Identification of Underground Faults using Internet of Things (IoT)

R Rengaraj, G R Venkatakrishnan*, Adithya Pillai R, Abinandhan R, Dev Ganesh S, Aravind K

Department of Electrical and Electronics Engineering, Sri Sivasubramaniam Nadar College of Engineering, Old Mahabalipuram Rd, OMR, Kalavakkam, Tamil Nadu 603110

venkatakrishnangr@ssn.edu.in, rengarajr@ssn.edu.in, adithyapillai16006@eee.ssn.edu.in, abinandhan16004@eee.ssn.edu.in, devganesh160023@eee.ssn.edu.in, aravind16012@eee.ssn.edu.in

Abstract. This paper aims at the detection of the fault occurring in an underground cable system, pinpoint the location of the fault using Extreme Learning Machine and convey the parameters of the fault occurred using Internet of Things (IoT). The underground cables have seen a steep increase in usage due to its many inherent advantages compared to the overhead cables. However, various methods to pinpoint the presence of fault in the underground cables of large lengths has proved futile and the communication of the fault information has been proved to be expensive. Our proposed project uses Extreme Learning Machine to detect the type of fault and to pinpoint the location of the fault in the system. Extreme Learning Machine Classification and Regression Algorithms are utilized to predict the type of fault in the system and the distance of the fault from the sending end. The regression algorithm is compared to other ML algorithms relevant to the subject matter at hand. The fault location conveyed by the use of GSM is limited by the fact that only text messages in the form of Short Messaging Services can be sent to the user concerned increasing the cost incurred by the user. Our project uses the internet to communicate through a big arsenal of methods, like e-mail and other social media services possible, thus widening the scope of communication while decreasing the size of the product.

Keywords. Fault location, Extreme Learning machine (ELM), Fault Identification, Time Domain Reflectometry

1. Introduction
The first evidences of undergrounding can be traced back to its use in mining explosives and undersea telegraph cables in the early 19th century. Due to the immense development of early electrical systems a few years later, the need for undergrounding increased leading to the application of underground cables for the transmission of electrical power. From Thomas Edison using Jute to insulate his underground DC “street pipes” in the late nineteenth century to the conception of synthetic polymers with high insulation resistance, high dielectric strength and several other properties to combat more modern problems in the arena, the level of technology associated with the underground cables has come a long way.

The inherent advantages of undergrounding have played a pivotal role in the exponential growth in technology associated with it. Less risk of damage, reduced risk of electromagnetic radiation in the surrounding and decreased risk of fire in the event of a fault are some of the reasons that supported the need for research in this area. Combating the drawbacks of underground cables has however...
consumed more resources than its development and application. The expensive nature of undergrounding coupled with difficulties in voltage control due to high charging currents owing to high reactive power in the cables has hindered its application in many a field. The maintenance and repair of the underground cables in the event of damage takes days or weeks which has resulted in the existence of redundant lines adding to the its list of shortcomings. In these detrimental areas, the overhead lines pose a strong advantage leading to its prominent usage.

A staggering amount of research has been applied into the accurately locating the fault in an underground cable using a various arsenal of methods. Although a majority of those methods proved to be fruitful, only a handful of them were actually implemented in real time underground cable systems. Determining fault parameters in an underground system is facilitated in two steps where in the fault type is determined first followed by the location of the fault. The communication of fault parameters is crucial to interpret the fault type and location [1]. Extreme Learning Machine is primarily used in the single layer feed forward networks where the hidden layer doesn’t need to be like neurons. It has inherent advantages over traditional methods, such as faster learning, and neglecting local minima issues. It also uses only one hidden layer, thus making it easier to tune the network for the required purpose. A wide range of methods has proved to be useful in analysing the parameters of fault, ranging from Cable thumping methods to High Voltage Radar methods [2]. The aforementioned techniques and methods have succeeded in providing an approximate point of fault in underground cable systems while completely overlooking the type of fault that has occurred.

2. Proposed Methodology
The process of detection and location of underground faults is shown in figure 1. The first step of this process is to simulate the fault conditions of an underground system.

![Figure 1. Block Diagram of Proposed Methodology](image)

The simulations are performed using the PSCAD software for different faults like LL and LLL faults. The outputs from this simulation is stored as data for the next step. The device used for performing Extreme learning machine is a Raspberry Pi, which is capable of running a Linux environment. The flow of data in this paper is shown in figure 2.

