The Journal of Engineering

The 14th International Conference on Developments in Power System Protection (DPSP 2018)

Automatic analysis of faulty low voltage network asset using deep neural networks

eISSN 2051-3305
Received on 4th May 2018
Accepted on 23rd May 2018
E-First on 7th September 2018
doi: 10.1049/joe.2018.0249
www.ietdl.org

Marcello Mastroleo[1], Roberto Ugolotti[1] ✉, Luca Mussi[1], Emilio Vicari[1], Federico Sassi[1], Francesco Sciocchetti[1], Bob Beasant[2], Colin McIlroy[2]
[1]*Camlin Italy, Strada Budellungo 2, Parma, Italy*
[2]*Camlin Technologies, 31 Ferguson Drive, Lisburn, Northern Ireland*
✉ *E-mail: r.ugolotti@camlintechnologies.com*

Abstract: Electrical distribution network is constantly ageing worldwide. Therefore, probability of cable faults is increasing over time. Fast recovering of damaged networks is of vital importance and a quick and automatic identification of the failure source may help to promptly recover the functionality of the network. The scenario we are taking into consideration is a vast number of recording devices spread across a network that constantly monitor low voltage cables. When the current of a cable reaches a very high value, data is sent to a central server which analyses it through a variant of a Variational Auto Encoder (VAE), a deep neural network. This VAE has been trained by using historical data collected from several hundreds of faults recorded, but in which only a handful of them has been labelled by an on-site analysis of the fault. Data used for training is simply the recorded levels of voltages and currents, after a simple pre-processing step. The final goal is to let the network distinguish if the fault occurred in a point of the cable, on a joint, or at the pot-end located at the termination. A preliminary evaluation of its ability to generalise over the non-labelled samples shows encouraging results.

1 Introduction

Electrical distribution network is constantly ageing worldwide. The power system currently installed in many countries has been mostly developed during the 1950s and 1960s. In the UK, some of the oldest low voltage (LV) assets are almost 100 years old and the weighted average age of the existing network is 40 years. It is estimated that in the in 2050, the annual electricity demand in the UK will raise >50% [1]. Considering also other sources of stress, like the increasing use of heat pumps and electric vehicles, the probability of cable faults is increasing over time.

Fast recovering of damaged networks is therefore of vital importance. A quick and automatic identification of the failure source may help to promptly recover the functionality of the network. This can bring great benefits to distribution network operators, as well as to connected clients.

Machine learning (ML) is the field of computer science that allows computer to learn without being explicitly taught. In the last years, the spread of this set of techniques has covered all possible field of applications. The analysis of electrical distribution network is no exception.

According to a study presented in 2016 by SAS [2], the majority of utilities interviewed agree that ML is an important technology trend and that will be critical for their company's future success but only 20% are already using it and even less have a specific and comprehensive strategy.

Nevertheless, applications of ML techniques in the utility sector, in particular in tasks like fault detection, diagnosis and localisation, started during the nineties [3] and their usage increased since then. A quick overview of publications of the last 10 years shows ML applications on fields that vary from the analysis of non-technical loss [4], improvement of the reliability of the network [5], smart grid management [6], fault localisation [7], fault prediction [8] and fault type classification [9].

In this paper, we developed and employed a novel variant of a deep neural network to analyse the data associated with a fault over a power network. In particular, this network receives in input a few cycles of currents and voltages before and after a fault and tries to estimate if the fault occurred in a point of the cable, on a joint, or at the pot-end located at the termination.

This paper is structured in this way: the Sapient architecture that collects the data used in the experiments is presented, as well as the ML technique we have used. The following chapters treat the experimental set-up, followed by results and a brief discussion of them.

2 Sapient architecture

Sapient (https://www.camlingroup.com/sapient) is an intelligent and integrated system able to assist customers' needs, from asset monitoring, health reporting and fleet ranking, to failure analysis, overload evaluation and design review.

