


Agent-based tool to reduce the maintenance cost of energy distribution networks

Pablo Chamoso¹  · Juan F. De Paz¹ · Javier Bajo² · Gabriel Villarrubia²

Received: 14 October 2016 / Revised: 1 September 2017 / Accepted: 10 October 2017
© Springer-Verlag London Ltd. 2017

Abstract There has been continuous research in the energy distribution sector because of its huge impact on modern societies. Nonetheless, aerial high voltage power lines are still supported by old transmission towers which involve some serious risks. Those risks may be avoided with periodic and expensive reviews. The main objective of this work is to reduce the number of these periodic reviews so that the maintenance cost of power lines is also reduced. More specifically, the work is focused on reducing the number of periodic reviews of transmission towers to avoid step and touch potentials, which are very dangerous for humans. A virtual organization-based multi-agent system is proposed in conjunction with different artificial intelligence methods and algorithms. The developed system is able to propose a sample of transmission towers from a selected set to be reviewed. The system ensures that the whole set will have similar values without needing to review all the transmission towers. As a result of this work, a website application is provided to manage all the review processes and all the transmission towers of Spain. It allows the companies that review the transmission towers to initiate a new review process for a whole line or area, while the system indicates the transmission towers to review. The system is also able to recommend the best place to locate a new transmission tower or the best type of structure to use when a new transmission tower must be used.

Keywords Virtual organizations · Transmission towers · Maintenance · Case-based reasoning

✉ Pablo Chamoso
chamoso@usal.es

¹ BISITE Research Group, University of Salamanca, Edificio I+D+i, Calle Espejo 2, 37007 Salamanca, Spain

² Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Campus Montegancedo, 28660 Boadilla del Monte, Madrid, Spain

1 Introduction

Power line maintenance is a topic that has generated a variety of research lines [1–3]. In almost all developed countries, the transmission towers (TTs) that support the aerial power lines must be reviewed on a regular basis, depending on their characteristics. There are different parameters that have to be measured when reviewing TTs, which are necessary to determine the ground resistance, as well as the step potential (Kp) and touch potential (Kc).

These reviews involve a significantly high cost because they require qualified personnel to work in remote locations with expensive equipment. However, many of the values measured in the reviews could in fact be predicted; this is because most of the TTs share the same features and are located on similar terrain, so they have similar behavior. Therefore, the possibility of reducing the costs associated with this kind of maintenance is not only attractive, but quite reasonable.

As technology has continued to advance, there have been different approaches that attempt to apply innovations both in the review and the maintenance processes, resulting in a common need to reduce costs. Indeed, this is precisely the reason for having created the proposed predictive maintenance system.

There are four common maintenance types for TTs: (i) corrective, to solve existing problems; (ii) preventive, to prevent system failures; (iii) predictive, to predict possible irregularities; (iv) proactive, which is a combination of preventive and predictive maintenances [4]. The present work is focused on the predictive maintenance, where different techniques are already being used: some authors have used artificial neural networks to model the environment, including [5], while others use neural networks to set failure times of the devices [6].

This work proposes an organization-based multi-agent system capable of carrying out a predictive maintenance of TTs by selecting only a sample of TTs to review when a company is going to start a new revision process. The company need only select one or more power lines supported by different TTs. This is the only input of the system and is processed by an agent. This software agent then communicates the information to other agents of the system which are in charge of executing the different algorithms and functionalities. The sample is determined and the system then proposes to the user a subset of the selected TTs to review.

The algorithms that the system executes for the selection of the TTs to review is completely autonomous and is based on different TT parameters that make it possible to determine the status of all the selected TTs, whether they have been reviewed or not in the current review process.

In addition, the system is able to predict different values that are measured in the revision process, making it possible to determine the best location or model to use when including a new TT.

The system has built-in statistical sampling techniques combined with neural networks to estimate the ground resistance, as well as the step and touch potentials. In addition, the system provides different geopositioning tools to facilitate the search and selection of the TTs and lines to be reviewed.

The remainder of this paper is organized as follows: Sect. 2 includes a revision of related work, Sect. 3 describes the proposal developed in this work, Sect. 4 provides the preliminary results, including the software tool and the evaluation of the advantages obtained from its use, and finally, Sect. 5 presents the conclusions obtained and future research lines.

2 Background

In this section, the state of the art is detailed from three perspectives: (i) industrial maintenance and its application to high voltage power lines, (ii) virtual organizations of software agents, and (iii) other complementary artificial intelligence techniques that have been used in this work.

2.1 Industrial maintenance in aerial energy distribution networks

As mentioned in the introduction section, there are different kinds of maintenance applied to aerial energy distribution networks [7]: corrective, preventive, proactive, and predictive.

Corrective maintenance consists of fixing existing failures or problems to make the system works properly again [8]. This kind of maintenance can be unplanned, with the condition being fixed randomly without intervention or planning; the failure is corrected intentionally because it was previously detected by predictive or preventive maintenance.

Preventive maintenance tries to reduce the equipment failures by searching for solutions to different problems before they occur, thus avoiding unplanned corrective maintenance. While this kind of maintenance leads to increased reliability and cost savings, it does not, however, guarantee that failures will not occur in the future. Preventive maintenance has been previously applied in different works such as [9] to confront the problem of power lines maintenance.

Predictive maintenance emerges as a complement for corrective and preventive maintenances. In this kind of maintenance, a set of parameters are monitored and analyzed to determine possible anomalies. The process essentially refers to the capacity of generating assumptions or estimations about the status of a component. When predicting well-known processes, especially in Control Theory [10], it is possible to generate a mathematical model which represents reality in a reliable way [11]. However, experimental techniques are needed in other processes, for example classification algorithms [12] or artificial neural networks [5]. This approach tries to extract and model the system features from historical data.

Proactive maintenance is a strategy used to maintain the stability and performance of a machine [13]. The useful life of the equipment is extended, while flaws or malfunctions are avoided. There are two types of repair [14]: perfect maintenance, when the machine returns to its ideal state (in this case the cost is usually high), and imperfect maintenance, where the quality is sacrificed to reduce maintenance costs.

Support vector machines are used in [15] to predict the amount of ice that will be accumulated in aerial power lines. This is a serious problem that can interrupt electrical service for a significant time, and the solution may end up being significantly expensive. As a result of that study, it is possible to estimate the level of ice (with a minimum error) by using historical meteorological data.

