would rely more on machine learning methods. This sort of new explaining knowledge is to become evermore frequent in real-world applications as well as in academic research.

5 Conclusion

In this paper, we have proposed and analyzed a two-step methodology for selecting faults in double-circuit transmission lines. In the first step, the DFT was used to pre-process the raw data from the transmission lines. Subsequently, different learning algorithms were employed in the second step to detect and classify any fault based on a training period, and their performances were compared through numerical simulations. The presented two-step approach has been proven to be highly robust against high resistance faults and faults that occur in lines with high mutual impedance. The results have shown high phase selection for all types of faults and even identified recordings that do not present faulty states.

Among the different benchmarked learning methods, deep neural networks have reached an accuracy of 98.33% of correct selection, while the QARMA reached 98% accuracy. However, interestingly, the QARMA is also an explainable algorithm (i.e., the outcomes have explainable explicit internal relations between features) and also robust against noisy measurements unlike ANNs. This makes QARMA a highly suitable approach to achieve high robustness and high accuracy with explainable model outcomes. Future work will include the communication delay of the current and voltage signals sent to the central processing unit from the IEDs to evaluate the performance of the proposed method.

Acknowledgements This paper is partly supported by Academy of Finland (AKA) via EnergyNet Research Fellowship n.321265/n.328869. This work is also part of FIREMAN project supported by the CHIST-ERA Grants: (a) CHIST-ERA-17-BDSI-003 (b) T9EPA3-00017 and by Academy of Finland (n. 326270). We would like to thank Dr. Hanna Niemelä for valuable comments and for helping to proofread this paper.

Funding Open Access funding provided by LUT University (previously Lappeenranta University of Technology (LUT))

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abdelgayed TS, Morsi WG, Sidhu TS (2018) Fault detection and classification based on co-training of semi-supervised machine learning. *IEEE Trans Ind Electron* 65(2):1595–1605
2. Adamo J-M (2001) Data mining for association rules and sequential patterns. Springer, New York
3. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: 20th International conference on very large databases, pp 487–499
4. Bak CL, da Silva FF (2018) Deploying correct fault loop in distance protection of multiple-circuit shared tower transmission lines with different voltages. *J Eng* 2018(15):1087–1090
5. Chamoso P, De Paz JF, Bajo J, Villarrubia G (2018) Agent-based tool to reduce the maintenance cost of energy distribution networks. *Knowl Inf Syst* 54(3):659–675
6. Chang H-H, Linh NV, Lee W-J (2018) A novel non-intrusive fault identification for power transmission networks using power-spectrum-based hyperbolic s-transform—part I: fault classification. *IEEE Trans Ind Appl* 54(6):5700–5710
7. Chen K, Hu J, He J (2018) Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder. *IEEE Trans Smart Grid* 9(3):1748–1758
8. Chen YQ, Fink O, Sansavini G (2018) Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction. *IEEE Trans Ind Electron* 65(1):561–569
9. Chollet F (2018) Deep learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG, Blaufelden
10. Christou I et al (2020) End-to-end industrial iot platform for actionable predictive maintenance. In: 4th IFAC Workshop on advanced maintenance engineering, services and technologies. IFAC
11. Christou I et al (2020) Predictive and explainable machine learning for industrial internet of things applications. In: IEEE Distributed computing on sensor systems conference on, workshop on IoT applications and industry 4.0. IEEE
12. Christou IT (2019) Avoiding the hay for the needle in the stack: online rule pruning in rare events detection. In: 16th International symposium on wireless communication systems (ISWCS). IEEE, pp 661–665
13. Christou IT, Amolochitis E, Tan Z-H (2018) A parallel/distributed algorithmic framework for mining all quantitative association rules. arXiv preprint [arXiv:1804.06764](https://arxiv.org/abs/1804.06764)
14. da Silva FF, Bak CL (2017) Distance protection of multiple-circuit shared tower transmission lines with different voltages—part II: fault loop impedance. *IET Gener Transm Distrib* 11(10):2626–2632
15. Domzalski MJ, Nickerson KP, Rosen PR (2001) Application of mho and quadrilateral distance characteristics in power systems [relay protection]. In: 2001 Seventh international conference on developments in power system protection (IEE), pp 555–558