It is based on a large number of recording devices spread across the UK grid which constantly monitor electrical currents and voltages of low-voltage cables. When, for example, the current of a cable exceeds a pre-set value, data is recorded and sent to a central server.

These devices are placed in the LV protection fuse position. They are internally developed and use various techniques (smart fuses, vacuum circuit breakers etc.) to close the power line and re-energise cables after a pre-determined time to reduce the customer minutes lost. They are designed to let the network be autonomously restored drastically reducing the duration of interruption for the customers. At the same time, they collect diagnostic information used for the many purposes of Sapient [10].

One of the main tasks Sapient performs is the precise localisation of intermittent and permanent faults for a fast recovery of the network. Currently, an algorithm named single-ended location of faults (SELF) uses the voltages and currents recorded around the fault event in conjunction with information on the cable parameters (type, length, shape, etc.) to localise its source.

The work presented in this paper describes a decision support system for SELF based on deep neural networks (DNNs). The goal is not to directly understand in which location the fault has happened but to provide an answer to which type of asset has been involved:

- A fault that occurred within the cable itself (e.g. deterioration of insulation, mechanical accidents etc.)

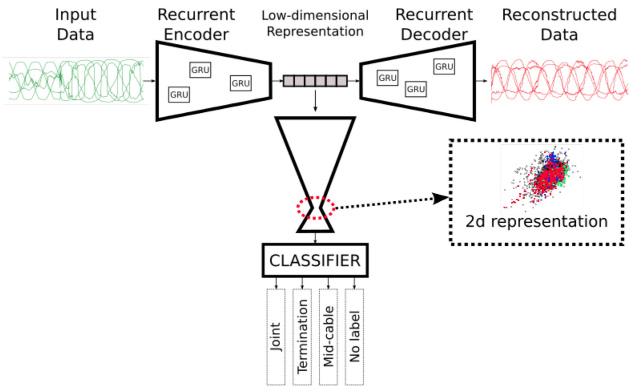

Fig. 1 *Representation of the network. The 2D representation shown in Fig. 3 is taken from an intermediate layer of the network that feeds the classifier*

- A fault that occurred in a joint of two or more cables (e.g. due to joint deterioration or human errors done at the time when these operations have been performed etc.)
- A fault that occurred at the pot-end (or termination) of the cable.

3 Neural networks

In this section, we are going to introduce the computational tools we have used in the experiments, trying to avoid, when it is possible, a too technical language.

3.1 Artificial neural networks (ANNs)

An ANN is a computational model inspired from structure and functions of biological neural networks.

They undergo a training phase in which they try to learn a function that maps an output Y from an input X. This function exploits the complex structure of an ANN in which there are many interconnected non-linear functions whose parameters are learnt during this training phase

$$Y \simeq F(X)$$

Once the training phase is completed, the network can receive an input and computing the corresponding output.

When a network's structure is composed of different layers, it is commonly called DNN.

3.2 Variational autoencoders (AEs)

An AE is a type of DNN mostly used for unsupervised learning or dimensionality reduction. In other words, it exploits the interdependence and redundancies in highly dimensional data so to learn a low-dimensional simpler structure containing the same information.

The AE is the composition of two different ANNs: an encoder and a decoder, which are trained together. The goal of the encoder is to learn this low-dimensional representation Z, while the encoder does the opposite and reconstructs a lossy approximation (X') of the original data from Z. By doing this, the low-dimensional representation is forced to represent the original set of data as closely as possible

$$Z \simeq F_{\text{encoder}}(X)$$

$$X' \simeq F_{\text{decoder}}(Z)$$

If we consider the whole sequence of encoder and decoder, the entire AE mimics a function that re-creates the input:

$$X' \simeq F(X)$$

A variational autoencoder (VAE) [11] is an AE variant in which there is a further layer placed between the encoder and the decoder. The job of this layer is to learn low-dimensional probability distribution that can approximate the high-dimensional data. In this way, the original data can be seen as a series of functions in a low-dimensional domain, making the problem easier to tackle.