Another point to take into account is the machine performance (generators and current transformers). Due to the natural deterioration or machines over time, they must undergo reviews over their useful lifespan. Periodic maintenance can be carried out, whereby machine performance is evaluated on a regular basis, regardless of the machine status. This solution is not optimal when the review period is short and the machines are in perfect working condition. An alternative is based on monitoring the status of the equipment and evaluating some of their parameters. From the combination of these two options, a new model is presented in [16], where performance loss is predicted by using failure rate and performance degradation in conjunction with their derivatives and a Weibull distribution [17].

In other works such as [7], current transformer failures are analyzed by Dissolved Gas Analysis (DGA). Existing methods are based on monitoring every substance ratio in oil; limit values are then established to determine all the possible failures [18]. The possibility of applying artificial neural networks and similar techniques to try to predict values is also presented in [7]. Weibull [17] applies Principal Component Analysis (PCA) and back-propagation artificial neural networks (BP-ANN) to obtain a discrete transformer status (normal, waiting for confirmation, abnormal). The accuracy levels reached are between 92 and 96% largely due to the input data PC treatment. Without this technique, the accuracy varies from 69 to 75%.

2.2 Virtual organizations of agents

Organizations of software agents are increasingly common in current research. VOs are open systems where groups of heterogeneous agents collaborate, and there is a clear separation between the structure and function defining how the entities behave [19,20]. In [15], a VO is defined as a collection of roles that maintain certain relationships with one another and that take part in patterns of interaction with other roles in an institutionalized and systematic fashion. VO can adopt different topologies, which Rodríguez presents in [21]: hierarchy, oligarchy, coalition, groups, matrix, federations and congregations. The topology that is best suited to the proposed system is group one, where groupings of cooperative agents work together to achieve a common goal. This kind of topology is usually applied when solving problems and can be achieved better with joint work. In order to face the sampling objective, all the TTs involved in the revision process have to collaborate to determine the subset of TTs that must be reviewed, which is their common goal.

Different VO architectures have been proposed in recent research in an attempt to deal with problem management and planning [21]. This work is no centered in the definition of new management or planning system in VO, the main goal of this work is to involve planning the reviews for a set of selected TTs. The use of open VO allows to modify agents and their assigned roles at runtime, this makes it easier to alter functionalities in the case of having to modify different parameter estimation procedures.

2.3 Complementary technologies

Different artificial intelligence techniques have been used in this work. Artificial Neural Networks (ANN) and Case-Based Reasoning (CBR) systems are the most significant.

ANN can be applied in many fields to classify patterns, make clusters, or perform predictive, optimization and control tasks, among others. Among neural networks, this work focuses on neural networks of supervised learning applied to prediction, among the most well-known existing neural networks we have the multilayer perceptron (MLP) [22]. Radial basis function (RBF) [23] or recurrent neural networks as long-term memory (LSTM) [24]. This technique has been used for years now in electric applications, primarily to estimate, predict failures, predict risks or traffic flows, among others [23–27]. Although it is not a new artificial intelligence technique, the use of ANN is still a trend and is especially useful in predictive maintenance [28], which is the maintenance model used in this work. In addition to neural networks, there are other models that can be applied in prediction systems such as Support Vector Regression (SVR) or different linear and nonlinear models. Although these models do not work in general, they are applicable in certain case studies [29,30]. In this work we opt for handling a multi-sensor perceptron since, according to Kolmogorov, any function can be accurately approximated by a single hidden layer. In addition, we introduce restrictions on the number of hidden layer neurons and the functions of activation and trans-

fer. However, these activation and transfer functions were only designated to exist; however, they are not meant for calculating the approximate, since this results in inaccuracy, instead generic functions are used.

CBR tries to adapt old solutions to meet new demands, using old cases to explain new situations or critique new solutions, and applying a reasoning process to previous cases to interpret a new situation or create an equitable solution to a new problem [31]. The main CBR process involves the cases that define a problem, the solution and the final result. In addition, when the system receives a new case, that case is solved by a reasoning cycle composed of the following stages: retrieve, reuse, revise and retain. In these stages of the reasoning cycle, other machine learning techniques can be introduced that allow to calculate the solution to a new problem from the most similar cases recovered. In the current case, a CBR is used to access previous solutions and adapt them to the demand for new TT reviews. Parameters usually depend on the TT structure and location, so TTs that are near each other and share a similar structure are supposed to share their behavior.

This case study uses a CBR system that incorporates an MLP neural network, which is described in the following section.

3 Proposal

The objective of this study is twofold: first, to predict the resistance of existing TTs with unknown values; and second, the most important objective, to reduce the number of TTs to be reviewed with samples. To carry out these tasks, a multi-agent system (MAS) based on virtual organizations (VOs) of agents is proposed, which follows the structure shown in Fig. 1.

As shown, there are three VOs. The main one is 'Data processing', which includes different roles to process the information. The information is managed by the agents that implement the roles included in the 'Information' VO. Finally, the third VO, 'Communications', is responsible for processing all the messages exchanged between the system and the user.

In the '**Data processing**' VO, there are 4 roles to predict: resistance, resistivity, K_p (step potential) and K_c (touch potentials). An additional role provides a MLP to carry out these predictions. The K_p and K_c values are calculated by the K_p agents and the K_c agents, according to the K_R value, by using the MLP implemented by the MLP agent. Resistance and Resistivity roles are implemented by two agents which are in charge of calculating the values for the resistance and the resistivity by applying the algorithm described in Sect. 3.1.

Other virtual organizations have less importance in the system. The '**Information**' VO contains three roles. The TT role is implemented by a set of agents that represent the information of every TT included in the system. There is also a Line role, implemented by an agent that manages the relationship of a set of connected TTs. The third role included in this VO is Sampling, implemented by an agent that represents the sample of towers to be revised.

The '**Communications**' VO includes roles to receive and send messages to the web application (represented by a User agent) and the delivery of the received messages to the system. The messages are first sent to the Information VO, which will use the agents allocated in the data processing virtual organization when needed.