16. Fan X, Zhu Y (2010) Study on fault phase selection based on FFT and phase-separation current phase difference of high-voltage transmission lines. In: 2010 IEEE International conference on mechatronics and automation. IEEE, pp 762–767
17. Gashteroodkhani O, Majidi M, Etezadi-Amoli M (2020) A combined deep belief network and time-time transform based intelligent protection scheme for microgrids. *Electric Power Syst Res* 182:106239
18. Gururajapathy S, Mokhlis H, Illias H (2017) Fault location and detection techniques in power distribution systems with distributed generation: a review. *Renew Sustain Energy Rev* 74:949–958
19. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. *ACM SIGMOD Record* 29(2):1–12
20. Holbeck JI, Lantz MJ (1943) The effects of mutual induction between parallel transmission lines on current flow to ground faults. *Electr Eng* 62(11):712–715
21. Huang H, Yan Q, Zhao Y, Lu W, Liu Z, Li Z (2017) False data separation for data security in smart grids. *Knowl Inf Syst* 52(3):815–834
22. Jia Y, Bailey J, Ramamohanarao K, Leckie C, Ma X (2019) Exploiting patterns to explain individual predictions. *Knowl Inf Syst* 62(3):927–950
23. Junior GM, Di Santo SG, Rojas DG (2016) Fault location in series-compensated transmission lines based on heuristic method. *Electric Power Syst Res* 140:950–957
24. Kingma DP, Ba J (2017) Adam—a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
25. Mahfouz MM, El-Sayed MA (2016) Smart grid fault detection and classification with multi-distributed generation based on current signals approach. *IET Gener Transm Distrib* 10(16):4040–4047
26. Mishra M, Rout PK (2018) Detection and classification of micro-grid faults based on HHT and machine learning techniques. *IET Gener Transm Distrib* 12(2):388–397
27. Pazoki M (2018) A new fault classifier in transmission lines using intrinsic time decomposition. *IEEE Trans Ind Inform* 14(2):619–628
28. Price E, Einarsson T (2008) The performance of faulted phase selectors used in transmission line distance applications. In: Proceedings of 61st annual conference for protective relay engineers, pp 484–490
29. Rajaraman P, Sundaravaradan N, Meyur R, Reddy MJB, Mohanta D (2016) Fault classification in transmission lines using wavelet multi-resolution analysis. *IEEE Potentials* 35(1):38–44
30. Taheri MM, Seyedi H, Mohammadi-ivatloo B (2017) DT-based relaying scheme for fault classification in transmission lines using MODP. *IET Gener Transm Distrib* 11(11):2796–2804
31. Towell GG, Shavlik J (1994) Knowledge-based artificial neural networks. *Artif Intell* 70(1–2):119–165
32. Vyas BY, Das B, Maheshwari RP (2016) Improved fault classification in series compensated transmission line: comparative evaluation of Chebyshev neural network training algorithms. *IEEE Trans Neural Netw Learn Syst* 27(8):1631–1642

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Daniel Gutierrez-Rojas received the B.Sc. degree in Electrical Engineering from University of Antioquia, Colombia in 2016 and the M.Sc. degree in Protection of Power Systems University of São Paulo, Brazil, in 2017. From 2017 to 2019, he worked as security of operation and fault analyst for Colombia's National electrical operator. He is currently working toward the Ph.D. degree at the School of Energy Systems at LUT University, Finland. His research interests include predictive maintenance, power systems, microgrids, mobile communication systems and electrical protection systems.



Ioannis T. Christou received the Dipl. Ing. degree in electrical engineering from the National Technical University of Athens, Greece, in 1991, the MSc and PhD degrees in computer sciences from the University of Wisconsin at Madison in 1993 and 1996, respectively, and the MBA degree from NTUA and Athens University of Economics and Business in 2006. He has been with Delta Technology, Inc., as a senior developer, with Intracom S.A. as an area leader in data and knowledge engineering, and with Lucent Technologies Bell Labs as a member of technical staff. He has also been an adjunct assistant professor with the Department of Computer Engineering and Informatics, University of Patras, Greece, adjunct professor at the Information Networking Institute of Carnegie-Mellon University in Pittsburgh, PA, and professor of big data at Athens Information Technology. He is currently a Sr. research data scientist at the Dept. of Research and Innovation Development at NetCompany-Intrasoft (Luxemburg), and an associate professor at the American College of Greece (Greece). He has published

more than 80 papers in high-impact journals and conferences in the areas of computational intelligence, data mining, optimization, and parallel computing.