![Figure 2. ELM](image)

The data generated from the simulation is given as input to the normalising algorithm generated to improve the accuracy of the Extreme Learning Machine (ELM) algorithm. One of the ELM classifiers identify the type of fault and another ELM regressor finds the location of the fault. The percentage error of this algorithm is given below.

\[
%error = \left(\frac{\text{Fault Locator Output} - \text{Actual Fault Location}}{\text{Total Length of line}}\right) \times 100 \quad \ldots (1)
\]
3. Tools and Concepts

3.1. PSCAD
PSCAD version 4.5 was used to simulate the three phase fault circuit as shown in figure 4. The fault circuit comprises of two cables connected to each other with parameters as shown in Table 1. The cable parameters and construction is alterable and the construction made use of in this paper is shown in figure 3.

3.2. MATLAB
MATLAB version R2018a was used as the interface to code the Extreme Learning Machine algorithm and implement it to obtain results.

3.3. Raspberry Pi
A Raspberry Pi capable of running the Linux environment is used to integrate the software and the hardware discussed in the paper and uses Internet of Things to communicate the fault information.

3.4. Extreme Learning Machine
Extreme machine learning is used primarily used in the single layer feed forward networks where the hidden layer need not be neuron like. The output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model. ELM has some inherent advantages over the traditional feedforward network like faster learning speed, better generalisation characteristics and neglecting the issues like local minima and the lot.[3]

3.5. Time Domain Reflectometry
An estimation technique in which a pulse is generated to propagate down a cable, after which the reflected signal returns to the generator and is then interpreted based on its shape, phase, and delay. The results can be used to determine the length of a cable, if and where there is an open circuit, what kind of load a cable is terminated with, and even the relative permittivity and permeability of a dielectric.[4]

![Figure 3. Cross-section view of the Underground cable](image-url)
Table 1. Specifications of Underground Cable

| Parameter                      | Cable 1A     | Cable 4A     |
|--------------------------------|--------------|--------------|
| Cable Name                     | Cable 1A     | Cable 4A     |
| Steady State Frequency         | 50hz         | 50hz         |
| Segment Length                 | 10km         | 90km         |
| Number of Conductors           | 0 (default)  | 0 (default)  |
| Termination Style              | Remote Ends (default) | Remote Ends (default) |
| Coupling of this segment to others | Disabled     | Disabled     |
| Coupled Segment Tag Name       | Row (default) | Row(default) |
| Horizontal Translation of this Segment | 0.0 m (default) | 0.0 m (default) |

4. Simulation and Results

4.1. PSCAD Modelling

To obtain the training data for the ELM network, an underground system is designed with the PSCAD software. This simulated system is a three-phase underground cable system with the capability of introducing various symmetrical and unsymmetrical faults into the circuit. This circuit is shown in figure 4. Two cables of finite length are connected to each other. The length of the first cable is taken as the location at which the fault occurs in the system and the second cable’s length, adding up with the first cable’s length determines the total length of the underground cable. The underground cables used in the circuit shown in figure 4 have the specifications as shown in Table 1. The cross section of the underground cables used in the simulation is shown in figure 3. The cross section and the construction of the cable can be altered to the requirements of the user in the PSCAD software.
4.2. Extreme Learning Machine – Classification of Fault Type

The multiple linear regression model [5] was chosen initially for obtaining the ELM coefficients due to the dependence of the fault distance on various independent variables. The root mean square error (RMSE) obtained from this model was 29.7087, which is extremely high. A sample data set obtained from the PSCAD simulation is shown in Table 2. The ELM algorithm used in this project is implemented in the MATLAB environment [6]. The data is passed as an input to the ELM model in the form of a text file with the expected value in the first column and the independent variables are placed in the successive columns. This model can be used for both classification and regression applications. For classification of the fault type, the Fault Type column was One Hot Encoded as 0 for LL fault and 1 for LLL fault and the rest of the columns were normalised using a normalising algorithm. A sample of the normalised data for the classification of the fault type is shown in Table 3. The classifier output of the ELM has a training accuracy of 0.3755 and a testing accuracy of 0.4804. The output is shown against the expected value in Table 4.