The agent system has been developed by using a VO-based MAS platform called PANGEA [32], which uses FIPA ACL for the message structure specification [33]. PANGEA also incorporates agents that are not illustrated in Fig. 1, such as the manager agent, the service agent, the normative agent, the organization agent, the information agent, the database agent

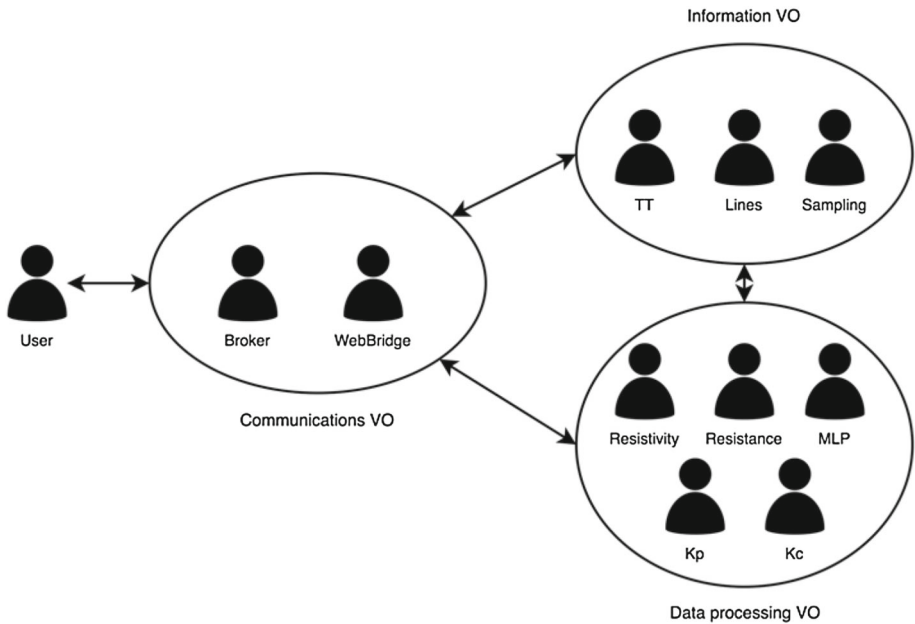


Fig. 1 Virtual organizations of agents schema

which is in charge of managing the norms that control the VO, the existing agents, the services and the communications that take place in the VO. A greater description of these agents can be obtained in the works [32]. All the roles included in the VOs are explained below:

- **User:** This role is implemented by agents in charge of the direct interaction between the users which access through the web browser and the functionality of the system.
- **WebBridge:** This role is implemented by one agent in charge of receiving the PHP requests from the User agents and translate them to FIPA/ACL. To do this, a socket service is open to listen petitions through the port 1234. Whenever a new message is received through the socket, this agent sends the received string to the Broker agent. When a response is obtained, the WebBridge agent resends it through the socket to the User agent that initiated the operation.
- **Broker:** This role is implemented by one agent in charge of dispatching all the messages received from the WebBridge agent to the corresponding agents; this depends on the type of the received message. Once the response is received, the Broker agent sends it to the WebBridge agent.
- **Kc:** This role is implemented by agents that, when they start the execution, they load the data from a trained network in order to predict the Kc value. If there is no previous training, the agent trains the network. When a message is received from the Broker agent, the Kc agent uses the K_R value to obtain the corresponding predicted Kc value by using the neural network. Once the prediction is obtained, the resulting value is sent to the Broker agent.
- **Kp:** This role is implemented by agents which first load the data from a trained network to predict the Kp value. If there is no previous training, the agent trains the network. When a message is received from the Broker agent, the K_R value is used to obtain the

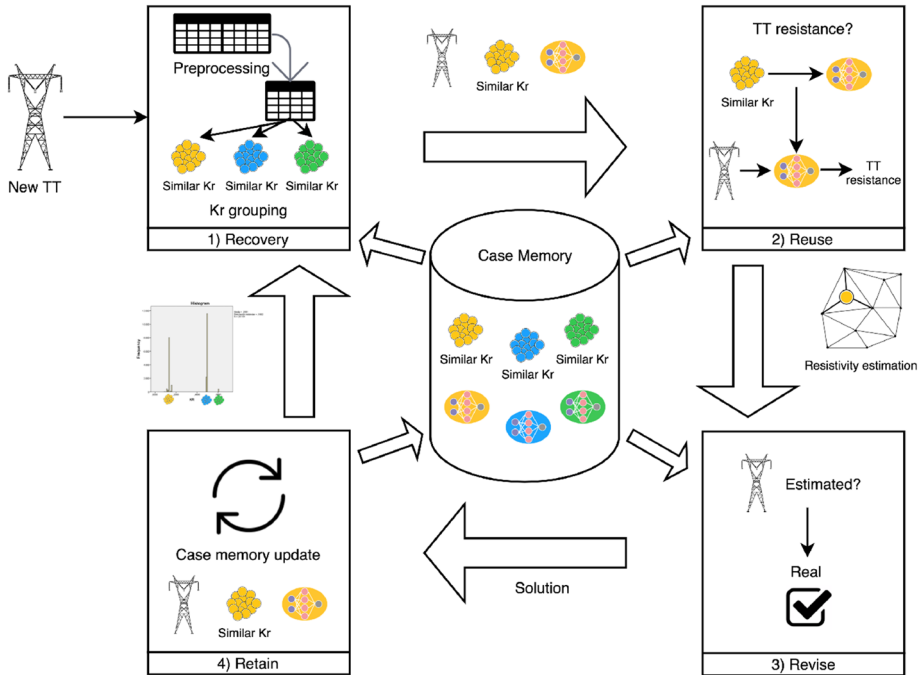


Fig. 2 CBR cycle for the resistivity and resistance estimation

predicted K_p value by using that neural network. Once the prediction is obtained, the agent sends the value to the Broker agent.

- **Resistance:** This role is implemented by agents that load the data from a trained network to predict the resistivity values. If there is no previous training, the agent trains the network. When a new message is received from the Broker agent, the K_R , resistivity and temperature values included in that message are used to obtain the predicted resistance value by using the network. Then, the predicted value is sent to the Broker agent.
- **Resistivity:** This role is implemented by agents that interact with the Broker agent to obtain resistivity predictions from the existing values by considering the filters specified by the user.