3.3 Recurrent neural networks (RNNs)

RNNs are a family of ANN that has the ability to process temporal sequences. They are able to do so by keeping an internal state S that models the past values of the input sequence. Therefore, following the previous notation, the output is obtained as follows:

$$Y \simeq F(X, S)$$

During the years, several problems have been noticed with classic RNNs, especially their inability to remember events that happen in a 'distant time' due to mathematical reasons. For this reason, new paradigms have been proposed (LSTM, GRU [12]). They save the state by means of memory blocks, like it happens in a computer. In this way, they are more effective in remembering events that have happened in the past which will have an effect in the future.

3.4 Recurrent VAEs

Combining the two paradigms hereby presented, it is possible to obtain an AE that is able to process temporal sequences, such as, in our case, time series of current and voltages.

The first implementation of a recurrent VAE (RVAE) has been presented in [13] for noise reduction in speech recognition.

In this case, the encoder is composed of a combination of RNNs (or one of their variants) and takes in input the temporal series one instant at a time. The low-dimensional representation Z is the output of these RNNs throughout the entire input sequence. The decoder recreates the sequence X taking as input a series whose values are Z for the entire time. So, from a logical point of view, the behaviour is roughly the same as standard AEs.

3.5 Semi-supervised RVAE

Although AEs and all their variants have been created for unsupervised learning (no label associated to the input and no ability to perform classification tasks), we can easily extend this by taking into account the labels L together with data:

$$Z \simeq F_{\text{encoder}}(X, L)$$

$$X', L' \simeq F_{\text{decoder}}(Z)$$

In our case (for a more precise description of the dataset used, see Section 4), we have labels only for a subset of the data. This scenario is called semi-supervised learning.

There are many ways to include this information into an RVAE. We decided to use the low-dimensional representation as input to an ANN that works as a classifier. Obviously, it considers only data with a label associated and ignores data without it.

Fig. 1 shows the final architecture used. Data is passed to the encoder of the RVAE to create the low-dimensional representation. This representation is passed to another ANN followed by a simple classifier which gives the final answer.

With this kind of architectures, the representation of data is well separated from the classification task. This modular structure helps in highlighting the relevant features within data regardless the final task, thus mitigating the natural over-fitting risk that there can be with a low number of available labels.

The authors thank the semi-supervised paradigm, it is possible to infer the missing labels by propagating the known ones. The goal we want to achieve is to quantify the ability of such a network in performing this task.

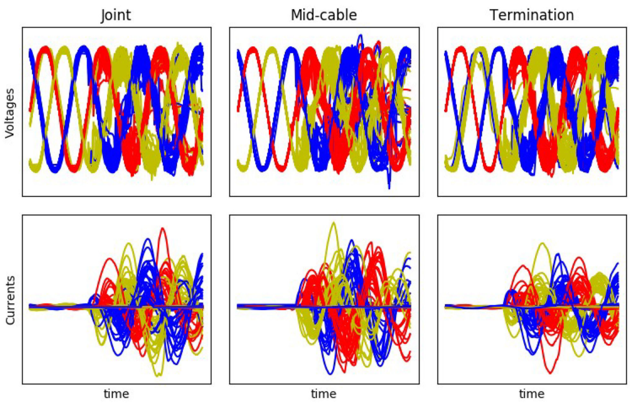

Fig. 2 *Data samples divided by label. First column represents faults that happened on a joint, second within the cable itself, and last at the termination of the cable. Lines in red, yellow and blue correspond to phases 1, 2 and 3, respectively*

Table 1 Composition of the training set

Label	No. of Cables	No. of Dumps
joint	58	1029
termination	9	252
mid cable	14	245
no label	426	8521
total	507	10,047

4 Experimental set-up

4.1 Description of the dataset

The dataset we use in our experiments is composed of data acquired by Sapient over the last few years in several areas of the United Kingdom.