Table 2. Sample Data

| Fault Type | Fault Phases | Phase A | Phase B | Phase C | Fault Current A | Fault Current B | Fault Current C | Distance (KM) |
|------------|--------------|---------|---------|---------|----------------|----------------|----------------|--------------|
| LL         | AB           | 149.91  | 138.66  | 138     | 83.12          | 112.07         | 138            | 5            |
| LL         | BC           | 126.48  | 135.72  | 139.3   | 118.81         | 133.01         | 145.09         | 5            |
| LLL        | ABC          | 359.92  | 344.76  | 364.48  | 89.9           | 344.4          | 364.91         | 2            |
| LLL        | ABC          | 150.71  | 148.74  | 153.16  | 37.65          | 147.99         | 153.86         | 5            |

Table 3. Normalised data to identify type of fault

| Target | Feature 1 | Feature 2 | Feature 3 | Feature 4 | Feature 5 | Feature 6 | Feature 7 |
|--------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0      | -0.18483  | -0.24683  | -0.25047  | -0.5529   | -0.39336  | -0.25047  | -0.98341  |
| 0      | -0.31395  | -0.26303  | -0.2433   | -0.35622  | -0.27797  | -0.2114   | -0.98341  |
| 0      | -0.22534  | -0.25515  | -0.21994  | -0.60636  | -0.32117  | -0.17056  | -0.98341  |
| 0      | -0.59065  | -0.63524  | -0.60636  | -0.80111  | -0.6133   | -0.60388  | -0.95586  |
| 0      | -0.61049  | -0.63232  | -0.60085  | -0.64615  | -0.67965  | -0.60234  | -0.95586  |
| 0      | -0.60349  | -0.64229  | -0.58619  | -0.78017  | -0.60972  | -0.58388  | -0.95586  |
| 1      | -0.89028  | -0.89028  | -0.88642  | -0.98066  | -0.88146  | -0.88201  | -0.81809  |
| 1      | -0.91039  | -0.91039  | -0.90808  | -0.98859  | -0.90025  | -0.90643  | -0.77951  |
| 1      | -0.91723  | -0.91739  | -0.91701  | -0.98733  | -0.90604  | -0.91028  | -0.76298  |
| 1      | -0.92687  | -0.92478  | -0.92296  | -0.98964  | -0.91772  | -0.91965  | -0.73542  |

Table 4. Expected Value vs Actual value for determining the type of fault in ELM

| Expected Value | Actual Value Obtained |
|----------------|-----------------------|
| 1              | 1                     |
| 1              | 0                     |
| 1              | 1                     |
| 1              | 0                     |
| 0              | 0                     |
| 0              | 0                     |
| 1              | 1                     |
| 1              | 0                     |
4.3. Extreme Learning Machine – Predicting the distance of fault

For the fault distance prediction the distance column was kept as the expected value and the rest of the columns were normalised using the normalising algorithm [7]. A sample of the normalised data for the regression used in distance prediction using ELM is shown in Table 5. The normalised data is further split into training and testing data to be fed to the ELM model. The training accuracy for the regression model was 10.7039 and the testing accuracy was found to be 10.5881, which is lesser than 29.7087 which was the accuracy obtained by the Multiple Linear Regression model. The comparison with the predicted value and the expected value is shown in Table 6. The error value calculated is as per formula 1. The expected value and the actual value obtained from the ELM regression model to predict the distance of the fault from the sending end are almost equal. This shows that the ELM model is highly accurate and better than the other ML algorithms for regression and classification purposes. The data extracted from PSCAD for this project is not continually updated over time, thus making it fixed. However, time series analysis models like AR require the dataset to be large and that which updates periodically. This is prominent in large scale industries where this project can be implemented. AR modelling is best for long term usage of this idea, but for the data used in this project, AR does not provide the optimal results. Thus, ELM modelling is preferred as it gives better predictions standalone than when AR is combined with ELM.