3.1 Resistance and resistivity estimation

To estimate resistance as related to the different measured parameters, the only parameters used are those that were shown to influence the final value. To determine what the most influent parameters are, the correlation analysis and Kruskal–Wallis [34] methods were used. Kruskal–Wallis method is used to test if a dataset has originated from the same distribution, determining the dependency between the studied variable and the rest of variables.

Once these parameters were known, the estimation was carried out by a CBR system, as shown in Fig. 2. The case memory is grouped by the TT types as defined by their K_R value. To group them by their K_R value, Partitioning Around Medoids (PAM) is applied; the Henning and Liao [35] proposal is then used to determine the number of clusters. More specifically, three clusters are generated from a list of approximately 60.000 TTs in Spain, of which we only have values for the location and the K_R value. In these three clusters, the

cases base is organized so that every case contains: resistivity, temperature, humidity, K_R and resistance values.

Every cluster has a trained MLP where the inputs are resistivity, humidity, temperature and K_R , and the output is the resistance. The input and output values of the neural network are rescued in the range [0.2, 0.8] since the selected activation function is sigmoidal and we do not want any extreme values for the training. In the middle layer $2n+1$ neurons are placed where n is the number of neurons in the input layer. This criterion is based on Kolmogorov's theorem.

In the recovery phase, the system recovers the previously trained network associated with the new K_R value. In the adaptation phase, the network is used to generate the prediction. Finally, the data and training are updated in the revise phase.

With regards to the proposed CBR algorithm, its input is based either on the known information about a TT that already exists in the system, or a fictitious TT to be placed in a determined location. The only difference for the algorithm is the evaluation of the nearest neighbors.

First, the nearest neighbors have to be found and identified according to different established parameters such as the maximum distance to be considered a neighbor, or the maximum number of neighbors to use.

Once determined, if the TT resistivity is unknown, it is estimated by the Inverse Distance Weighting (DW) method by applying this equation:

$$\bar{\rho} = \sum_1^n \rho(TT_i) * \frac{D_{\max}/d_i}{\sum_1^n D_{\max}/d_i} \quad (1)$$

where $\rho(TT_i)$ is the resistivity of the i th TT, D_{\max} is the largest distance is the distance of the i th TT. The equation provides the resistivity value with an estimated deviation of σ^2 (2), where:

$$\sigma = \sqrt{\frac{1}{n} \sum_1^n (\rho_i - \bar{\rho})^2 \frac{D_{\max}/d_i}{\sum_1^n D_{\max}/d_i}} \quad (2)$$

When the resistivity value and range of the TT are obtained, the value of the resistance can be estimated. In order to estimate the resistance, an artificial neural network is used, specifically, an MLP with four inputs: resistivity, K_R value, ground humidity and temperature. The hidden layer consists of 9 neurons and provides a single output with the value of the estimated resistance.

In addition, agents that implement Kc and Kp roles find the relationship between the K_R value and the Kc and Kp values. To carry out the search of those relationships, another CBR system has been included (Fig. 3). Its behavior is equivalent to both Kc and Kp values. To estimate a new Kc and Kp value, the corresponding ANN is recovered in the first step. In the reuse step, the network is used to estimate the value and, finally, the revise and retain steps store the values once they have been checked. The ANN is trained periodically both manually, by using a graphic user interface to facilitate the process, and automatically, when a defined number of new cases has been included in the system. In manual training, the evolution of the error is analyzed with a set of training data. When the error begins to reduce at a slower rate, the neuronal network is validated with the test data. The training continues, while the error produced in the data of the test reduces.

The ANN functioning and configuration for both step and touch potentials assume the K_R value to be the following: an input neuron with the K_R value, an output neuron with the Kp or Kc value as appropriate, 3 neurons in the hidden layer and sigmoid activation functions.

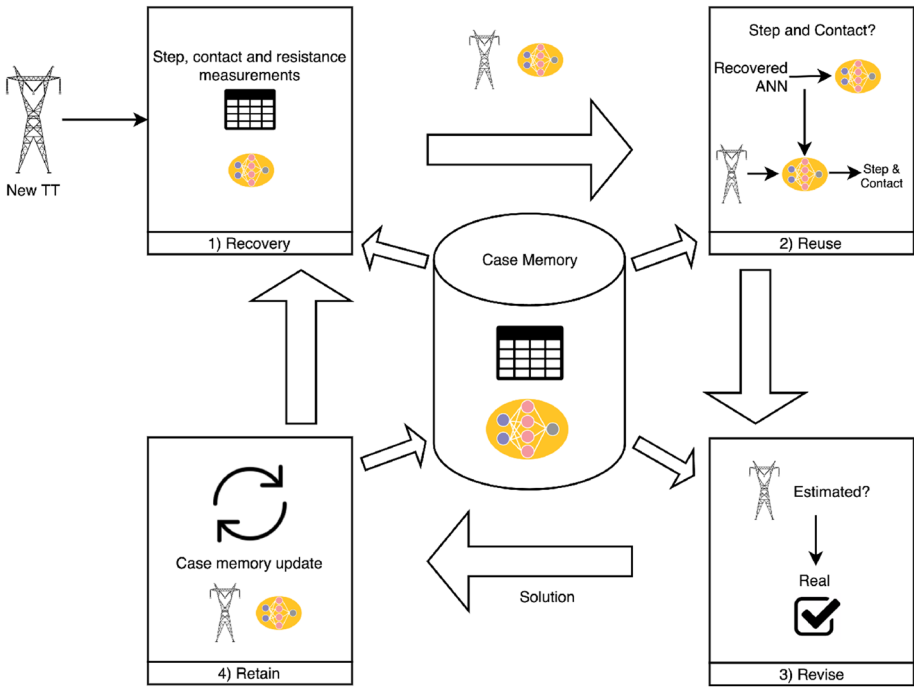


Fig. 3 CBR cycle to predict K_c and K_p from K_R

The individual neurons of the network have a bias and the momentum is included during the ANN training (3) [36].

$$\begin{aligned}
 w_{kj}^p(t+1) &= w_{kj}^p(t) + \eta (d_k^p - y_k^p) (1 - y_k^p) y_j^p k y_j^p + \mu (w_{kj}^p(t) - w_{kj}^p(t-1)) \\
 \theta_k^p(t+1) &= \theta_k^p(t) + \eta (d_k^p - y_k^p) (1 - y_k^p) y_k^p + \mu (\theta_k^p(t) - \theta_k^p(t-1))
 \end{aligned}
 \tag{3}$$

where θ_k^p is the bias associated with the neuron k of the output layer, j is the weight between neuron k in the hidden layer and neuron in the output layer, η is the learning rate, d_k^p is the desired value for the neuron k in the output layer, y_j^p is the output value for the neuron j in the hidden layer, and μ is the momentum. Bias and input layer weights are updated as defined in equation (4).