Daniel Dantas received the B.S. degree in electric and electronic engineering from the University of Brasilia, in 2012 and the Ph.D. degree in electrical power systems from the University of São Paulo, in 2019. He worked in protection and control systems in different capacities for utilities in Brazil, Canada, and as a researcher in the University of Manchester. Currently, he is at National Grid UK.



Arun Narayanan received his B.E. degree in Electrical Engineering from Visvesvaraya National Institute of Technology, Nagpur, India and M.Sc. in Energy Technology from Lappeenranta University of Technology (LUT), Finland, in 2002 and 2013, respectively. He subsequently completed his Ph.D. from the School of Energy Systems, LUT University, in 2019. He is currently a Postdoctoral researcher with LUT University, Lappeenranta, Finland, in the research group “Cyber-Physical Systems Group.” His research interests include renewable energy-based smart microgrids, electricity distribution and markets, demand-side management, energy management systems, and information and communications technology. He focuses on applying optimization, computational concepts, and artificial intelligence techniques to renewable electrical energy problems.



Pedro H. J. Nardelli received the B.S. and M.Sc. degrees in electrical engineering from the State University of Campinas, Brazil, in 2006 and 2008, respectively. In 2013, he received his doctoral degree from University of Oulu, Finland, and State University of Campinas following a dual degree agreement. He is currently Associate Professor (tenure track) in IoT in Energy Systems at LUT University, Finland, and holds a position of Academy of Finland Research Fellow with a project called Building the Energy Internet as a large-scale IoT-based cyber-physical system that manages the energy inventory of distribution grids as discretized packets via machine-type communications (EnergyNet). He leads the Cyber-Physical Systems Group at LUT and is Project Coordinator of the CHIST-ERA European consortium Framework for the Identification of Rare Events via Machine Learning and IoT Networks (FIREMAN). He is also Adjunct Professor at University of Oulu in the topic of “communications strategies and information processing in energy systems”. His research focuses on wireless communications

particularly applied in industrial automation and energy systems. He received a best paper award of IEEE PES Innovative Smart Grid Technologies Latin America 2019 in the track “Big Data and Internet of Things”. He is also IEEE Senior Member. More information: <https://sites.google.com/view/nardelli/>



Yongheng Yang received the B.Eng. degree in Electrical Engineering and Automation from Northwestern Polytechnical University, China, in 2009 and the Ph.D. degree in Energy Technology (power electronics and drives) from Aalborg University, Denmark, in 2014. He was a postgraduate student with Southeast University, China, from 2009 to 2011. In 2013, he spent three months as a Visiting Scholar at Texas A&M University, USA. Since 2014, he has been with the Department of Energy Technology, Aalborg University, where he became a tenured Associate Professor in 2018. In January 2021, he joined Zhejiang University, China, where he is currently a ZJU100 Professor with the Institute of Power Electronics, College of Electrical Engineering. His current research interests include the grid-integration of photovoltaic systems and control of power converters, in particular, the mechanism and control of grid-forming power converters and systems. Dr. Yang was the Chair of the IEEE Denmark Section (2019-2020). He is an Associate Editor for several IEEE Transactions/Journals. He is a Deputy

Editor of the IET Renewable Power Generation for Solar Photovoltaic Systems. He was the recipient of the 2018 IET Renewable Power Generation Premium Award and was an Outstanding Reviewer for the IEEE TRANSACTIONS ON POWER ELECTRONICS in 2018. He received the 2021 Richard M. Bass Outstanding Young Power Electronics Engineer Award from the IEEE Power Electronics Society (PELS). In addition, he has received two IEEE Best Paper Awards. He is currently the Secretary of the IEEE PELS Technical Committee on Sustainable Energy Systems.