As we previously said, data is collected on-site by a device which sends data to a central server when current reaches a pre-determined threshold. Therefore the data collected is a recording (not less than 0.2 s at a frequency of 12.8 kHz) of three-phase cable voltages and currents before and after a possible fault has occurred.

In some of the historical data, when the fault actually has interrupted the current flow and a manual intervention was necessary to recover the power line, the element responsible of the fault has been recorded. This represents the label that will be associated to the data.

Fig. 2 shows an example of data. Each column represents some examples of a label. Top rows are voltages, bottom rows are currents. The first third of data (easily recognisable by lower current values) is the pre-fault data. It can be seen that no clear label distinction can be made by looking at raw data. The complexity of the problem is confirmed by experiments in which we applied classical ML techniques, all of them leading to results without a significant level of robustness. This is the reason that brought us to tackle the problem by employing deep learning techniques.

4.2 Data format and processing

Data is analysed by the network as it arrives from the devices. A few pre-processing operations are done. The first one is to cut all sequences with the same length centred on the actual fault start. This is known method to simplify the learning task of sequences by means of neural networks. The sequences start 0.02 s before the first zero-crossing before the fault happened and end 0.04 s afterwards. So, a piece of datum is generally composed of three full cycles of currents and voltages.

All acquisitions are re-aligned to start with the zero-crossing of phase 1 to simplify the task of the network.

The second step is to compute the absolute value of the signals to reduce the variability and enhance the regularities of data.

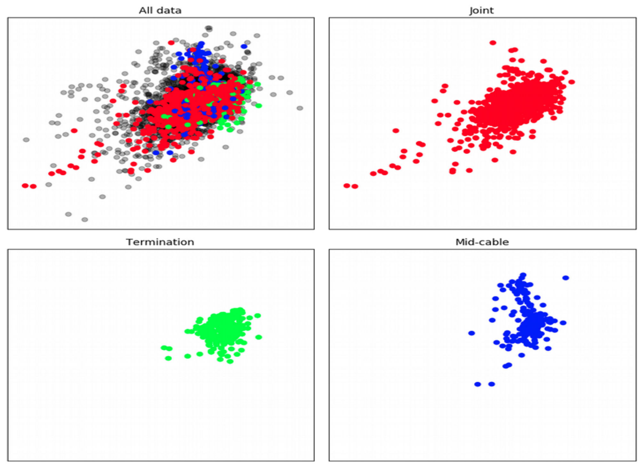

Fig. 3 *Low-dimensional encoding obtained by the network. The first scatter plot shows the disposition of the entire dataset, while the other three show only the disposition of labelled data (in clockwise order, joint, mid-cable and termination)*

Data is normalised before being passed to the network so to have values bounded between 0 and 1.

Labels are passed to the network following a one-hot encoding. When there is no label is available, all entries associated to labels are set to zero.

Table 1 shows the cardinality of the dataset. Note that for each cable there can be several dumps.

The network input is a 768×6 entries matrix. The encoder reduces it to 768 values, which are then processed by the classifier, which has two hidden layers with 32 and 2 neurons, respectively. The output of the classifier is a 'soft' classification, i.e. a probability distribution over the labels. The last branch of the network is the decoder, which outputs the reconstructed input, which obviously has 786×6 entries.

5 Results

Considering the low amount of labels currently available, the presented results are mainly qualitative.

The first way we present the results is to show how data is displaced in the low-dimensional space (see Fig. 1). Each point in Fig. 3 shows the two-dimensional representation of data from a single fault; in other words each fault event is represented by a single point. The colour represents the LV asset associated with the fault. If it is unknown, the dot is black.