$$\begin{aligned}
 w_{ji}^p(t+1) &= w_{ji}^p(t) + \eta (1 - (y_j)^p) y_j^p \left(\sum_{k=1}^M (d_k^p - y_k^p) (1 - y_k^p) y_k^p w_{kj} \right) x_i^p \\
 &\quad + \mu (w_{ji}^p(t) - w_{ji}^p(t-1)) \\
 \theta_j^p(t+1) &= \theta_j^p(t) + \eta (1 - y_j^p) y_j^p \left(\sum_{k=1}^M (d_k^p - y_k^p) (1 - y_k^p) y_k^p w_{kj} \right) + \mu (\theta_j^p(t) \\
 &\quad - \theta_j^p(t-1))
 \end{aligned}
 \tag{4}$$

where w_{ji}^p is the weight between neuron i in the output layer and neuron j in the hidden layer, x_i^p is the input value for the neuron i , and θ_j^p is the neuron j bias in the hidden layer.

3.2 Sampling

The main objective of this work is to reduce the number of reviews in order to reduce the maintenance cost of TTs. To attain this objective, a new algorithm was proposed. It uses several TT parameters as inputs, such as the K_R value, which determines the type of the TT, and the TT resistivity value to create stratum which are defined based on both parameters. The proposed algorithm is shown in Algorithm 1 which follows these steps:

1. The sample provided as an algorithm input is analyzed to determine how many K_R intervals will be generated. For every interval, a list with the corresponding TTs is created, so TTs are now grouped by type (K_R value).
2. All lists are sorted in ascending order according to TT resistivity.
3. Once the lists are sorted, the deciles are calculated for each of the lists created according to the resistivity of similar K_R TTs groups (step 2). The lowest (ρ_{\min}) and highest resistivity (ρ_{\max}) values of every list are also stored.
4. For every list of K_R groups (step 2), there are 10 sublists (step 3) containing TTs with a similar structure (associated with the K_R value) and resistance. The algorithm will now contain lists (equal to the number of K_R intervals selected in step 1), which are in turn divided into 10 sublists (d), which are generated after calculating the deciles in the lists found at the first level. Variance (σ^2) is calculated and stored for every sublist created in step 3.
5. The maximum error e is calculated and stored for every sublist from step 3. This error is considered as the maximum between 0.1 mid-point of one decile and 0.1 distance of each decile (5):

$$e[i][j] = \max\left[0.1 * \frac{\rho_{\min}[i][j] + \rho_{\max}[i][j]}{2}, 0.1 * (\rho_{\max}[i][j] - \rho_{\min}[i][j])\right] \quad (5)$$

6. From the sublists generated in step 3, the variance σ^2 calculated in step 4, and the maximum error calculated in step 5, the number of TTs to be reviewed for every sublist is defined by the following equation:

$$sampleSize[i][j] = \frac{|d[i][j]| * z^2 * \sigma^2[i][j]}{((|d[i][j]| - 1) * e[i][j]^2 + z^2 * \sigma^2[i][j])} \quad (6)$$

where: $|d[i][j]|$ is the size number of TTs of the decile $d[i][j]$, $sampleSize$ is the number of TTs to review for every sublist, $z = 1.9599$, $\sigma^2[i][j]$ is the variance of the decile $d[i][j]$, $e[i][j]$ is the maximum error allowed.

7. The output of the algorithm, $sampleSize$ is provided for every sublist d .

If the sample size is too high in any of the sublists d , a variant of the algorithm is applied, as explained below:

1. Both the selected sublists d and the calculated number of reviews are taken as input.
2. A new and empty sublist is created where the lower value in the resistivity sublist is stored.
3. Then, the list is iterated and:
 - If the i th element is higher than the maximum number of reviews allowed, the average value between the i th decile and the $(i + 1)$ th decile is inserted into the list and then, the $(i + 1)$ th.
 - If the i th element is not higher than the maximum, only the $(i + 1)$ th value is inserted.

So the corresponding subdivision ranges are being inserted into the decile list.

Algorithm 1 Sampling algorithm

```

1: function SAMPLING( $\bar{T}\bar{T}$ )
2:    $TT_{K_R} = \text{groupByType}(\bar{T}\bar{T})$                                 ▷ Step 1: Get  $n$  TT lists grouped by  $K_R$  intervals
3:    $n = |\bar{T}\bar{T}_{K_R}|$                                             ▷  $n$  is the number of TT types
4:   for  $i = 1; i \leq n$  do
5:      $dim_i = |\bar{T}\bar{T}_{K_R}[i]|$                                     ▷  $dim_i$  is the number of elements in every type
6:      $TT_{K_R}[i] = \text{orderByResistivity}(TT_{K_R}[i], asc)$         ▷ Step 2
7:     for  $j = 1; j \leq 10$  do                                  ▷ Step 3
8:        $d[i][j] = \text{getResistivityDecile}(j, TT_{K_R}[i])$ 
9:        $\rho_{max}[i][j] = \text{getMaxResistivity}(d[i][j])$            ▷ Higher resistivity value for decile  $j$  of the list  $i$ 
10:       $\rho_{min}[i][j] = \text{getMinResistivity}(d[i][j])$            ▷ Lower resistivity value for decile  $j$  of the list  $i$ 
11:       $\sigma^2[i][j] = \text{getResistivityVariance}(d[i][j])$        ▷ Step 4
12:    end for
13:  end for
14:  for  $i = 1; i \leq n$  do
15:    for  $j = 1; j \leq 10$  do
16:       $e[i][j] = \text{getMaxError}(d[i][j], d[i][j+1], \rho_{max}[i][j], \rho_{min}[i][j])$  ▷ Step 5: Get maximum error
17:    end for
18:  end for
19:  for  $i = 1; i \leq n$  do
20:     $sampleSize[i][j] = |d[i][j]| * z^2 / ((|d[i][j]| - 1) * e[i][j]^2 + z^2 * \sigma^2[i][j])$  ▷ Step 6: get sample size
    for every  $TT_{K_R}[i]$ 
21:  end for
22:  return  $sampleSize$                                        ▷ Step 7: return the sample size for every  $TT_{K_R}[i]$ 
23: end function

```

4. With this new list, steps 4–7 from the Algorithm 1 are applied.
5. If the number of reviews is still too high, the algorithm returns to step 1 of this algorithm and the process is repeated.