| Target | Feature 1    | Feature 2    | Feature 3    | Feature 4    | Feature 5    | Feature 6    |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|
| 5     | -0.18483     | -0.24683     | -0.25047     | -0.5529      | -0.39336     | -0.25047     |
| 15    | -0.73217     | -0.74529     | -0.73542     | -0.8511      | -0.74375     | -0.73542     |
| 20    | -0.80751     | -0.81081     | -0.80354     | -0.88998     | -0.82426     | -0.80205     |
| 20    | -0.8064      | -0.81726     | -0.80235     | -0.80384     | -0.82988     | -0.79974     |
| 99    | -0.97035     | -0.96688     | -0.96743     | -1           | -0.96269     | -0.96302     |

4.4. Communication of fault parameters through Internet of Things methods

After the fault type and fault location are identified, this information has to be sent to the respective authority to inform them about the fault. The SMTP server used here to send the email is from Gmail, and the respective SMTP port is used to get access to Gmail in order to send the email. The ID from which the mail is sent is specified in “GMAIL_USERNAME” and the password to that account is entered in “GMAIL_PASSWORD”. The class “Emailer” is used to create an “Emailer” object to send the mail. The function “sendmail()” is executed to send the mail, and the function definition is specified in the “Emailer” class. The “sendTo” variable specifies the emails to which the fault information has to be sent. This program, when executed, automatically emails from the host ID to all...
the client IDs specified in the code, alerting them about the fault and gives them information about the type of fault and location of the fault.

4.4.1 Algorithm. The algorithm to send an email regarding parameters of fault is shown below.

1. Import the smtplib library
2. Define the SMTP server
3. Define the SMTP port
4. Define the email and password of the email used to send the email
5. Create class Emailer
   a. Define function sendmail() inside Emailer with Recipient, Content, Subject as arguments
      i. Create headers for the email
      ii. Start a new SMTP session
      iii. Initialize connection to the SMTP server
      iv. Establish secure connection with the server
      v. Login to the session
      vi. Send the email
      vii. Quit the session
6. Create Emailer object
7. Define the recipient, content, and subject
8. Call the sendmail() function for the emailer object
9. Stop

5. Conclusion
The results of the Extreme Learning Machine Classifier algorithm to classify the fault and the Extreme Learning Machine Regression algorithm to predict the distance of the fault from the sending end are not without flaw and the sending of fault parameters via social media services clearly shows the wide range of fields covered in this project which ranges from EMTDC simulations to machine learning models to Internet of Things Concepts. The simulation of faults in PSCAD software was accurate and simple due to the presence of a fault block within the software. The generated data set was too large to be processed by the ELM in a short period of time, so regression algorithms were tried out to find the most efficient model and use that model for obtaining coefficients to be used in the ELM model. Multiple Linear regression models proved to be futile due to its high RMSE value and inability to work with continuously monitored data. Auto regression proved to be an efficient model for this, but only when regularly updated large scale data sets are used. The Auto Regression Algorithm can be used in this venture by the implementation of Seasonal AR algorithms like SARIMAX available in the statsmodels library during the presence of large datasets [8]. For smaller data sets like the one used in this project, the usage of ELM proves to be sufficient due to its small size and fast computation speeds.

Acknowledgment
The authors would like to thank SSN College of Engineering for providing the required infrastructure ensuring the success of this project.
References
[1] M. Dinesh, K. Vairaperumal and P. Senthilkumar Design and Detection of Underground Cable Fault Using Raspberry Pi & IoT System International Journal of Trend in Scientific Research and Development ISSN: 2456-6470, Volume-3 | Issue-1, December 2018, pp.668-672
[2] Kiran Daware, Locating Underground Cable Faults, www.electricaleasy.com, 5th May, 2020
[3] Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks, Neurocomputing, vol 70, pp.489-501, 2006
[4] Benjamin Crabtree Building your own Time Domain Reflectometer https://www.allaboutcircuits.com/projects/build-your-own-time-domain-reflectometer/, 6th June, 2020
[5] SuperDataScience Multiple Linear Regression in Python, www.superdatascience.com, 5th May, 2020
[6] Nanyang Technical University, Basic ELM Algorithms, https://www.ntu.edu.sg/, 4th May, 2020
[7] StackExchange How to normalize data in 0 to 1 range, stats.stackexchange.com, 3rd May, 2020
[8] Josef Perktold, Skipper Seabold and Jonathan Taylor statsmodels v0.12.0.dev0(+250) Statsmodels Documentation, 3rd May, 2020