Several interesting conclusions can be drawn from these, in particular:

- If we consider labelled data, there are some areas only covered by a single kind of fault (most evident, the lower left part only covered by the 'joint' label). This is an evidence of some specific characteristics of this kind of fault.
- If we consider also unlabelled data, there are some areas only covered by them. It means that the labels available to us are not covering the entire possible fault variations that can happen.
- There is an area in which different labels coexist. This may be the symptom that the data representation we are using is not rich enough to capture all the complexity of the problem at hand. For instance, among the missing information we can mention the frequency and duration of the fault activity, the cable topology and all cable structural information in general.

We set-up the next experiment with the main goal of testing the ability of such a network to correctly propagate the known labels over the unknown samples of data. To do so, by exploiting the semi-supervised capability of the network, we hid some known labels during the training of the network and see its ability to retrieve them.

Fig. 4 shows the confusion matrix of the entries with a known label that were left out from the training phase as a test set. It can be seen that the overall accuracy is 61%. However, if we merge events that happen outside the cable (therefore, in connections or terminations, which share many physical properties), the classification accuracy raises up to 92%. As already explained, this is not intended as a classification analysis mainly due to the lack of labels. Nevertheless, this exploration highlights the ability of this kind of networks to cope with the phenomena involved in cable faults.

The map shown in Fig. 5 depicts the soft classification of the network. Each vertex of the triangle corresponds to a label type, while the position of the points is obtained by averaging the triangle vertices positions with the soft classifications gained by the network on the test set data, the colour is the known label. Therefore, the closest the point is to a vertex the more confident the network is about the label being the one on the vertex; when the colour of the point is the same of the closest vertex the network guess is correct.

The ability of the network to distinguish between faults that happened on the cable from the others (terminations and joints) is clearly evident by looking at the dashed grey line that separates the receptive areas of the two sources.

6 Discussion and future works

In this paper, we presented an ML-based method that may help to understand which kind of asset is the cause of a fault. This indication may be of very high value for a fast recover of a damaged power line since it directs the repair team intervention to the correct faulty asset.

The network we built receives in input three cycles of the three voltages and current recorded when the fault occurs and, although a proper classification has not been provided, proved its ability to preliminary distinguish if the fault happened in a joint, at a termination, or in the cable itself.

We are currently enriching our database. When enough labels will be available, we will re-train the very same network and see its actual ability to give a classification result. Moreover, we will be able to have better insights on what properties of data the network has used to separate labels.

Another interesting development will be to enrich the representation of the data given in input to the network; in fact the current format ignores many information that are used by experts of the field such as the duration of a fault, the kind of cables used (or, more in general, the structure of the network in which the fault happens) etc. The goal of this approach is to let the NN ‘think’ in the same way as experts usually do.

7 Acknowledgments

The authors thank the colleagues in Lisburn and Parma that contributed to the paper, with suggestions, expertise and hard work, in particular, Alessandro Pietro Bardelli, Pietro Cottone, John McClure and David Wilcox.

8 References

[1] Cambridge Economic Policy Associates: ‘Economic lives of energy network assets’. Technical Report, 2010

[2] SAS Institute: ‘The autonomous grid: machine learning and IoT for utilities’. Technical Report, 2016

[3] Ebron, S., Lubkeman, D.L., White, M.: ‘A neural network approach to the detection of incipient faults on power distribution feeders’, *IEEE Trans. Power Deliv.*, 1990, **5**, pp. 905–914

[4] Nizar, A.H., Dong, Z.Y., Wang, Y.: ‘Power utility nontechnical loss analysis with extreme learning machine method’, *IEEE Trans. Power Syst.*, 2008, **23**, pp. 946–955

[5] Bouhouras, A.S., Labridis, D.P., Bakirtzis, A.G.: ‘Cost/worth assessment of reliability improvement in distribution networks by means of artificial intelligence’, *Electr. Power Energy Syst.*, 2009, **32**, pp. 530–538

[6] Fang, X., Misra, S., Xue, G., *et al.*: ‘Smart grid – the new and improved power grid: a survey’, *IEEE Commun. Surv. Tutorials*, 2012, **14**, (4), pp. 944–980