4 Results

The dataset used to create the system has a revision history of 600,000 TTs operating over 42,000 km. of high voltage power lines. This dataset is provided by one of the companies that participate in this work and it cannot be published. However, data could be generated with a simulator. The range of values used for the most important parameters is described below:

K_R values, as mentioned, are related to the type of TT. The company only uses three types of TT in Spain, whose values are included in the following ranges: (0.089–0.155), (0.491–0.4991), (0.6–0.672).

There are no available digital resistivity maps of Spain, so the last resistivity value stored in the revision history of each TT has been used. Resistivity is a value that normally ranges from (1 to 1000) Ω m. In the current case, we have resistivity values between (1 and 432.94) Ω m.

The resistance values of TTs must be lower than 20 Ω because it must be lower than the resistance of a human body, so in case of an electric shock, the current flows through the structure of the TT to the ground, instead of doing it through the human body.

The result of the process is a software application with a series of tools for companies in charge of reviewing the TTs. In this application, users can select one or more lines (every line includes a set of TTs) to be reviewed, and the system will show the user a subset of TTs to review that would ensure that the system works fine with a confidence level of 95% (this

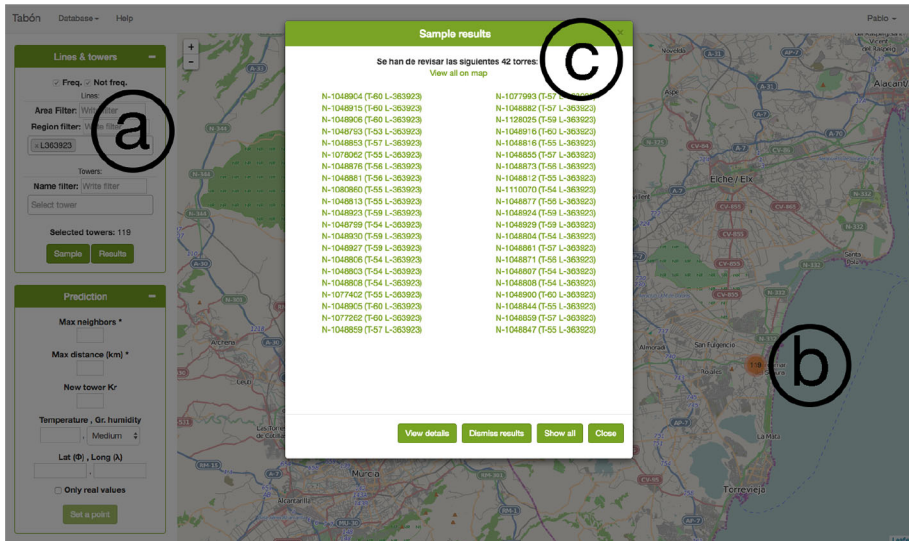


Fig. 4 Web application, sampling results

percentage is directly related to the value of z). This makes it possible to reduce the number of TTs to be reviewed, with the added benefit of improving efficiency and reducing costs.

Figure 4 shows the process to execute the algorithm and obtain the result. The figure shows an example where some TTs close to the region of Valencia (Spain) are selected (“a”) and shown on the map (“b”). When the algorithm is executed, the system proposes the review sample. In this case, 42 TTs from the initial 119 TTs are proposed (“c”). The list identifies all the TTs that have to be reviewed. The user can see where the proposed TTs are on the map, as well as the details of every algorithm step by using the different options.

The web application provides a functionality to determine the best structure model to use in every location. In addition, unknown resistivity and resistance values of existing TTs can be estimated. This is a very frequent situation because there is no information from most TTs throughout Spain. Figure 5 shows an example of this functionality, where values for a new TT (represented by the red marker at “3”) are estimated by taking into consideration just 5 neighbors (this is for demonstration purposes only; in reality, 5 is a really poor number for real cases), represented by blue markers.

In general, the efficiency of the algorithm is evaluated according to the accuracy of the estimation and the proposed reduction percentage, which are shown in Tables 1 and 2. To evaluate the system, 1600 TTs were reviewed and 1200 were included in the system. Then, the algorithms were applied.

First, the resistivity and the resistance of the remaining 400 TTs were estimated. When the real value is inside the range estimated by the system (the proposed value \pm the deviation value), it is considered correct, otherwise is a mistake.

The system shows a very high accuracy (98.85%) when estimating values for TTs whose neighbors were within an average distance of 1 km. The value is slightly lower when the average distance is increased to 5 km, 97.74%. Taking longer distances into account for the estimation makes no sense if there are enough neighbors at a closer distance.

Regarding the sampling algorithm that reduces the number of TTs to be reviewed from a selected set of TTs, the system achieved the results shown in Table 2.

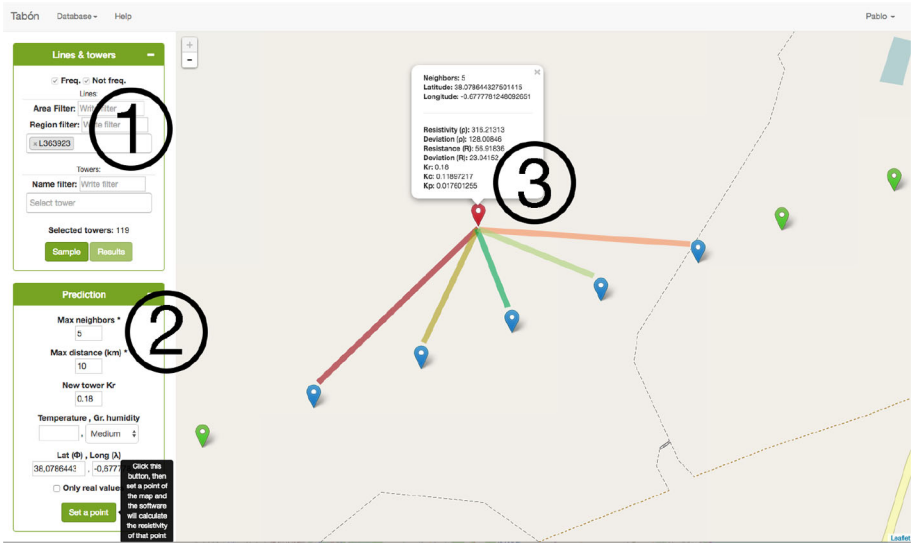


Fig. 5 Web application, new or existing TT values prediction process (color figure online)

Table 1 Accuracy percentage on the estimation according to the mean distance

Average distance (km)	Accuracy of the estimation (%)
~ 1 (km)	98.85
~ 5 (km)	97.74

Table 2 Reduction percentage depending on the number of selected TTs and the average distance

TTs	Average distance (km)	Proposed TT	Reduction (%)
100	~ 50	27	73
200	~ 50	35	82
500	~ 50	64	87.2
800	~ 50	83	89.625
800	~ 300	154	80.75

The reduction percentage is higher when considering TTs with the closer average distance and when more TTs are selected.

The economic savings obtained because of this reduction are not directly linked to the TT revision reduction percentage. When evaluating the costs associated with revisions, different indirect parameters are related, such as the time that operators take to access the TT. Due to the characteristics of the sampling algorithm, the TTs that operators have to check are usually far from each other (since TTs that are close usually have common characteristics), so the cost of displacement increases.

5 Conclusions and future work

As previously seen, the best results are achieved when the more TTs are selected and when their average distance is closer. This is because of the heterogeneity of the ground. Therefore, when a full database with the resistivity of all TTs is built, this will not represent a problem, suggesting the efficiency and accuracy of the system is expected to increase with future work as the information and the values of existing TTs become more complete.

The current methodology requires that all TTs are revised. The proposed tool based on predictive maintenance would reduce the maintenance costs since only a part of the existing TTs would need to be revised. At present, no similar tools are applied to this problem.

The advantage of using VO is the ability to apply new prediction techniques without modifying other parts of the system, so the techniques applied could be edited in the future when all the data are real.

In the developed system, CBR systems are used to store the information of the reviewed TTs. In order to make predictions, CBR systems incorporate neural networks in their technical stages. These prediction techniques are updated dynamically according to the CBR definition. Consequently, the CBR system allows to define the systems evolution process as New data are being added to the system.

As stated, the process of training neural networks can be automatic. Automatic training is based on the decrease of error from a dataset used for validation. This automatic process hinders the possibility of applying other techniques, such as statistical regression models. This is so, because when applying concrete regression models data is required to follow a particular distribution; this cannot be achieved automatically, instead, manual labor is necessary.

Consequently, that is precisely the future work for this project. It is necessary to increase the real cases database in order to evaluate the system with more real data and obtain better results which are not dependent on the ground resistivity estimation.

Another interesting line of research that has been dealt with in this work and will be continued is the analysis of the relationship between economic savings and TT revision reduction provided by the presented sampling system.

Acknowledgements This work has been supported by the European Commission H2020 MSCA-RISE-2014: Marie Skłodowska-Curie project DREAM-GO Enabling Demand Response for short and real-time Efficient And Market Based Smart Grid Operation—An intelligent and real-time simulation approach Ref 641794. The research of Pablo Chamoso has been financed by the Regional Ministry of Education in Castilla y León and the European Social Fund (Operational Programme 2014–2020 for Castilla y León, EDU/310/2015 BOCYL).

References

1. Eltawil MA, Zhao Z (2010) Grid-connected photovoltaic power systems: technical and potential problems—a review. *Renew Sustain Energy Rev* 14(1):112–129
2. Gonçalves RS, Carvalho JCM (2013) Review and latest trends in mobile robots used on power transmission lines. *Int J Adv Robot Syst* 10:1–14
3. Smith CA., Corripio AB, Basurto SDM (1991) Control automático de procesos: teoría y práctica. Limusa
4. Taher SA, Sadeghkhani I (2010) Estimation of magnitude and time duration of temporary overvoltages using ANN in transmission lines during power system restoration. *Simul Model Pract Theory* 18(6):787–805
5. Trappey AJ, Trappey CV, Ma L, Chang JC (2015) Intelligent engineering asset management system for power transformer maintenance decision supports under various operating conditions. *Comput Ind Eng* 84:3–11
6. Yam RCM, Tse PW, Li L, Tu P (2001) Intelligent predictive decision support system for condition-based maintenance. *Int J Adv Manuf Technol* 17(5):383–391