[7] Livani, H., Evrenosoglu, C.Y.: ‘A machine learning and wavelet-based fault location method for hybrid transmission lines’, *IEEE Trans. Smart Grid*, 2014, **5**, pp. 51–59

[8] Rudin, C., Waltz, D., Anderson, R., *et al.*: ‘Machine learning for the New York city power grid’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, pp. 328–345

[9] Jiang, J.A., Chuang, C.L., Wang, Y.C., *et al.*: ‘A hybrid framework for fault detection, classification, and location – Part I: concept, structure and methodology’, *IEEE Trans. Power Deliv.*, 2013, **26**, pp. 1988–1998

[10] Gillick, M., Rodgers, J.: ‘Management of Scottish and southern energy's low voltage distribution cable network using smart fuses with online fault detection’. 22nd Int. Conf. Electricity Distribution, Stockholm, Sweden, 2013

[11] Kingma, D.P., Welling, M.: ‘Auto-encoding variational Bayes’. Proc. 2nd Int. Conf. Learning Representations, Banff, Canada, 2014

[12] Chung, J., Gulcehre, C., Cho, K., *et al.*: ‘Empirical evaluation of gated recurrent neural networks on sequence modeling’. NIPS 2014 Deep Learning and Representation Learning Workshop, Montreal, Canada, 2014

[13] Maas, A.L., Le, Q.V., O'Neil, T.M., *et al.*: ‘Recurrent neural networks for noise reduction in robust ASR’. Thirteenth Annual Conf. Int. Speech Communication Association, Portland, OR, USA, 2012

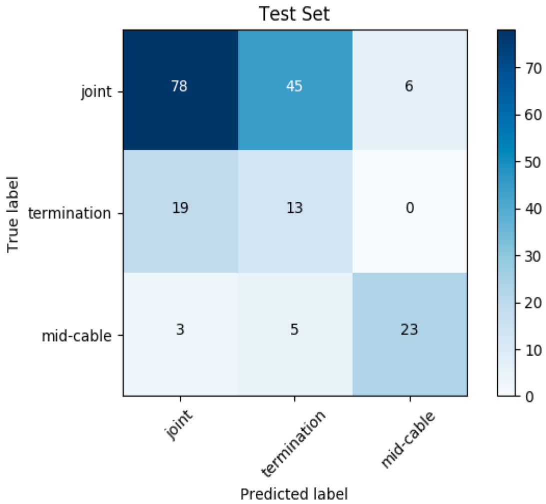

Fig. 4 *Confusion matrix on data with known labels in the test set. Entries of the matrix represent the actual number of dumps*

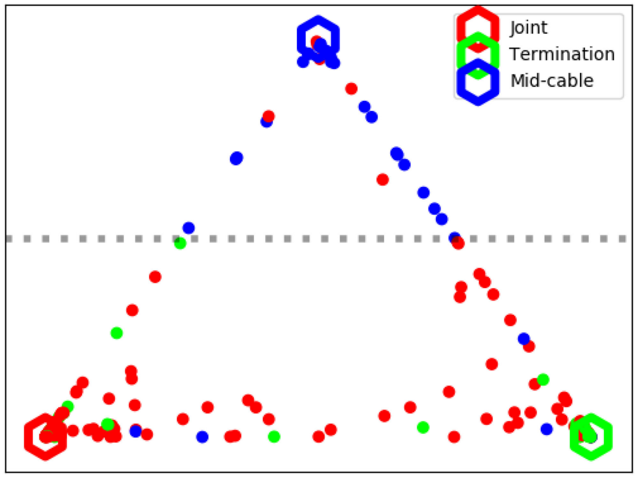

Fig. 5 *Soft classification of the network on the test set. When the network is sure about a label, the corresponding point lies within one of the hexagons at the vertices of the triangle; when in doubt between two classes, on the edges connecting the vertices. When in doubt among the three classes, the point lies within the triangle*