7. de Faria H, Costa JGS, Olivás JLM (2015) A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis. *Renew Sustain Energy Rev* 46:201–209
8. Zambonelli F, Jennings NR, Wooldridge M (2003) Developing multiagent systems: the Gaia methodology. *ACM Trans Softw Eng Methodol (TOSEM)* 12(3):317–370
9. Ghazvini MF, Morais H, Vale Z (2012) Coordination between mid-term maintenance outage decisions and short-term security-constrained scheduling in smart distribution systems. *Appl Energy* 96:281–291
10. Swanson L (2001) Linking maintenance strategies to performance. *Int J Prod Econ* 70(3):237–244
11. Na MG (2001) Auto-tuned PID controller using a model predictive control method for the steam generator water level. *IEEE Trans Nucl Sci* 48(5):1664–1671
12. Krishnanand KR, Dash PK, Naeem MH (2015) Detection, classification, and location of faults in power transmission lines. *Int J Electr Power Energy Syst* 67:76–86
13. Higgins LR, Mobley RK, Smith R (2002) *Maintenance engineering handbook*. McGraw-Hill, New York
14. Do P, Voisin A, Levrat E, Lung B (2015) A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliab Eng Syst Saf* 133:22–32
15. Zarnani A, Musilek P, Shi X, Ke X, He H, Greiner R (2012) Learning to predict ice accretion on electric power lines. *Eng Appl Artif Intell* 25(3):609–617
16. Zhou D, Zhang H, Weng S (2014) A novel prognostic model of performance degradation trend for power machinery maintenance. *Energy* 78:740–746
17. Weibull W (1951) Wide applicability. *J Appl Mech* 103:33
18. Duval M, DePabla A (2001) Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases. *IEEE Electr Insul Mag* 17(2):31–41
19. Boella G, Hulstijn J, Van Der Torre L (2005) Virtual organizations as normative multiagent systems. In: *Proceedings of the 38th annual Hawaii international conference on system sciences*. IEEE, p 192c, Jan 2005
20. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. *Int J High Perform Comput Appl* 15(3):200–222
21. Rodríguez S, Julián V, Bajo J, Carrascosa C, Boti V, Corchado JM (2011) Agent-based virtual organization architecture. *Eng Appl Artif Intell* 24(5):895–910
22. de Paz JF, Tapia DI, Alonso RS, Pinzón C, Bajo J, Corchado JM (2013) Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks. *Knowl Inf Syst* 34(1):193–217
23. Bajo J, Borrajo ML, de Paz JF, Corchado JM, Pellicer MA (2012) A multi-agent system for web-based risk management in small and medium business. *Expert Syst Appl* 39(8):6921–6931
24. Polson NG, Sokolov VO (2017) Deep learning for short-term traffic flow prediction. *Transp Res C Emerg Technol* 79:1–17
25. Cardoso G, Rolim JG, Zurn HH (2004) Application of neural-network modules to electric power system fault section estimation. *IEEE Trans Power Deliv* 19(3):1034–1041
26. Dudek G (2016) Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting. *Int J Forecast* 32(3):1057–1060
27. Sazzad MD, Zhi Ong ZC, Ismail Z, Khoo SY (2017) A comparative study of vibrational response based impact force localization and quantification using radial basis function network and multilayer perceptron. *Expert Syst Appl* 65:87–98
28. Krenek J, Kuca K, Blazek P, Krejcar O, Jun D (2016) Application of artificial neural networks in condition based predictive maintenance. In: Król D, Madeyski L, Nguyen N (eds) *Recent developments in intelligent information and database systems*. Springer, Berlin, pp 75–86
29. Aazi FZ, Abdesselam R, Achchab B, Elouardighi A (2016) Feature selection for multiclass support vector machines. *AI Commun* 29(5):583–593
30. de Paz JF, Bajo J, González A, Rodríguez S, Corchado JM (2012) Combining case-based reasoning systems and support vector regression to evaluate the atmosphere–ocean interaction. *Knowl Inf Syst* 30(1):155–177
31. Kolodner J (2014) *Case-based reasoning*. Morgan Kaufmann, Los Altos
32. Zato C, Villarrubia G, Sánchez A, Barri I, Rubión E, Fernández A, Rebate C, Cabo JA, Álamos T, Sanz J, Seco J, Bajo J, Corchado JM (2012) PANGEA–Platform for automatic construction of organizations of intelligent agents. In: Omatu S, De Paz Santana J, González S, Molina J, Bernardos A, Rodríguez J (eds) *Advances in intelligent and soft computing*, vol 151. Springer, Berlin, pp 229–239
33. Singh J, Gandhi K, Kapoor M, Dwivedi A (2013) New approaches for live wire maintenance of transmission lines. *MIT Int J Electr Instrum* 3:67–71
34. Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. *J Am Stat Assoc* 47(260):583–621

35. Hennig C, Liao T (2013) How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *J R Stat Soc Ser C Appl Stat* 62:309–369
36. de Paz JD, Bajo J, López VF, Corchado JM (2013) Biomedic organizations: an intelligent dynamic architecture for KDD. *Inf Sci* 224:49–61



Pablo Chamoso is currently a Ph.D. Student on Computer Engineer. He is member of the BISITE Research group. He obtained a Technical Engineering in Computing in 2010 at the University of Salamanca, then an Engineering in Computer Sciences degree in 2013 at the same university. Then, he obtained a postgraduate in Intelligent Systems at the University of Salamanca in 2014. He is co-author of more than 20 papers published in recognized journal, workshops and symposiums, most of them related to Artificial Intelligence, Visual Analysis and Distributed Systems. He has been member of the organizing and scientific committee of several international symposiums such as PAAMS, FIADI, SAC, ISAMI, PACBB.



Juan F. De Paz received a Ph.D. in Computer Science from the University of Salamanca (Spain) in 2010. He is Assistant Professor at the University of Salamanca and researcher at the BISITE research group (<http://bisite.usal.es>). He obtained a Technical Engineering in Systems Computer Sciences degree in 2003, an Engineering in Computer Sciences degree in 2005 at the University of Salamanca and Statistic degree in 2007 in the same University. He has been co-author of published papers in several journals, workshops and symposiums.



Javier Bajo is Associate Professor at the ETS Ingenieros Informáticos—Universidad Politécnica de Madrid (UPM). He is currently Coordinator of the Research Master in Artificial Intelligence at UPM. Previously, he hold the position of Associate Professor at the Universidad Pontificia de Salamanca (2003–2012) and Director of the Data Center in the same University (2010–2012). He obtained a PhD in computer sciences from the Universidad de Salamanca (with honors) (2007) and a Master in E-Commerce in the same University (2006). He obtained a bachelor in computer sciences at the Universidad de Valladolid (2001) and a MsC in computer sciences at the Universidad Pontificia de Salamanca (2003). His research efforts are focused on multi-agent systems, social computing and ambient intelligence. He has participated in more than 50 research projects (funded by European Commission, National or Regional entities) and contracts with private companies, acting as principal researcher in 11 of these projects. He is co-author of more than 300 scientific publications (books, Journal papers, conference papers).

49 of these publications have been published in international journals indexed in the ISI JCR reference index.

He has been co-chairman of the organizing committee of more than 30 recognized international conferences (ACM SAC, IEEE FUSION, PAAMS, ...).



Gabriel Villarrubia received a Ph.D. in Computer Engineering. Currently, he is Assistant Professor at the Department of Computer Science at the University of Salamanca and member of the BISITE Research group. He obtained a Technical Engineering in Management of Computer in 2009 at the Pontifical University of Salamanca, then an Engineering in Computer Sciences degree in 2010 at the same university and postgraduate in Intelligent Systems at the University of Salamanca in 2011. He is co-author of more than 50 papers published in recognized journal, workshops and symposiums, most of them related to Intelligent Systems, Ambient Intelligence, and Distributed Systems. He has been member of the organizing and scientific committee of several international symposiums such as FUSION, PAAMS, MIS4TEL, ISAMI, PACBB.